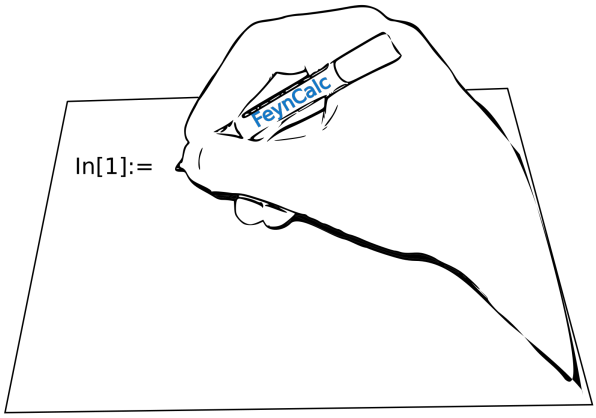
Guide to FeynCalc



A MATHEMATICA package for Quantum Field Theory practitioners

Vladyslav Shtabovenko,
Rolf Mertig and Frederik Orellana

January 3, 2023

This documentation snapshot was generated on **2023-01-03 17:10:25 UTC** from the commit **12512b**

Contents

1 Useful information

1.1 Upper and lower indices

1.1.1 See also

Overview.

Here we list a set of rules that allows to reconstruct the positions of indices (upstairs or downstairs) appearing in FeynCalc expressions

- Every expression must satisfy Einsteins's summation convention, both for Lorentz and Cartesian indices. Single terms containing more than two identical Lorentz or Cartesian indices are illegal and will lead to inconsistent results.
- In a contraction of two Lorentz indices it is understood that one of them is upstairs and the other is downstairs.
- In a contraction of two Cartesian indices, both indices are understood to be upper indices.
- A free Lorentz or Cartesian index is always understood to be an upper index

1.2 Master integrals

1.2.1 See also

Overview.

For the sake of the users that try to employ FeynCalc in calculations beyond 1-loop but do not belong to the multiloop community, let us summarize some basic facts about master integrals in multiloop calculations

1.2.2 Analytic results

- Even at 2-loop there is no library containing analytic results for all master integrals with arbitrary mass distributions. That is, there is simply nothing similar to Package-X beyond 1-loop.
- The available libraries usually focus on very specific integral families, e.g. Mincer (3-loop massless 2-point functions), Forcer (4-loop massless 2-point functions), MATAD, MATAD-ng (massive 3-loop tadpoles), FMFT (massive 4-loop tadpoles), ON-SHELL2 (on-shell 2-point functions with one mass scale).
- The main source for analytic results are scientific publications. When people calculate new master integrals needed for their research, they often provide explicit analytic results in the paper itself or put them into ancillary files accompanying the preprint. Unfortunately, there is no compendium of all calculated integrals that would tell you where to find the corresponding expression. The Loopedia project is an attempt to create something like a search engine for loop integrals, but its database is still far from being comprehensive.

- Some relevant publications for 2-loop 2-point functions include (this list is far from being complete) arXiv:hep-ph/9907431, arXiv:hep-ph/0202123, hep-ph/0307101

1.2.3 Numerical results

- Numerical results are much simpler to obtain and universal libraries that can calculate almost any integral (given enough time and computing resources) are publicly available. Two prominent examples are pySecDec and FIESTA
- Apart from that, there are also libraries that cover specific integral families and may offer better numerical stability due to the corresponding optimizations. Two very useful tools are TVID2 for the evaluation of 3-loop 2-point functions with arbitrary masses and 3VIL for the calculation of 3-loop tadpoles with arbitrary masses.

1.2.4 References

- The most comprehensive pedagogical reference for learning different calculational techniques is the book Feynman Integral Calculus by V. Smirnov.
- The course on Feynman integrals by S. Weinzierl also provides a lot of valuable information.
- Introduction to Loop Calculations by G. Heinrich might be a good start for beginners.

1.3 FeynArts sign conventions

1.3.1 See also

Overview.

The overall sign of an amplitude $i\mathcal{M}$ is always a convention. It does not have to agree between different textbooks and papers and there is nothing wrong with that. Nevertheless, it should be always possible to trace the origins of various signs appearing in different pieces of the amplitude. Understanding how the overall sign comes about allows you to *adjust* it to the convention you prefer.

The amplitude generated by FeynArts' **`CreateFeynAmp`** function has three sources of signs that contribute to the final overall sign. These are the overall prefactor, the definition of the vertices and the fermion sign.

1.3.2 Overall prefactor

- **`CreateFeynAmp`** generates $i\mathcal{M}$ multiplied by the value of the option **`PreFactor`**.
- The default setting of this option is **`-I*(2Pi)^(-4 LoopNumber)`**.
- This means that for a tree-level diagram you get $\mathcal{M}$ and for a 1-loop diagram $(2\pi)^{-4}\mathcal{M}$ is returned.
- To obtain $i\mathcal{M}$ change the value of **`PreFactor`** to **`1`** as in **`CreateFeynAmp[diags, PreFactor -> 1]`**.

1.3.3 Vertices

The signs in the vertices always depend on the model under consideration. The built-in **SM** model is based on the conventions used in arXiv:0709.1075. This means that

- the lepton-gauge boson vertex (e.g. *QED electron-photon vertex*) is proportional to $ie\gamma^\mu$
- the quark-gauge boson vertex is proportional to $-iQg_c\gamma^\mu$, where g_c is the coupling constant and Q is thee electric charge of the quark
- for charged gauge bosons (γ, Z, $W^\pm$) we have $Q = 2/3$ for the up-type quarks (u,c,t) and $-1/3$ for the down-type quarks (d,s,b)
- for gluons one sets $Q = 1$ and so the *QCD quark-gluon vertex* corresponds to $-ig_s\gamma^\mu$

This FeynArts vertex convention agrees with the one used in Gauge Theories of the Strong and Electroweak Interaction by M. Bohm, A. Denner and H. Joos. However, it disagrees with the convention in An Introduction to Quantum Field Theory by M. Peskin and D. Schroeder. There the QED electron-photon vertex is proportional to $-ie\gamma^\mu$, the QCD quark-gluon vertex amounts to $ig_s\gamma^\mu$ and the quark-photon vertex yields $iQg_s\gamma^\mu$.

Notice also that models generated with FeynRules will agree with FeynArts on the QED vertex convention but disagree on the sign of the QCD vertex.

1.3.4 Fermion sign

Diagrams with external fermions receive additional signs that stem from the anticommuting properties of Grassmann fields when applying Wick's theorem. One of the simplest processes that exhibits this effect is the tree-level QED Bhabha scattering $e^+e^- \to e^+e^-$. There are no ambiguities regarding the fact that both amplitudes have a relative minus sign. However, we are free to choose which of the two amplitudes should be multiplied by **+1** and which by **-1**.

The fermion sign algorithm implemented in FeynArts (flip rules) is described in Section 6.6 of the program manual. In the case of the process $e^-e^+ \to e^-e^+$ the s-channel diagram is multiplied by -1, while the t-channel diagram receives a prefactor of $+1$. However, if one generates the physically equivalent process $e^-e^+ \to e^+e^-$, the signs will flip due to the reversed ordering of the final state particles.

Since March 2022 FeynArts features an option to make the fermion sign of each diagram explicit. To this end you just need to evaluate **FermionSign = fSign;** before generating the diagrams. Here **fSign** is a head that will be wrapped around fermion signs. In the case of the Bhabha scattering you can explicitly see that

```
diagsV1 =   InsertFields[CreateTopologies[0, 2 -> 2], {F[2, {1}], -F[2,
{1}]} ->
{F[2, {2}], -F[2, {2}]}, InsertionLevel -> {Classes}, Restrictions ->
QEDOnly];
CreateFeynAmp[diagsV1,PreFactor->1]
```

contains **fSign[-1]**, while

```
diagsV2 =   InsertFields[CreateTopologies[0, 2 -> 2], {F[2, {1}], -F[2,
{1}]} ->
{-F[2, {2}], F[2, {2}]}, InsertionLevel -> {Classes}, Restrictions ->
QEDOnly];
CreateFeynAmp[diagsV2,PreFactor->1]
```

has **fSign[1]**.

1.3.5 General advice

If you are doing a calculation where the overall sign of the amplitude must agree with a particular convention, follow these steps

1. Set the option **PreFactor** of **CreateFeynAmp** to **1** to generate $i\mathcal{M}$.
2. Check compatibility of all vertex signs between your preferred convention and the employed model. If necessary, adjust the signs in the generated amplitudes.
3. Use **FermionSign = fSign;** to figure out the fermion sign of each diagram. If it doesn't agree with your convention, flips the signs for all diagrams in the given process via **amps/. fSign[x_] :> -x**. This will preserve the physical relative signs but change the conventional overall sign.

1.4 Dirac algebra

1.4.1 See also

Overview.

This section contains some explicit formulas used by FeynCalc when simpifying chains of Dirac matrices. Such relations can be found e.g. in Veltman's Gammatrica or rederived by hand.

1.4.2 BMHV algebra

In the Breitenlohner-Maison-'t Hooft-Veltman scheme we are dealing with matrices in D, 4 and $D-4$ dimensions. Many identities of the BMHV algebra can be proven by by decomposing Dirac matrices into two pieces

$$\begin{aligned} &\dim(\gamma^\mu) = D, && \gamma^\mu = \bar{\gamma}^\mu + \hat{\gamma}^\mu, \\ &\dim(\bar{\gamma}^\mu) = 4, && g^{\mu\nu} = \bar{g}^{\mu\nu} + \hat{g}^{\mu\nu}, \\ &\dim(\hat{\gamma}^\mu) = D-4. && p^\mu = \bar{p}^\mu + \hat{p}^\mu. \end{aligned} \tag{1.1}$$

The anticommuatators between Dirac matrices in different dimensions are given by

$$\{\gamma^\mu, \gamma^\nu\} = 2g^{\mu\nu}, \tag{1.2}$$
$$\{\bar{\gamma}^\mu, \bar{\gamma}^\nu\} = \{\gamma^\mu, \bar{\gamma}^\nu\} = 2\bar{g}^{\mu\nu}, \tag{1.3}$$
$$\{\hat{\gamma}^\mu, \hat{\gamma}^\nu\} = \{\gamma^\mu, \hat{\gamma}^\nu\} = 2\hat{g}^{\mu\nu}, \tag{1.4}$$

$$\{\bar{\gamma}^\mu, \hat{\gamma}^\nu\} = 0. \tag{1.5}$$

Notice that while γ^5 anticommutes with all other Dirac matrices in 4 dimensions, it commutes with them in $D-4$ dimensions. However, in D dimensions the anticommutator is not zero

$$\{\bar{\gamma}^\mu, \gamma^5\} = [\hat{\gamma}^\mu, \gamma^5] = 0 \tag{1.6}$$
$$\{\gamma^\mu, \gamma^5\} = \{\hat{\gamma}^\mu, \gamma^5\} = 2\hat{\gamma}^\mu\gamma^5 = 2\gamma^5\hat{\gamma}^\mu. \tag{1.7}$$

For the chiral projectors we obtain

$$P_{L/R}\bar{\gamma}^\mu = \bar{\gamma}^\mu P_{R/L} \tag{1.8}$$
$$P_{L/R}\hat{\gamma}^\mu = \hat{\gamma}^\mu P_{L/R} \tag{1.9}$$
$$P_{L/R}\gamma^\mu = \bar{\gamma}^\mu P_{R/L} + \hat{\gamma}^\mu P_{L/R} = \gamma^\mu P_{R/L} \mp \hat{\gamma}^\mu\gamma^5 \tag{1.10}$$
$$\gamma^\mu P_{L/R} = P_{R/L}\bar{\gamma}^\mu + P_{L/R}\hat{\gamma}^\mu = P_{R/L}\gamma^\mu \mp \gamma^5\hat{\gamma}^\mu \tag{1.11}$$

and

$$P_{L/R}(\bar{\gamma}\cdot\bar{p} + m) = \bar{\gamma}\cdot\bar{p}P_{R/L} + mP_{L/R} = (\bar{\gamma}\cdot\bar{p} - m)P_{R/L} + m \tag{1.12}$$
$$P_{L/R}(\bar{\gamma}\cdot\hat{p} + m) = (\bar{\gamma}\cdot\hat{p} + m)P_{L/R} \tag{1.13}$$
$$P_{L/R}(\bar{\gamma}\cdot p + m) = (\bar{\gamma}\cdot\hat{p} + m)P_{L/R} + \bar{\gamma}\cdot\bar{p}P_{R/L} = (\bar{\gamma}\cdot\bar{p} - m)P_{R/L} + \bar{\gamma}\cdot\hat{p}P_{L/R} + m \tag{1.14}$$

$$(\bar{\gamma}\cdot\bar{p} + m)P_{L/R} = P_{R/L}\bar{\gamma}\cdot\bar{p} + P_{L/R}m = P_{R/L}(\bar{\gamma}\cdot\bar{p} - m) + m$$
$$(\bar{\gamma}\cdot\hat{p} + m)P_{L/R} = P_{L/R}(\bar{\gamma}\cdot\hat{p} + m) \tag{1.15}$$
$$(\bar{\gamma}\cdot p + m)P_{L/R} = P_{L/R}(\bar{\gamma}\cdot\hat{p} + m) + P_{R/L}\bar{\gamma}\cdot\bar{p} = P_{R/L}(\bar{\gamma}\cdot\bar{p} - m) + P_{L/R}\bar{\gamma}\cdot\hat{p} + m \tag{1.16}$$

Notice that package **TRACER** resolves the redundancy of having $\gamma^\mu = \bar{\gamma}^\mu + \hat{\gamma}^\mu$ by eliminating $\bar{\gamma}^\mu$ and offering a function that reintroduces it at the end of the calculation.

Contractions of Dirac matrices and vectors with the metric read

$$\begin{aligned}
&g^{\mu\nu}\gamma_\nu = \gamma^\mu, && g^{\mu\nu}p_\nu = p^\mu,\\
&\bar{g}^{\mu\nu}\bar{\gamma}_\nu = g^{\mu\nu}\bar{\gamma}_\nu = \bar{g}^{\mu\nu}\gamma_\nu = \bar{\gamma}^\mu, && \bar{g}^{\mu\nu}\bar{p}_\nu = g^{\mu\nu}\bar{p}_\nu = \bar{g}^{\mu\nu}p_\nu = \bar{p}^\mu,\\
&\hat{g}^{\mu\nu}\hat{\gamma}_\nu = g^{\mu\nu}\hat{\gamma}_\nu = \hat{g}^{\mu\nu}\gamma_\nu = \hat{\gamma}^\mu, && \hat{g}^{\mu\nu}\hat{p}_\nu = g^{\mu\nu}\hat{p}_\nu = \hat{g}^{\mu\nu}p_\nu = \hat{p}^\mu,\\
&\bar{g}^{\mu\nu}\hat{\gamma}_\nu = \hat{g}^{\mu\nu}\bar{\gamma}_\nu = 0, && \bar{g}^{\mu\nu}\hat{p}_\nu = \hat{g}^{\mu\nu}\bar{p}_\nu = 0.
\end{aligned} \tag{1.17}$$

Contractions of the metric with itself

$$\begin{aligned}
&g^{\mu\nu}g_{\nu\rho} = g^\mu_\rho && g^{\mu\nu}g_{\mu\nu} = d,\\
&\bar{g}^{\mu\nu}\bar{g}_{\nu\rho} = g^{\mu\nu}\bar{g}_{\nu\rho} = \bar{g}^{\mu\nu}g_{\nu\rho} = \bar{g}^\mu_\rho && \bar{g}^{\mu\nu}\bar{g}_{\mu\nu} = g^{\mu\nu}\bar{g}_{\mu\nu} = \bar{g}^{\mu\nu}g_{\mu\nu} = 4,\\
&\hat{g}^{\mu\nu}\hat{g}_{\nu\rho} = g^{\mu\nu}\hat{g}_{\nu\rho} = \hat{g}^{\mu\nu}g_{\nu\rho} = \hat{g}^\mu_\rho && \hat{g}^{\mu\nu}\hat{g}_{\mu\nu} = g^{\mu\nu}\hat{g}_{\mu\nu} = \hat{g}^{\mu\nu}g_{\mu\nu} = d-4,\\
&\bar{g}^{\mu\nu}\hat{g}_{\nu\rho} = \hat{g}^{\mu\nu}\bar{g}_{\nu\rho} = 0, && \bar{g}^{\mu\nu}\hat{g}_{\mu\nu} = \hat{g}^{\mu\nu}\bar{g}_{\mu\nu} = 0.
\end{aligned} \tag{1.18}$$

Contractions of Dirac matrices and vectors with themselves

$$\begin{aligned}
&\gamma^\mu\gamma_\mu = D, && p^\mu p_\mu = p^2,\\
&\bar\gamma^\mu\bar\gamma_\mu = \gamma^\mu\bar\gamma_\mu = \bar\gamma^\mu\gamma_\mu = 4, && \bar p^\mu \bar p_\mu = \bar p^\mu p_\mu = p^\mu \bar p_\mu = \bar p^2,\\
&\hat\gamma^\mu\hat\gamma_\mu = \gamma^\mu\hat\gamma_\mu = \hat\gamma^\mu\gamma_\mu = D-4, && \hat p^\mu \hat p_\mu = \hat p^\mu p_\mu = p^\mu \hat p_\mu = \hat p^2,\\
&\bar\gamma^\mu\hat\gamma_\mu = \hat\gamma^\mu\bar\gamma_\mu = 0, && \bar p^\mu \hat p_\mu = \hat p^\mu \bar p_\mu = 0.
\end{aligned} \tag{1.19}$$

Dirac slashes

$$\begin{aligned}
&\gamma^\mu p_\mu = \gamma\cdot p,\\
&\bar\gamma^\mu \bar p_\mu = \bar\gamma^\mu p_\mu = \gamma^\mu \bar p_\mu = \bar\gamma\cdot\bar p,\\
&\hat\gamma^\mu p_\mu = \hat\gamma^\mu p_\mu = \gamma^\mu \hat p_\mu = \hat\gamma\cdot\hat p,\\
&\bar\gamma^\mu \hat p_\mu = \hat\gamma^\mu \bar p_\mu = 0.
\end{aligned} \tag{1.20}$$

Index pairs with one, two, three, four or five free indices

$$\begin{aligned}
&\gamma^\mu\gamma^\nu\gamma_\mu = -(d-2)\gamma^\nu, && \bar\gamma^\mu\bar\gamma^\nu\bar\gamma_\mu = -2\bar\gamma^\nu, && \hat\gamma^\mu\hat\gamma^\nu\hat\gamma_\mu = -(d-6)\hat\gamma^\nu,\\
&\gamma^\mu\bar\gamma^\nu\gamma_\mu = -(d-2)\bar\gamma^\nu, && \bar\gamma^\mu\gamma^\nu\bar\gamma_\mu = -4\gamma^\nu + 2\bar\gamma^\nu, && \hat\gamma^\mu\gamma^\nu\hat\gamma_\mu = -(d-4)\gamma^\nu + 2\hat\gamma^\nu,\\
&\gamma^\mu\hat\gamma^\nu\gamma_\mu = -(d-2)\hat\gamma^\nu, && \bar\gamma^\mu\hat\gamma^\nu\bar\gamma_\mu = -4\hat\gamma^\nu, && \hat\gamma^\mu\bar\gamma^\nu\hat\gamma_\mu = -(d-4)\bar\gamma^\nu,
\end{aligned} \tag{1.21}$$

$$\gamma^\mu\gamma^\nu\gamma^\rho\gamma_\mu = (d-4)\gamma^\nu\gamma^\rho + 4g^{\nu\rho}I, \tag{1.22}$$
$$\gamma^\mu\bar\gamma^\nu\bar\gamma^\rho\gamma_\mu = (d-4)\bar\gamma^\nu\bar\gamma^\rho + 4\bar g^{\nu\rho}I, \tag{1.23}$$
$$\gamma^\mu\hat\gamma^\nu\hat\gamma^\rho\gamma_\mu = (d-4)\hat\gamma^\nu\hat\gamma^\rho + 4\hat g^{\nu\rho}I, \tag{1.24}$$
$$\bar\gamma^\mu\bar\gamma^\nu\bar\gamma^\rho\bar\gamma_\mu = 4\bar g^{\nu\rho}, \tag{1.25}$$
$$\bar\gamma^\mu\gamma^\nu\gamma^\rho\bar\gamma_\mu = 4\gamma^\nu\gamma^\rho - 2\bar\gamma^\nu\gamma^\rho + 2\bar\gamma^\rho\gamma^\nu, \tag{1.26}$$
$$\bar\gamma^\mu\hat\gamma^\nu\hat\gamma^\rho\bar\gamma_\mu = 4\hat\gamma^\nu\hat\gamma^\rho, \tag{1.27}$$
$$\hat\gamma^\mu\hat\gamma^\nu\hat\gamma^\rho\hat\gamma_\mu = (d-8)\hat\gamma^\nu\hat\gamma^\rho + 4\hat g^{\nu\rho}I, \tag{1.28}$$
$$\hat\gamma^\mu\gamma^\nu\gamma^\rho\hat\gamma_\mu = (d-4)\gamma^\nu\gamma^\rho - 2\hat\gamma^\nu\gamma^\rho + 2\hat\gamma^\rho\gamma^\nu, \tag{1.29}$$
$$\hat\gamma^\mu\bar\gamma^\nu\bar\gamma^\rho\hat\gamma_\mu = (d-4)\bar\gamma^\nu\bar\gamma^\rho, \tag{1.30}$$

$$\gamma^\mu\gamma^\nu\gamma^\rho\gamma^\sigma\gamma_\mu = -(d-4)\gamma^\nu\gamma^\rho\gamma^\sigma - 2\gamma^\sigma\gamma^\rho\gamma^\nu \tag{1.31}$$
$$\gamma^\mu\bar\gamma^\nu\bar\gamma^\rho\bar\gamma^\sigma\gamma_\mu = -(d-4)\bar\gamma^\nu\bar\gamma^\rho\bar\gamma^\sigma - 2\bar\gamma^\sigma\bar\gamma^\rho\bar\gamma^\nu \tag{1.32}$$
$$\gamma^\mu\hat\gamma^\nu\hat\gamma^\rho\hat\gamma^\sigma\gamma_\mu = -(d-4)\hat\gamma^\nu\hat\gamma^\rho\hat\gamma^\sigma - 2\hat\gamma^\sigma\hat\gamma^\rho\hat\gamma^\nu \tag{1.33}$$
$$\bar\gamma^\mu\bar\gamma^\nu\bar\gamma^\rho\bar\gamma^\sigma\bar\gamma_\mu = -2\bar\gamma^\sigma\bar\gamma^\rho\bar\gamma^\nu \tag{1.34}$$
$$\bar\gamma^\mu\gamma^\nu\gamma^\rho\gamma^\sigma\bar\gamma_\mu = -4\gamma^\nu\gamma^\rho\gamma^\sigma + 2\bar\gamma^\nu\gamma^\rho\gamma^\sigma - 2\bar\gamma^\rho\gamma^\nu\gamma^\sigma + 2\bar\gamma^\sigma\gamma^\nu\gamma^\rho, \tag{1.35}$$
$$\bar\gamma^\mu\hat\gamma^\nu\hat\gamma^\rho\hat\gamma^\sigma\bar\gamma_\mu = -4\hat\gamma^\nu\hat\gamma^\rho\hat\gamma^\sigma, \tag{1.36}$$
$$\hat\gamma^\mu\hat\gamma^\nu\hat\gamma^\rho\hat\gamma^\sigma\hat\gamma_\mu = -(d-8)\hat\gamma^\nu\hat\gamma^\rho\hat\gamma^\sigma - 2\hat\gamma^\sigma\hat\gamma^\rho\hat\gamma^\nu \tag{1.37}$$
$$\hat\gamma^\mu\gamma^\nu\gamma^\rho\gamma^\sigma\hat\gamma_\mu = -(d-4)\gamma^\nu\gamma^\rho\gamma^\sigma + 2\hat\gamma^\nu\gamma^\rho\gamma^\sigma - 2\hat\gamma^\rho\gamma^\nu\gamma^\sigma + 2\hat\gamma^\sigma\gamma^\nu\gamma^\rho, \tag{1.38}$$

$$\hat{\gamma}^\mu\bar{\gamma}^\nu\bar{\gamma}^\rho\bar{\gamma}^\sigma\hat{\gamma}_\mu = -(d-4)\bar{\gamma}^\nu\bar{\gamma}^\rho\bar{\gamma}^\sigma, \tag{1.39}$$

$$\gamma^\mu\gamma^\nu\gamma^\rho\gamma^\sigma\gamma^\tau\gamma_\mu = (d-4)\gamma^\nu\gamma^\rho\gamma^\sigma\gamma^\tau + 2\gamma^\sigma\gamma^\rho\gamma^\nu\gamma^\tau + 2\gamma^\tau\gamma^\nu\gamma^\rho\gamma^\sigma \tag{1.40}$$
$$\gamma^\mu\bar{\gamma}^\nu\bar{\gamma}^\rho\bar{\gamma}^\sigma\bar{\gamma}^\tau\gamma_\mu = (d-4)\bar{\gamma}^\nu\bar{\gamma}^\rho\bar{\gamma}^\sigma\bar{\gamma}^\tau + 2\bar{\gamma}^\sigma\bar{\gamma}^\rho\bar{\gamma}^\nu\bar{\gamma}^\tau + 2\bar{\gamma}^\tau\bar{\gamma}^\nu\bar{\gamma}^\rho\bar{\gamma}^\sigma \tag{1.41}$$
$$\gamma^\mu\hat{\gamma}^\nu\hat{\gamma}^\rho\hat{\gamma}^\sigma\hat{\gamma}^\tau\gamma_\mu = (d-4)\hat{\gamma}^\nu\hat{\gamma}^\rho\hat{\gamma}^\sigma\hat{\gamma}^\tau + 2\hat{\gamma}^\sigma\hat{\gamma}^\rho\hat{\gamma}^\nu\hat{\gamma}^\tau + 2\hat{\gamma}^\tau\hat{\gamma}^\nu\hat{\gamma}^\rho\hat{\gamma}^\sigma \tag{1.42}$$
$$\bar{\gamma}^\mu\bar{\gamma}^\nu\bar{\gamma}^\rho\bar{\gamma}^\sigma\bar{\gamma}^\tau\bar{\gamma}_\mu = 2\bar{\gamma}^\sigma\bar{\gamma}^\rho\bar{\gamma}^\nu\bar{\gamma}^\tau + 2\bar{\gamma}^\tau\bar{\gamma}^\nu\bar{\gamma}^\rho\bar{\gamma}^\sigma \tag{1.43}$$
$$\bar{\gamma}^\mu\gamma^\nu\gamma^\rho\gamma^\sigma\gamma^\tau\bar{\gamma}_\mu = 4\gamma^\nu\gamma^\rho\gamma^\sigma\gamma^\tau \tag{1.44}$$
$$- 2\bar{\gamma}^\nu\gamma^\rho\gamma^\sigma\gamma^\tau + 2\bar{\gamma}^\rho\gamma^\nu\gamma^\sigma\gamma^\tau - 2\bar{\gamma}^\sigma\gamma^\nu\gamma^\rho\gamma^\tau + 2\bar{\gamma}^\tau\gamma^\nu\gamma^\rho\gamma^\sigma, \tag{1.45}$$
$$\bar{\gamma}^\mu\hat{\gamma}^\nu\hat{\gamma}^\rho\hat{\gamma}^\sigma\hat{\gamma}^\tau\bar{\gamma}_\mu = 4\hat{\gamma}^\nu\hat{\gamma}^\rho\hat{\gamma}^\sigma\hat{\gamma}^\tau, \tag{1.46}$$
$$\hat{\gamma}^\mu\hat{\gamma}^\nu\hat{\gamma}^\rho\hat{\gamma}^\sigma\hat{\gamma}^\tau\hat{\gamma}_\mu = (d-8)\hat{\gamma}^\nu\hat{\gamma}^\rho\hat{\gamma}^\sigma\hat{\gamma}^\tau + 2\hat{\gamma}^\sigma\hat{\gamma}^\rho\hat{\gamma}^\nu\hat{\gamma}^\tau + 2\hat{\gamma}^\tau\hat{\gamma}^\nu\hat{\gamma}^\rho\hat{\gamma}^\sigma \tag{1.47}$$
$$\hat{\gamma}^\mu\gamma^\nu\gamma^\rho\gamma^\sigma\gamma^\tau\hat{\gamma}_\mu = (d-4)\gamma^\nu\gamma^\rho\gamma^\sigma\gamma^\tau \tag{1.48}$$
$$- 2\hat{\gamma}^\nu\gamma^\rho\gamma^\sigma\gamma^\tau + 2\hat{\gamma}^\rho\gamma^\nu\gamma^\sigma\gamma^\tau - 2\hat{\gamma}^\sigma\gamma^\nu\gamma^\rho\gamma^\tau + 2\hat{\gamma}^\tau\gamma^\nu\gamma^\rho\gamma^\sigma, \tag{1.49}$$
$$\hat{\gamma}^\mu\bar{\gamma}^\nu\bar{\gamma}^\rho\bar{\gamma}^\sigma\bar{\gamma}^\tau\hat{\gamma}_\mu = (d-4)\bar{\gamma}^\nu\bar{\gamma}^\rho\bar{\gamma}^\sigma\bar{\gamma}^\tau, \tag{1.50}$$

$$\gamma^\mu\gamma^\nu\gamma^\rho\gamma^\sigma\gamma^\tau\gamma^\kappa\gamma_\mu = -(d-4)\gamma^\nu\gamma^\rho\gamma^\sigma\gamma^\tau\gamma^\kappa \tag{1.51}$$
$$- 2\gamma^\sigma\gamma^\rho\gamma^\nu\gamma^\tau\gamma^\kappa - 2\gamma^\tau\gamma^\nu\gamma^\rho\gamma^\sigma\gamma^\kappa + 2\gamma^\kappa\gamma^\nu\gamma^\rho\gamma^\sigma\gamma^\tau \tag{1.52}$$
$$\gamma^\mu\bar{\gamma}^\nu\bar{\gamma}^\rho\bar{\gamma}^\sigma\bar{\gamma}^\tau\bar{\gamma}^\kappa\gamma_\mu = -(d-4)\bar{\gamma}^\nu\bar{\gamma}^\rho\bar{\gamma}^\sigma\bar{\gamma}^\tau\bar{\gamma}^\kappa \tag{1.53}$$
$$- 2\bar{\gamma}^\sigma\bar{\gamma}^\rho\bar{\gamma}^\nu\bar{\gamma}^\tau\bar{\gamma}^\kappa - 2\bar{\gamma}^\tau\bar{\gamma}^\nu\bar{\gamma}^\rho\bar{\gamma}^\sigma\bar{\gamma}^\kappa + 2\bar{\gamma}^\kappa\bar{\gamma}^\nu\bar{\gamma}^\rho\bar{\gamma}^\sigma\bar{\gamma}^\tau \tag{1.54}$$
$$\gamma^\mu\hat{\gamma}^\nu\hat{\gamma}^\rho\hat{\gamma}^\sigma\hat{\gamma}^\tau\hat{\gamma}^\kappa\gamma_\mu = -(d-4)\hat{\gamma}^\nu\hat{\gamma}^\rho\hat{\gamma}^\sigma\hat{\gamma}^\tau\hat{\gamma}^\kappa \tag{1.55}$$
$$- 2\hat{\gamma}^\sigma\hat{\gamma}^\rho\hat{\gamma}^\nu\hat{\gamma}^\tau\hat{\gamma}^\kappa - 2\hat{\gamma}^\tau\hat{\gamma}^\nu\hat{\gamma}^\rho\hat{\gamma}^\sigma\hat{\gamma}^\kappa + 2\hat{\gamma}^\kappa\hat{\gamma}^\nu\hat{\gamma}^\rho\hat{\gamma}^\sigma\hat{\gamma}^\tau \tag{1.56}$$
$$\bar{\gamma}^\mu\bar{\gamma}^\nu\bar{\gamma}^\rho\bar{\gamma}^\sigma\bar{\gamma}^\tau\bar{\gamma}^\kappa\bar{\gamma}_\mu = 2\bar{\gamma}^\tau\bar{\gamma}^\sigma\bar{\gamma}^\rho\bar{\gamma}^\nu\bar{\gamma}^\kappa + 2\bar{\gamma}^\kappa\bar{\gamma}^\nu\bar{\gamma}^\rho\bar{\gamma}^\sigma\bar{\gamma}^\tau = -2\bar{\gamma}^\kappa\bar{\gamma}^\tau\bar{\gamma}^\sigma\bar{\gamma}^\rho\bar{\gamma}^\nu \tag{1.57}$$
$$\bar{\gamma}^\mu\gamma^\nu\gamma^\rho\gamma^\sigma\gamma^\tau\gamma^\kappa\bar{\gamma}_\mu = -4\gamma^\nu\gamma^\rho\gamma^\sigma\gamma^\tau\gamma^\kappa + 2\bar{\gamma}^\nu\gamma^\rho\gamma^\sigma\gamma^\tau\gamma^\kappa - 2\bar{\gamma}^\rho\gamma^\nu\gamma^\sigma\gamma^\tau\gamma^\kappa \tag{1.58}$$
$$+ 2\bar{\gamma}^\sigma\gamma^\nu\gamma^\rho\gamma^\tau\gamma^\kappa - 2\bar{\gamma}^\tau\gamma^\nu\gamma^\rho\gamma^\sigma\gamma^\kappa + 2\bar{\gamma}^\kappa\gamma^\nu\gamma^\rho\gamma^\sigma\gamma^\tau \tag{1.59}$$
$$\bar{\gamma}^\mu\hat{\gamma}^\nu\hat{\gamma}^\rho\hat{\gamma}^\sigma\hat{\gamma}^\tau\hat{\gamma}^\kappa\bar{\gamma}_\mu = -4\hat{\gamma}^\nu\hat{\gamma}^\rho\hat{\gamma}^\sigma\hat{\gamma}^\tau\hat{\gamma}^\kappa, \tag{1.60}$$
$$\hat{\gamma}^\mu\hat{\gamma}^\nu\hat{\gamma}^\rho\hat{\gamma}^\sigma\hat{\gamma}^\tau\hat{\gamma}^\kappa\hat{\gamma}_\mu = -(d-8)\hat{\gamma}^\nu\hat{\gamma}^\rho\hat{\gamma}^\sigma\hat{\gamma}^\tau\hat{\gamma}^\kappa \tag{1.61}$$
$$- 2\hat{\gamma}^\sigma\hat{\gamma}^\rho\hat{\gamma}^\nu\hat{\gamma}^\tau\hat{\gamma}^\kappa - 2\hat{\gamma}^\tau\hat{\gamma}^\nu\hat{\gamma}^\rho\hat{\gamma}^\sigma\hat{\gamma}^\kappa + 2\hat{\gamma}^\kappa\hat{\gamma}^\nu\hat{\gamma}^\rho\hat{\gamma}^\sigma\hat{\gamma}^\tau \tag{1.62}$$
$$\hat{\gamma}^\mu\gamma^\nu\gamma^\rho\gamma^\sigma\gamma^\tau\gamma^\kappa\hat{\gamma}_\mu = -(d-4)\gamma^\nu\gamma^\rho\gamma^\sigma\gamma^\tau\gamma^\kappa + 2\hat{\gamma}^\nu\gamma^\rho\gamma^\sigma\gamma^\tau\gamma^\kappa - 2\hat{\gamma}^\rho\gamma^\nu\gamma^\sigma\gamma^\tau\gamma^\kappa \tag{1.63}$$
$$+ 2\hat{\gamma}^\sigma\gamma^\nu\gamma^\rho\gamma^\tau\gamma^\kappa - 2\hat{\gamma}^\tau\gamma^\nu\gamma^\rho\gamma^\sigma\gamma^\kappa + 2\hat{\gamma}^\kappa\gamma^\nu\gamma^\rho\gamma^\sigma\gamma^\tau, \tag{1.64}$$
$$\hat{\gamma}^\mu\bar{\gamma}^\nu\bar{\gamma}^\rho\bar{\gamma}^\sigma\bar{\gamma}^\tau\bar{\gamma}^\kappa\hat{\gamma}_\mu = -(d-4)\bar{\gamma}^\nu\bar{\gamma}^\rho\bar{\gamma}^\sigma\bar{\gamma}^\tau\bar{\gamma}^\kappa, \tag{1.65}$$

Index contractions

If the first and the last matrix are in different dimensions, we can always write them as

$$\gamma^\mu\ldots\bar{\gamma}_\mu = \bar{\gamma}^\mu\ldots\gamma_\mu = \bar{\gamma}^\mu\ldots\bar{\gamma}_\mu, \tag{1.66}$$
$$\gamma^\mu\ldots\hat{\gamma}_\mu = \hat{\gamma}^\mu\ldots\gamma_\mu = \hat{\gamma}^\mu\ldots\hat{\gamma}_\mu, \tag{1.67}$$
$$\bar{\gamma}^\mu\ldots\hat{\gamma}_\mu = \hat{\gamma}^\mu\ldots\bar{\gamma}_\mu = 0. \tag{1.68}$$

For general index pairs we have that

$$\gamma^\mu \bar{\gamma}^{\nu_1} \dots \bar{\gamma}^{\nu_n} \gamma_\mu = \bar{\gamma}^\mu \bar{\gamma}^{\nu_1} \dots \bar{\gamma}^{\nu_n} \bar{\gamma}_\mu + (-1)^n (D-4) \bar{\gamma}^{\nu_1} \dots \bar{\gamma}^{\nu_n}, \tag{1.69}$$

$$\gamma^\mu \hat{\gamma}^{\nu_1} \dots \hat{\gamma}^{\nu_n} \gamma_\mu = \hat{\gamma}^\mu \hat{\gamma}^{\nu_1} \dots \hat{\gamma}^{\nu_n} \hat{\gamma}_\mu + 4(-1)^n \hat{\gamma}^{\nu_1} \dots \hat{\gamma}^{\nu_n}, \tag{1.70}$$

$$\bar{\gamma}^\mu \hat{\gamma}^{\nu_1} \dots \hat{\gamma}^{\nu_n} \bar{\gamma}_\mu = 4(-1)^n \hat{\gamma}^{\nu_1} \dots \hat{\gamma}^{\nu_n}, \tag{1.71}$$

$$\hat{\gamma}^\mu \bar{\gamma}^{\nu_1} \dots \bar{\gamma}^{\nu_n} \hat{\gamma}_\mu = (D-4)(-1)^n \bar{\gamma}^{\nu_1} \dots \bar{\gamma}^{\nu_n}, \tag{1.72}$$

This means that if we have a general formula for $\gamma^\mu \gamma^{\nu_1} \dots \gamma^{\nu_n} \gamma_\mu$ in D dimensions, we can easily obtain 7 of 9 possible combinations of dimensions. The other two cases are special and related with each other

$$\bar{\gamma}^\mu \gamma^{\nu_1} \dots \gamma^{\nu_n} \bar{\gamma}_\mu = \gamma^\mu \gamma^{\nu_1} \dots \gamma^{\nu_n} \gamma_\mu - \hat{\gamma}^\mu \gamma^{\nu_1} \dots \gamma^{\nu_n} \hat{\gamma}_\mu. \tag{1.73}$$

If we know $\bar{\gamma}^\mu \gamma^{\nu_1} \dots \gamma^{\nu_n} \bar{\gamma}_\mu$ we can easily compute $\hat{\gamma}^\mu \gamma^{\nu_1} \dots \gamma^{\nu_n} \hat{\gamma}_\mu$ and vice versa.

For purely four dimensional chains it is known that if the number of the Dirac matrices between the index pair is odd, then

$$\bar{\gamma}^\mu \bar{\gamma}^{\nu_1} \dots \bar{\gamma}^{\nu_{2n+1}} \bar{\gamma}_\mu = -2 \bar{\gamma}^{\nu_{2n+1}} \dots \bar{\gamma}^{\nu_1}. \tag{1.74}$$

This can be trivially generalized to even chains, i.e.

$$\bar{\gamma}^\mu \bar{\gamma}^{\nu_1} \dots \bar{\gamma}^{\nu_{2n+1}} \bar{\gamma}^\rho \bar{\gamma}_\mu = 2 \bar{\gamma}^{\nu_{2n+1}} \dots \bar{\gamma}^{\nu_1} \bar{\gamma}^\rho + 2 \bar{\gamma}^\rho \bar{\gamma}^{\nu_1} \dots \bar{\gamma}^{\nu_{2n+1}} \tag{1.75}$$

Following formula for $\gamma^\mu \gamma^{\nu_1} \dots \gamma^{\nu_n} \gamma_\mu$ in d dimensions and $n \geq 3$ is given in Veltman's Gammatrica

$$\gamma^\mu \gamma^{\nu_1} \dots \gamma^{\nu_n} \gamma_\mu = (d-4)(-1)^n \gamma^{\nu_1} \dots \gamma^{\nu_n} + 2(-1)^n \gamma^{\nu_3} \gamma^{\nu_2} \gamma^{\nu_1} \gamma^{\nu_4} \dots \gamma^{\nu_n} \tag{1.76}$$

$$+ 2 \sum_{j=4}^{m} (-1)^{n-j} \gamma^{\nu_j} \gamma^{\nu_1} \dots \gamma^{\nu_{j-1}} \gamma^{\nu_{j+1}} \dots \gamma^{\nu_n}. \tag{1.77}$$

Another useful and more compact formula for the same expression with $n \geq 2$ was derived my R. Mertig

$$\gamma^\mu \gamma^{\nu_1} \dots \gamma^{\nu_n} \gamma_\mu = (-1)^n \Bigg\{ (d-2n) \gamma^{\nu_1} \dots \gamma^{\nu_n} \tag{1.78}$$

$$- 4 \sum_{i=1}^{l-1} \sum_{j=i+1}^{l} (-1)^{j-i} \gamma^{\nu_1} \dots \gamma^{\nu_{i-1}} \gamma^{\nu_{i+1}} \dots \gamma^{\nu_{j-1}} \gamma^{\nu_{j+1}} \dots \gamma^{\nu_n} g_{\mu_i \mu_j} \Bigg\} \tag{1.79}$$

2 Basic objects

2.1 Abbreviation

Abbreviation is a function used by **OneLoop** and **PaVeReduce** for generating smaller files when saving results to the hard disk. The convention is that a definition like **GP = GluonPropagator** should be accompanied by the definition **Abbreviation[GluonPropagator] = HoldForm[GP]**.

2.1.1 See also

Overview, $Abbreviations, OneLoop, PaVeReduce, WriteOut, WriteOutPaVe, GluonPropagator, GluonVertex, QuarkPropagator.

2.1.2 Examples

```
GP[p, {\[Mu], a}, {\[Nu], b}]
```

$$\Pi^{\mu\nu}_{ab}(p)$$

2.2 AntiQuarkField

AntiQuarkField is the name of a fermionic field. **AntiQuarkField** is just a name with no functional properties. Only typesetting rules are attached.

2.2.1 See also

Overview, QuantumField, QuarkField.

2.2.2 Examples

```
AntiQuarkField
```

$$\bar{\psi}$$

2.3 QuarkField

QuarkField is the name of a fermionic field. This is just a name with no functional properties. Only typesetting rules are attached.

2.3.1 See also

Overview, AntiQuarkField, QuantumField.

2.3.2 Examples

```
QuarkField
```

$$\psi$$

2.4 QuarkFieldPsi

QuarkFieldPsi is the name of a fermionic field.This is just a name with no functional properties. Only typesetting rules are attached.

2.4.1 See also

Overview

2.4.2 Examples

```
QuarkFieldPsi
```

$$\psi$$

```
QuantumField[QuarkFieldPsiDagger] . GA[\[Mu]] . CovariantD[\[Mu]] .
QuantumField[QuarkFieldPsi]

ExpandPartialD[%]
```

$$\psi^{\dagger}.\bar{\gamma}^{\mu}.D_{\mu}.\psi$$

$$\bar{\gamma}^{\mu}\psi^{\dagger}.((\partial_{\mu}\psi)) - iT^{\text{c19}}g_s\bar{\gamma}^{\mu}\psi^{\dagger}.A_{\mu}^{\text{c19}}.\psi$$

2.5 QuarkFieldPsiDagger

QuarkFieldPsiDagger is the name of a fermionic field.This is just a name with no functional properties. Only typesetting rules are attached.

2.5.1 See also

Overview

2.5.2 Examples

```
QuarkFieldPsiDagger
```

$$\psi^{\dagger}$$

```
QuantumField[QuarkFieldPsiDagger] . GA[\[Mu]] . CovariantD[\[Mu]] .
QuantumField[QuarkFieldPsi]

ExpandPartialD[%]
```

$$\psi^{\dagger}.\bar{\gamma}^{\mu}.D_{\mu}.\psi$$

$$\bar{\gamma}^{\mu}\psi^{\dagger}.((\partial_{\mu}\psi)) - iT^{c19}g_s\bar{\gamma}^{\mu}\psi^{\dagger}.A_{\mu}^{c19}.\psi$$

2.6 QuarkFieldChi

QuarkFieldChi is the name of a fermionic field. This is just a name with no functional properties. Only typesetting rules are attached.

2.6.1 See also

Overview

2.6.2 Examples

```
QuarkFieldChi
```

$$\chi$$

```
QuantumField[QuarkFieldChiDagger] . GA[\[Mu]] . CovariantD[\[Mu]] .
QuantumField[QuarkFieldChi]

ExpandPartialD[%]
```

$$\chi^{\dagger}.\bar{\gamma}^{\mu}.D_{\mu}.\chi$$

$$\bar{\gamma}^{\mu}\chi^{\dagger}.((\partial_{\mu}\chi)) - iT^{\text{c19}}g_s\bar{\gamma}^{\mu}\chi^{\dagger}.A_{\mu}^{\text{c19}}.\chi$$

2.7 QuarkFieldChiDagger

QuarkFieldChiDagger is the name of a fermionic field. This is just a name with no functional properties. Only typesetting rules are attached.

2.7.1 See also

Overview

2.7.2 Examples

```
QuarkFieldChiDagger
```

$$\chi^{\dagger}$$

```
QuantumField[QuarkFieldChiDagger] . GA[\[Mu]] . CovariantD[\[Mu]] .
QuantumField[QuarkFieldChi]

ExpandPartialD[%]
```

$$\chi^{\dagger}.\bar{\gamma}^{\mu}.D_{\mu}.\chi$$

$$\bar{\gamma}^{\mu}\chi^{\dagger}.((\partial_{\mu}\chi)) - iT^{\text{c19}}g_s\bar{\gamma}^{\mu}\chi^{\dagger}.A_{\mu}^{\text{c19}}.\chi$$

2.8 CA

CA is one of the Casimir operator eigenvalues of $SU(N)$ (**CA** $= N$).

2.8.1 See also

Overview, CF, SUNSimplify.

2.8.2 Examples

```
CA
```

$$C_A$$

```
SUNSimplify[CA, SUNNToCACF -> False]
```

$$N$$

```
SUNN
```

$$N$$

2.9 CF

CF is one of the Casimir operator eigenvalues of $SU(N)$ (CF = $\frac{N^2-1}{2N}$).

2.9.1 See also

Overview, CA, SUNSimplify.

2.9.2 Examples

```
CF
```

$$C_F$$

```
SUNSimplify[CF, SUNNToCACF -> False]
```

$$\frac{N^2-1}{2N}$$

```
SUNN
```

$$N$$

```
SUNSimplify[SUNN^2 - 1, SUNNToCACF -> True]
```

$$2C_A C_F$$

2.10 CGA

CGA[i] can be used as input for γ^i in 4 dimensions, where **i** is a Cartesian index, and is transformed into **DiracGamma[CartesianIndex[i]]** by **FeynCalcInternal**.

2.10.1 See also

Overview, GA, DiracGamma.

2.10.2 Examples

```
CGA[i]
```

$$\overline{\gamma}^i$$

```
CGA[i, j] - CGA[j, i]
```

$$\overline{\gamma}^i.\overline{\gamma}^j - \overline{\gamma}^j.\overline{\gamma}^i$$

```
StandardForm[FCI[CGA[i]]]

(*DiracGamma[CartesianIndex[i]]*)
```

```
CGA[i, j, k, l]
```

$$\overline{\gamma}^i.\overline{\gamma}^j.\overline{\gamma}^k.\overline{\gamma}^l$$

```
StandardForm[CGA[i, j, k, l]]

(*CGA[i] . CGA[j] . CGA[k] . CGA[l]*)
```

```
DiracSimplify[DiracTrace[CGA[i, j, k, l]]]
```

$$4\bar{\delta}^{il}\bar{\delta}^{jk} - 4\bar{\delta}^{ik}\bar{\delta}^{jl} + 4\bar{\delta}^{ij}\bar{\delta}^{kl}$$

```
CGA[i] . (CGS[p] + m) . CGA[j]
```

$$\overline{\gamma}^{i} . (\overline{\gamma} \cdot \overline{p} + m) . \overline{\gamma}^{j}$$

2.11 CGS

CGS[p] is transformed into **DiracGamma[CartesianMomentum[p]]** by **FeynCalcInternal**.

CGS[p, q, ...] is equivalent to **CGS[p].CGS[q]**.

2.11.1 See also

Overview, GS, DiracGamma.

2.11.2 Examples

```
CGS[p]
```

$$\overline{\gamma} \cdot \overline{p}$$

```
CGS[p] // FCI // StandardForm

(*DiracGamma[CartesianMomentum[p]]*)
```

```
CGS[p, q, r, s]
```

$$(\overline{\gamma} \cdot \overline{p}) . (\overline{\gamma} \cdot \overline{q}) . (\overline{\gamma} \cdot \overline{r}) . (\overline{\gamma} \cdot \overline{s})$$

```
CGS[p, q, r, s] // StandardForm

(*CGS[p] . CGS[q] . CGS[r] . CGS[s]*)
```

```
CGS[q] . (CGS[p] + m) . CGS[q]
```

$$(\overline{\gamma}\cdot\overline{q}).(\overline{\gamma}\cdot\overline{p}+m).(\overline{\gamma}\cdot\overline{q})$$

2.12 CGAD

CGAD[i] can be used as input for γ^i in D dimensions, where **i** is a Cartesian index, and is transformed into **DiracGamma[CartesianIndex[i, D-1], D]** by **FeynCalcInternal**.

2.12.1 See also

Overview, GAD, DiracGamma.

2.12.2 Examples

```
CGAD[i]
```

$$\gamma^i$$

```
CGAD[i, j] - CGAD[j, i]
```

$$\gamma^i.\gamma^j-\gamma^j.\gamma^i$$

```
StandardForm[FCI[CGAD[i]]]

(*DiracGamma[CartesianIndex[i, -1 + D], D]*)
```

```
CGAD[i, j, k, l]
```

$$\gamma^i.\gamma^j.\gamma^k.\gamma^l$$

```
StandardForm[CGAD[i, j, k, l]]

(*CGAD[i] . CGAD[j] . CGAD[k] . CGAD[l]*)
```

```
DiracSimplify[DiracTrace[CGAD[i, j, k, l]]]
```

$$4\delta^{il}\delta^{jk} - 4\delta^{ik}\delta^{jl} + 4\delta^{ij}\delta^{kl}$$

```
CGAD[i] . (CGSD[p] + m) . CGAD[j]
```

$$\gamma^i.(m + \gamma \cdot p).\gamma^j$$

2.13 CGSD

CGSD[p] is transformed into **DiracGamma[CartesianMomentum[p, D-1], D]** by **FeynCalcInternal**.

CGSD[p, q, ...] is equivalent to **CGSD[p].CGSD[q]**.

2.13.1 See also

Overview, GSD, DiracGamma.

2.13.2 Examples

```
CGSD[p]
```

$$\gamma \cdot p$$

```
CGSD[p] // FCI // StandardForm

(*DiracGamma[CartesianMomentum[p, -1 + D], D]*)
```

```
CGSD[p, q, r, s]
```

$$(\gamma \cdot p).(\gamma \cdot q).(\gamma \cdot r).(\gamma \cdot s)$$

```
CGSD[p, q, r, s] // StandardForm

(*CGSD[p] . CGSD[q] . CGSD[r] . CGSD[s]*)
```

```
CGSD[q] . (CGSD[p] + m) . CGSD[q]
```

$$(\gamma \cdot q).(m + \gamma \cdot p).(\gamma \cdot q)$$

2.14 CGAE

CGAE[i] can be used as input for γ^i in $D-4$ dimensions, where **i** is a Cartesian index, and is transformed into **DiracGamma[CartesianIndex[i, D-4], D-4]** by **FeynCalcInternal**.

2.14.1 See also

Overview, GAE, DiracGamma.

2.14.2 Examples

```
CGAE[i]
```

$$\hat{\gamma}^i$$

```
CGAE[i, j] - CGAE[j, i]
```

$$\hat{\gamma}^i.\hat{\gamma}^j - \hat{\gamma}^j.\hat{\gamma}^i$$

```
StandardForm[FCI[CGAE[i]]]

(*DiracGamma[CartesianIndex[i, -4 + D], -4 + D]*)
```

```
CGAE[i, j, k, l]
```

$$\hat{\gamma}^i.\hat{\gamma}^j.\hat{\gamma}^k.\hat{\gamma}^l$$

```
StandardForm[CGAE[i, j, k, l]]

(*CGAE[i] . CGAE[j] . CGAE[k] . CGAE[l]*)
```

2.15 CGSE

CGSE[p] is transformed into **DiracGamma[CartesianMomentum[p, D-4], D-4]** by FeynCalcInternal.

CGSE[p, q, ...] is equivalent to **CGSE[p].CGSE[q]**.

2.15.1 See also

Overview, GSE, DiracGamma.

2.15.2 Examples

```
CGSE[p]
```

$$\hat{\gamma} \cdot \hat{p}$$

```
CGSE[p] // FCI // StandardForm

(*DiracGamma[CartesianMomentum[p, -4 + D], -4 + D]*)
```

```
CGSE[p, q, r, s]
```

$$(\hat{\gamma} \cdot \hat{p}) . (\hat{\gamma} \cdot \hat{q}) . (\hat{\gamma} \cdot \hat{r}) . (\hat{\gamma} \cdot \hat{s})$$

```
CGSE[p, q, r, s] // StandardForm

(*CGSE[p] . CGSE[q] . CGSE[r] . CGSE[s]*)
```

```
CGSE[q] . (CGSE[p] + m) . CGSE[q]
```

$$(\hat{\gamma} \cdot \hat{q}) . (m + \hat{\gamma} \cdot \hat{p}) . (\hat{\gamma} \cdot \hat{q})$$

2.16 CSI

CSI[i] can be used as input for 3-dimensional σ^i with 3-dimensional Cartesian index **i** and is transformed into **PauliSigma[CartesianIndex[i]]** by **FeynCalcInternal**.

2.16.1 See also

Overview, PauliSigma.

2.16.2 Examples

```
CSI[i]
```

$$\overline{\sigma}^i$$

```
CSI[i, j] - CSI[j, i]
```

$$\overline{\sigma}^i.\overline{\sigma}^j - \overline{\sigma}^j.\overline{\sigma}^i$$

```
StandardForm[FCI[CSI[i]]]

(*PauliSigma[CartesianIndex[i]]*)
```

```
CSI[i, j, k, l]
```

$$\overline{\sigma}^i.\overline{\sigma}^j.\overline{\sigma}^k.\overline{\sigma}^l$$

```
StandardForm[CSI[i, j, k, l]]

(*CSI[i] . CSI[j] . CSI[k] . CSI[l]*)
```

2.17 CSID

CSID[i] can be used as input for $D-1$-dimensional σ^i with $D-1$-dimensional Cartesian index **i** and is transformed into **PauliSigma[CartesianIndex[i, D-1], D-1]** by **FeynCalcInternal**.

2.17.1 See also

Overview, PauliSigma.

2.17.2 Examples

```
CSID[i]
```

$$\sigma^i$$

```
CSID[i, j] - CSID[j, i]
```

$$\sigma^i.\sigma^j - \sigma^j.\sigma^i$$

```
StandardForm[FCI[CSID[i]]]

(*PauliSigma[CartesianIndex[i, -1 + D], -1 + D]*)
```

```
CSID[i, j, k, l]
```

$$\sigma^i.\sigma^j.\sigma^k.\sigma^l$$

```
StandardForm[CSID[i, j, k, l]]

(*CSID[i] . CSID[j] . CSID[k] . CSID[l]*)
```

2.18 CSIE

`CSIE[i]` can be used as input for $D-4$-dimensional σ^i with $D-4$-dimensional Cartesian index **`i`** and is transformed into **`PauliSigma[CartesianIndex[i, D-4], D-4]`** by **`FeynCalcInternal`**.

2.18.1 See also

Overview, PauliSigma.

2.18.2 Examples

```
CSIE[i]
```

$$\hat{\sigma}^i$$

```
CSIE[i, j] - CSIE[j, i]
```

$$\hat{\sigma}^i.\hat{\sigma}^j - \hat{\sigma}^j.\hat{\sigma}^i$$

```
StandardForm[FCI[CSIE[i]]]

(*PauliSigma[CartesianIndex[i, -4 + D], -4 + D]*)
```

```
CSIE[i, j, k, l]
```

$$\hat{\sigma}^i.\hat{\sigma}^j.\hat{\sigma}^k.\hat{\sigma}^l$$

```
StandardForm[CSIE[i, j, k, l]]

(*CSIE[i] . CSIE[j] . CSIE[k] . CSIE[l]*)
```

2.19 CSIS

CSIS[p]can be used as input for 3-dimensional $\sigma^i p^i$ with 3-dimensional Cartesian vector p and is transformed into PauliSigma[CartesianMomentum[p]] by FeynCalcInternal.

2.19.1 See also

Overview, PauliSigma.

2.19.2 Examples

```
CSIS[p]
```

$$\overline{\sigma} \cdot \overline{p}$$

```
CSIS[p] // FCI // StandardForm

(*PauliSigma[CartesianMomentum[p]]*)
```

```
CSIS[p, q, r, s]
```

$$(\overline{\sigma}\cdot\overline{p}).(\overline{\sigma}\cdot\overline{q}).(\overline{\sigma}\cdot\overline{r}).(\overline{\sigma}\cdot\overline{s})$$

```
CSIS[p, q, r, s] // StandardForm

(*CSIS[p] . CSIS[q] . CSIS[r] . CSIS[s]*)
```

2.20 CSISD

CSISD[p] can be used as input for D-1-dimensional $\sigma^i p^i$ with D-1-dimensional Cartesian vector p and is transformed into PauliSigma[CartesianMomentum[p,D-1],D-1] by FeynCalcInternal.

2.20.1 See also

Overview, PauliSigma.

2.20.2 Examples

```
CSISD[p]
```

$$\sigma\cdot p$$

```
CSISD[p] // FCI // StandardForm

(*PauliSigma[CartesianMomentum[p, -1 + D], -1 + D]*)
```

```
CSISD[p, q, r, s]
```

$$(\sigma\cdot p).(\sigma\cdot q).(\sigma\cdot r).(\sigma\cdot s)$$

```
CSISD[p, q, r, s] // StandardForm

(*CSISD[p] . CSISD[q] . CSISD[r] . CSISD[s]*)
```

2.21 CSISE

CSISE[p] can be used as input for D-4-dimensional $\sigma^i p^i$ with $D-4$-dimensional Cartesian vector p and is transformed into **PauliSigma[CartesianMomentum[p, D-4], D-4]** by FeynCalcInternal.

2.21.1 See also

Overview, PauliSigma.

2.21.2 Examples

```
CSISE[p]
```

$$\hat{\sigma} \cdot \hat{p}$$

```
CSISE[p] // FCI // StandardForm

(*PauliSigma[CartesianMomentum[p, -4 + D], -4 + D]*)
```

```
CSISE[p, q, r, s]
```

$$(\hat{\sigma} \cdot \hat{p}) . (\hat{\sigma} \cdot \hat{q}) . (\hat{\sigma} \cdot \hat{r}) . (\hat{\sigma} \cdot \hat{s})$$

```
CSISE[p, q, r, s] // StandardForm

(*CSISE[p] . CSISE[q] . CSISE[r] . CSISE[s]*)
```

2.22 CSP

CSP[p, q] is the 3-dimensional scalar product of **p** with **q** and is transformed into **CartesianPair[CartesianMomentum[p], CartesianMomentum[q]]** by **FeynCalcInternal**.

CSP[p] is the same as **CSP[p, p]** ($= p^2$).

2.22.1 See also

Overview, SP, ScalarProduct, CartesianScalarProduct.

2.22.2 Examples

```
CSP[p, q] + CSP[q]
```

$$\overline{p} \cdot \overline{q} + \overline{q}^2$$

```
CSP[p - q, q + 2 p]
```

$$(\overline{p} - \overline{q}) \cdot (2\overline{p} + \overline{q})$$

```
Calc[ CSP[p - q, q + 2 p] ]
```

$$-\overline{p} \cdot \overline{q} + 2\overline{p}^2 - \overline{q}^2$$

```
ExpandScalarProduct[CSP[p - q]]
```

$$-2\,(\overline{p} \cdot \overline{q}) + \overline{p}^2 + \overline{q}^2$$

```
CSP[a, b] // StandardForm

(*CSP[a, b]*)
```

```
CSP[a, b] // FCI // StandardForm

(*CartesianPair[CartesianMomentum[a], CartesianMomentum[b]]*)
```

```
CSP[a, b] // FCI // FCE // StandardForm

(*CSP[a, b]*)
```

2.23 CSPD

CSPD[p, q] is the $D-1$-dimensional scalar product of **p** with **q** and is transformed into **CartesianPair[CartesianMomentum[p, D-1], CartesianMomentum[q, D-1]]** by **FeynCalcInternal**.

CSPD[p] is the same as **CSPD[p, p]** ($= p^2$).

2.23.1 See also

Overview, SPD, ScalarProduct, CartesianScalarProduct.

2.23.2 Examples

```
CSPD[p, q] + CSPD[q]
```

$$p \cdot q + q^2$$

```
CSPD[p - q, q + 2 p]
```

$$(p - q) \cdot (2p + q)$$

```
Calc[ CSPD[p - q, q + 2 p] ]
```

$$-p \cdot q + 2p^2 - q^2$$

```
ExpandScalarProduct[CSPD[p - q]]
```

$$-2(p \cdot q) + p^2 + q^2$$

```
CSPD[a, b] // StandardForm

(*CSPD[a, b]*)
```

```
CSPD[a, b] // FCI // StandardForm

(*CartesianPair[CartesianMomentum[a, -1 + D], CartesianMomentum[b, -1 +
D]]*)
```

```
CSPD[a, b] // FCI // FCE // StandardForm

(*CSPD[a, b]*)
```

```
FCE[ChangeDimension[CSP[p, q], D]] // StandardForm

(*CSPD[p, q]*)
```

2.24 CSPE

CSPE[p, q] is the $D-4$-dimensional scalar product of **p** with **q** and is transformed into **CartesianPair[CartesianMomentum[p, D-4], CartesianMomentum[q, D-4]]** by **FeynCalcInternal**.

CSPE[p] is the same as **CSPE[p, p]** ($= p^2$).

2.24.1 See also

Overview, SPE, ScalarProduct, CartesianScalarProduct.

2.24.2 Examples

```
CSPE[p, q] + CSPE[q]
```

$$\hat{p} \cdot \hat{q} + \hat{q}^2$$

```
CSPE[p - q, q + 2 p]
```

$$(\hat{p} - \hat{q}) \cdot (2\hat{p} + \hat{q})$$

```
Calc[ CSPE[p - q, q + 2 p] ]
```

$$-\hat{p} \cdot \hat{q} + 2\hat{p}^2 - \hat{q}^2$$

```
ExpandScalarProduct[CSPE[p - q]]
```

$$-2\,(\hat{p} \cdot \hat{q}) + \hat{p}^2 + \hat{q}^2$$

```
CSPE[a, b] // StandardForm

(*CSPE[a, b]*)
```

```
CSPE[a, b] // FCI // StandardForm

(*CartesianPair[CartesianMomentum[a, -4 + D], CartesianMomentum[b, -4 +
D]]*)
```

```
CSPE[a, b] // FCI // FCE // StandardForm

(*CSPE[a, b]*)
```

```
FCE[ChangeDimension[CSP[p, q], D - 4]] // StandardForm

(*CSPE[p, q]*)
```

2.25 CV

CV[p, i] is a 3-dimensional Cartesian vector and is transformed into **CartesianPair[CartesianMomentum[p], CartesianIndex[i]]** by **FeynCalcInternal**.

2.25.1 See also

Overview, FV, Pair, CartesianPair.

2.25.2 Examples

```
CV[p, i]
```

$$\overline{p}^i$$

```
CV[p - q, i]
```

$$(\overline{p} - \overline{q})^i$$

```
FCI[CV[p, i]] // StandardForm

(*CartesianPair[CartesianIndex[i], CartesianMomentum[p]]*)
```

ExpandScalarProduct is used to expand momenta in **CV**

```
ExpandScalarProduct[CV[p - q, i]]
```

$$\overline{p}^i - \overline{q}^i$$

2.26 CVD

CVD[p, i] is a $D-1$-dimensional Cartesian vector and is transformed into **CartesianPair[CartesianMomentum[p, D], CartesianIndex[i, D]]** by **FeynCalcInternal**.

2.26.1 See also

Overview, FVD, Pair, CartesianPair.

2.26.2 Examples

```
CVD[p, i]
```

$$p^i$$

```
CVD[p - q, i]
```

$$(p-q)^i$$

```
FCI[CVD[p, i]] // StandardForm

(*CartesianPair[CartesianIndex[i, -1 + D], CartesianMomentum[p, -1 + D]]*)
```

ExpandScalarProduct is used to expand momenta in **CVD**

```
ExpandScalarProduct[CVD[p - q, i]]
```

$$p^i - q^i$$

2.27 CVE

CVE[p, i] is a $D-4$-dimensional Cartesian vector and is transformed into **CartesianPair[CartesianMomentum[p, D-4], CartesianIndex[i, D-4]]** by **FeynCalcInternal**.

2.27.1 See also

Overview, FVE, Pair, CartesianPair.

2.27.2 Examples

```
CVE[p, i]
```

$$\hat{p}^i$$

```
CVE[p - q, i]
```

$$(\hat{p}-\hat{q})^i$$

```
FCI[CVE[p, i]] // StandardForm

(*CartesianPair[CartesianIndex[i, -4 + D], CartesianMomentum[p, -4 + D]]*)
```

ExpandScalarProduct is used to expand momenta in **CVE**

```
ExpandScalarProduct[CVE[p - q, i]]
```

$$\hat{p}^i-\hat{q}^i$$

2.28 CartesianIndex

CartesianIndex is the head of Cartesian indices. The internal representation of a 3-dimensional **i** is **CartesianIndex[i]**.

For other than three dimensions: **CartesianIndex[i, Dimension]**.

CartesianIndex[i, 3] simplifies to **CartesianIndex[i]**. The first argument cannot be an integer.

2.28.1 See also

Overview, LorentzIndex, ExplicitLorentzIndex.

2.28.2 Examples

This denotes a 3-dimensional Cartesian index.

```
CartesianIndex[i]
```

$$i$$

An optional second argument can be given for a dimension different from 3.

```
CartesianIndex[i, D - 1]
```

$$i$$

```
CartesianIndex[i, D - 4]
```

$$i$$

2.29 CartesianMomentum

`CartesianMomentum[p]` is the head of a 3-momentum **`p`**. The internal representation of a 3-dimensional **`p`** is **`CartesianMomentum[p]`**. For other than three dimensions: **`CartesianMomentum[p, Dimension]`**. **`CartesianMomentum[p, 3]`** simplifies to **`CartesianMomentum[p]`**.

2.29.1 See also

Overview, Momentum, TemporalMomentum.

2.29.2 Examples

This is a 3-dimensional momentum

```
CartesianMomentum[p]
```

$$\overline{p}$$

As an optional second argument the dimension must be specified if it is different from 3

```
CartesianMomentum[p, D - 1]
```

$$p$$

The dimension index is suppressed in the output.

```
CartesianMomentum[p, d - 1]
```

$$p$$

```
a1 = CartesianMomentum[-q]
```

$$-\overline{q}$$

```
a1 // StandardForm

(*-CartesianMomentum[q]*)
```

```
a2 = CartesianMomentum[p - q] + CartesianMomentum[2 q]
```

$$(\overline{p} - \overline{q}) + 2\overline{q}$$

```
a2 // StandardForm

(*CartesianMomentum[p - q] + 2 CartesianMomentum[q]*)
```

```
a2 // MomentumExpand // StandardForm

(*CartesianMomentum[p] + CartesianMomentum[q]*)
```

```
a2 // MomentumCombine // StandardForm

(*CartesianMomentum[p + q]*)
```

Notice that when changing the dimension, one must specify its value as if the the 3-vector were the spatial component of the corresponding 4-vector

```
ChangeDimension[CartesianMomentum[p], d - 1] // StandardForm

(*CartesianMomentum[p, -2 + d]*)
```

```
ChangeDimension[CartesianMomentum[p], d] // StandardForm

(*CartesianMomentum[p, -1 + d]*)
```

```
Clear[a1, a2]
```

2.30 CartesianPair

CartesianPair[a, b] is a special pairing used in the internal representation. **a** and **b** may have heads **CartesianIndex** or **CartesianMomentum**. If both **a** and **b** have head **CartesianIndex**, the Kronecker delta is understood. If **a** and **b** have head **CartesianMomentum**, a Cartesian scalar product is meant. If one of **a** and **b** has head **CartesianIndex** and the other **CartesianMomentum**, a Cartesian vector p^i is understood.

2.30.1 See also

Overview, Pair, TemporalPair.

2.30.2 Examples

This represents a three-dimensional Kronecker delta

```
CartesianPair[CartesianIndex[i], CartesianIndex[j]]
```

$$\bar{\delta}^{i\,j}$$

This is a $D-1$-dimensional Kronecker delta

```
CartesianPair[CartesianIndex[i, D - 1], CartesianIndex[j, D - 1]]
```

$$\delta^{i\,j}$$

If the Cartesian indices live in different dimensions, this gets resolved according to the t'Hoft-Veltman-Breitenlohner-Maison prescription

```
CartesianPair[CartesianIndex[i, D - 1], CartesianIndex[j]]
```

$$\bar{\delta}^{i\,j}$$

```
CartesianPair[CartesianIndex[i, D - 1], CartesianIndex[j, D - 4]]
```

$$\hat{\delta}^{i\,j}$$

```
CartesianPair[CartesianIndex[i], CartesianIndex[j, D - 4]]
```

$$0$$

A 3-dimensional Cartesian vector

```
CartesianPair[CartesianIndex[i], CartesianMomentum[p]]
```

$$\overline{p}^{i}$$

A $D-1$-dimensional Cartesian vector

```
CartesianPair[CartesianIndex[i, D - 1], CartesianMomentum[p, D - 1]]
```

$$p^{i}$$

3-dimensional scalar products of Cartesian vectors

```
CartesianPair[CartesianMomentum[q], CartesianMomentum[p]]
```

$$\overline{p} \cdot \overline{q}$$

```
CartesianPair[CartesianMomentum[p], CartesianMomentum[p]]
```

$$\overline{p}^2$$

```
CartesianPair[CartesianMomentum[p - q], CartesianMomentum[p]]
```

$$\overline{p} \cdot (\overline{p} - \overline{q})$$

```
CartesianPair[CartesianMomentum[p], CartesianMomentum[p]]^2
```

$$\overline{p}^4$$

```
CartesianPair[CartesianMomentum[p], CartesianMomentum[p]]^3
```

$$\overline{p}^6$$

```
ExpandScalarProduct[CartesianPair[CartesianMomentum[p - q],
CartesianMomentum[p]]]
```

$$\overline{p}^2 - \overline{p} \cdot \overline{q}$$

```
CartesianPair[CartesianMomentum[-q], CartesianMomentum[p]] +
  CartesianPair[CartesianMomentum[q], CartesianMomentum[p]]
```

$$0$$

2.31 DiracBasis

DiracBasis[any] is a head which is wrapped around Dirac structures (and the **1**) as a result of the function **DiracReduce**. For more details, see the documentation for **DiracReduce**.

2.31.1 See also

Overview

2.31.2 Examples

```
Options[DiracReduce]
```

$$\{\text{Contract} \to \text{True}, \text{DiracGammaCombine} \to \text{True}, \text{DiracSimplify} \to \text{False}, \text{DiracSpinorNormalization} \to \text{Relativistic}, \text{DiracOrder} \to \text{True}, \text{DotSimplify} \to \text{True}, \text{FeynCalcExternal} \to \text{False}, \text{FeynCalcInternal} \to \text{False}, \text{FCVerbose} \to \text{False}, \text{Factoring} \to \text{False}, \text{FinalSubstitutions} \to \{\text{DiracBasis} \to \text{Identity}\}, \text{SpinorChainEvaluate} \to \text{True}\}$$

```
DiracReduce[GA[\[Mu], \[Nu], \[Rho]], FinalSubstitutions -> {}]
```

$$i\,\text{DiracBasis}\left(\text{DiracBasis}\left(\bar{\gamma}^{\$\text{MU}(\$19)}\right).\text{DiracBasis}\left(\bar{\gamma}^{5}\right)\right)\bar{\epsilon}^{\mu\nu\rho\,\$\text{MU}(\$19)} + \text{DiracBasis}\left(\bar{\gamma}^{\rho}\right)\bar{g}^{\mu\nu} - \text{DiracBasis}\left(\bar{\gamma}^{\nu}\right)\bar{g}^{\mu\rho} + \text{DiracBasis}\left(\bar{\gamma}^{\mu}\right)\bar{g}^{\nu\rho}$$

2.32 DeltaFunction

`DeltaFunction[x]` is the Dirac delta-function $\delta(x)$.

Mathematica also provides a built-in function **`DiracDelta`** with comparable properties.

2.32.1 See also

Overview, Convolute, DeltaFunctionPrime, Integrate2, SimplifyDeltaFunction.

2.32.2 Examples

```
DeltaFunction[1 - x]
```

$$\delta(1-x)$$

```
Integrate2[DeltaFunction[1 - x] f[x], {x, 0, 1}]
```

$$f(1)$$

```
Integrate2[DeltaFunction[x] f[x], {x, 0, 1}]
```

$$f(0)$$

```
Integrate2[DeltaFunction[1 - x] f[x], {x, 0, 1}]
```

$$f(1)$$

```
Convolute[DeltaFunction[1 - x], x] /. FCGV[z_] :> ToExpression[z]
```

$$-x\delta(1-x)\log(x)$$

2.33 DeltaFunctionPrime

`DeltaFunctionPrime[1 - x]` is the derivative of the Dirac delta-function $\delta(x)$.

2.33.1 See also

Overview, Convolute, DeltaFunction, DeltaFunctionDoublePrime, Integrate2, SimplifyDeltaFunction.

2.33.2 Examples

```
DeltaFunctionPrime[1 - x]
```

$$\delta'(1-x)$$

```
Integrate2[DeltaFunctionPrime[1 - x] f[x], {x, 0, 1}]
```

$$f'(1)$$

```
Integrate2[DeltaFunctionPrime[1 - x] x^2, {x, 0, 1}]
```

$$2$$

2.34 DeltaFunctionDoublePrime

DeltaFunctionDoublePrime[1 - x] is the second derivative of the Dirac delta-function $\delta(x)$.

2.34.1 See also

Overview, DeltaFunctionPrime.

2.34.2 Examples

2.35 DiracGamma

DiracGamma[x, dim] is the head of all Dirac matrices and slashes (in the internal representation). Use **GA**, **GAD**, **GS** or **GSD** for manual (short) input.

DiracGamma[x, 4] simplifies to **DiracGamma[x]**.

DiracGamma[5] is γ^5.

DiracGamma[6] is $(1+\gamma^5)/2$.

DiracGamma[7] is $(1-\gamma^5)/2$.

2.35.1 See also

Overview, DiracGammaExpand, GA, DiracSimplify, GS, DiracTrick.

2.35.2 Examples

```
DiracGamma[5]
```

$$\bar{\gamma}^5$$

```
DiracGamma[LorentzIndex[\[Alpha]]]
```

$$\bar{\gamma}^\alpha$$

A Dirac-slash, i.e., $\gamma^\mu q_\mu$, is displayed as $\gamma \cdot q$.

```
DiracGamma[Momentum[q]]
```

$$\bar{\gamma} \cdot \bar{q}$$

```
DiracGamma[Momentum[q]] . DiracGamma[Momentum[p - q]]
```

$$(\bar{\gamma} \cdot \bar{q}) . (\bar{\gamma} \cdot (\bar{p} - \bar{q}))$$

```
DiracGamma[Momentum[q, D], D]
```

$$\gamma \cdot q$$

```
GS[p - q] . GS[p]

DiracGammaExpand[%]
```

$$(\bar{\gamma} \cdot (\bar{p} - \bar{q})) . (\bar{\gamma} \cdot \bar{p})$$

$$(\bar{\gamma} \cdot \bar{p} - \bar{\gamma} \cdot \bar{q}) . (\bar{\gamma} \cdot \bar{p})$$

```
ex = GAD[\[Mu]] . GSD[p - q] . GSD[q] . GAD[\[Mu]]
```

$$\gamma^{\mu}.(\gamma \cdot (p - q)).(\gamma \cdot q).\gamma^{\mu}$$

```
DiracTrick[ex]
```

$$4((p - q) \cdot q) + (D - 4)(\gamma \cdot (p - q)).(\gamma \cdot q)$$

```
DiracSimplify[ex]
```

$$D(\gamma \cdot p).(\gamma \cdot q) - Dq^2 - 4(\gamma \cdot p).(\gamma \cdot q) + 4(p \cdot q)$$

DiracGamma may also carry Cartesian indices or appear contracted with Cartesian momenta.

```
DiracGamma[CartesianIndex[i]]
```

$$\bar{\gamma}^i$$

```
DiracGamma[CartesianIndex[i, D - 1], D]
```

$$\gamma^i$$

```
DiracGamma[CartesianMomentum[p]]
```

$$\overline{\gamma} \cdot \overline{p}$$

```
DiracGamma[CartesianMomentum[p, D - 1], D]
```

$$\gamma \cdot p$$

Temporal indices are represented using **`ExplicitLorentzIndex[0]`**

```
DiracGamma[ExplicitLorentzIndex[0]]
```

$$\bar{\gamma}^0$$

2.36 GA

`GA[mu]` can be used as input for a 4-dimensional γ^μ and is transformed into **`DiracGamma[LorentzIndex[mu]]`** by FeynCalcInternal (=FCI).

`GA[mu , nu , ...]` is a short form for **`GA[mu].GA[nu]`**.

2.36.1 See also

Overview, DiracGamma, GAD, GS.

2.36.2 Examples

```
GA[\[Mu]]
```

$$\bar{\gamma}^\mu$$

```
GA[\[Mu], \[Nu]] - GA[\[Nu], \[Mu]]
```

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu} - \bar{\gamma}^{\nu}.\bar{\gamma}^{\mu}$$

```
StandardForm[FCI[GA[\[Mu]]]]

(*DiracGamma[LorentzIndex[\[Mu]]]*)
```

```
GA[\[Mu], \[Nu], \[Rho], \[Sigma]]
```

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\sigma}$$

```
StandardForm[GA[\[Mu], \[Nu], \[Rho], \[Sigma]]]

(*GA[\[Mu]] . GA[\[Nu]] . GA[\[Rho]] . GA[\[Sigma]]*)
```

```
GA[\[Alpha]] . (GS[p] + m) . GA[\[Beta]]
```

$$\bar{\gamma}^{\alpha}.\left(\bar{\gamma}\cdot\overline{p} + m\right).\bar{\gamma}^{\beta}$$

2.37 GA5

GA5 is equivalent to **DiracGamma[5]** and denotes γ^5.

2.37.1 See also

Overview, DiracGamma, GA, GS.

2.37.2 Examples

```
GA5
```

$$\bar{\gamma}^{5}$$

```
GA5 // StandardForm

(*DiracGamma[5]*)
```

2.38 GS

GS[p] can be used as input for a 4-dimensional $p^\mu \gamma_\mu$ and is transformed into **DiracGamma[Momentum[p]]** by **FeynCalcInternal** (=**FCI**).

GS[p, q, ...] is a short form for **GS[p].GS[q]**.

2.38.1 See also

Overview, DiracGamma, GA, GAD.

2.38.2 Examples

```
GS[p]
```

$$\bar{\gamma} \cdot \bar{p}$$

```
GS[p] // FCI // StandardForm

(*DiracGamma[Momentum[p]]*)
```

```
GS[p, q, r, s]
```

$$(\bar{\gamma} \cdot \bar{p}) . (\bar{\gamma} \cdot \bar{q}) . (\bar{\gamma} \cdot \bar{r}) . (\bar{\gamma} \cdot \bar{s})$$

```
GS[p, q, r, s] // StandardForm

(*GS[p] . GS[q] . GS[r] . GS[s]*)
```

```
GS[q] . (GS[p] + m) . GS[q]
```

$$(\bar{\gamma} \cdot \bar{q}) . (\bar{\gamma} \cdot \bar{p} + m) . (\bar{\gamma} \cdot \bar{q})$$

2.39 GAD

GAD[mu] can be used as input for a D-dimensional γ^μ and is transformed into **DiracGamma[LorentzIndex[mu, D], D]** by **FeynCalcInternal** (=**FCI**).

GAD[mu , nu , ...] is a short form for **GAD[mu].GAD[nu]**.

2.39.1 See also

Overview, DiracGamma, GA, GS.

2.39.2 Examples

```
GAD[\[Mu]]
```

$$\gamma^\mu$$

```
GAD[\[Mu], \[Nu]] - GAD[\[Nu], \[Mu]]
```

$$\gamma^\mu.\gamma^\nu - \gamma^\nu.\gamma^\mu$$

```
StandardForm[FCI[GAD[\[Mu]]]]

(*DiracGamma[LorentzIndex[\[Mu], D], D]*)
```

```
GAD[\[Mu], \[Nu], \[Rho], \[Sigma]]
```

$$\gamma^\mu.\gamma^\nu.\gamma^\rho.\gamma^\sigma$$

```
StandardForm[GAD[\[Mu], \[Nu], \[Rho], \[Sigma]]]

(*GAD[\[Mu]] . GAD[\[Nu]] . GAD[\[Rho]] . GAD[\[Sigma]]*)
```

```
GAD[\[Alpha]] . (GSD[p] + m) . GAD[\[Beta]]
```

$$\gamma^\alpha.(m + \gamma \cdot p).\gamma^\beta$$

2.40 GSD

GSD[p] can be used as input for a D-dimensional $p^\mu \gamma_\mu$ and is transformed into **DiracGamma[Momentum[p, D], D]** by **FeynCalcInternal** (=**FCI**).

GSD[p, q, ...] is a short form for **GSD[p].GSD[q]**.

2.40.1 See also

Overview, DiracGamma, GA, GAD.

2.40.2 Examples

```
GSD[p]
```

$$\gamma \cdot p$$

```
GSD[p] // FCI // StandardForm

(*DiracGamma[Momentum[p, D], D]*)
```

```
GSD[p, q, r, s]
```

$$(\gamma \cdot p).(\gamma \cdot q).(\gamma \cdot r).(\gamma \cdot s)$$

```
GSD[p, q, r, s] // StandardForm

(*GSD[p] . GSD[q] . GSD[r] . GSD[s]*)
```

```
GSD[q] . (GSD[p] + m) . GSD[q]
```

$$(\gamma \cdot q).(m + \gamma \cdot p).(\gamma \cdot q)$$

2.41 GAE

GAE[mu] can be used as input for a **D-4**-dimensional $\gamma^\mu$and is transformed into **DiracGamma[LorentzIndex[mu, D-4], D-4]** by **FeynCalcInternal** (**FCI**).

GAE[mu, nu , ...] is a short form for **GAE[mu].GAE[nu] ...**.

2.41.1 See also

Overview, DiracGamma, GA, GS, GAD.

2.41.2 Examples

```
GAE[\[Mu]]
```

$$\hat{\gamma}^{\mu}$$

```
GAE[\[Mu], \[Nu]] - GAE[\[Nu], \[Mu]]
```

$$\hat{\gamma}^{\mu}.\hat{\gamma}^{\nu} - \hat{\gamma}^{\nu}.\hat{\gamma}^{\mu}$$

```
StandardForm[FCI[GAE[\[Mu]]]]
(*DiracGamma[LorentzIndex[\[Mu], -4 + D], -4 + D]*)
```

```
GAE[\[Mu], \[Nu], \[Rho], \[Sigma]]
```

$$\hat{\gamma}^{\mu}.\hat{\gamma}^{\nu}.\hat{\gamma}^{\rho}.\hat{\gamma}^{\sigma}$$

```
StandardForm[GAE[\[Mu], \[Nu], \[Rho], \[Sigma]]]
(*GAE[\[Mu]] . GAE[\[Nu]] . GAE[\[Rho]] . GAE[\[Sigma]]*)
```

```
GAE[\[Alpha]] FVD[p, \[Alpha]] // Contract
```

$$\hat{\gamma} \cdot \hat{p}$$

```
GAE[\[Alpha]] FV[p, \[Alpha]] // Contract
```

$$0$$

In order to use Dirac algebra with $D - 4$-dimensional objects you need to activate the t'Hooft-Veltman-Breitenlohner-Maison scheme first

```
FCSetDiracGammaScheme["NDR"]

DiracSimplify[GAE[\[Mu]] . GAD[\[Mu]]]
```

NDR

DiracTrace: Expressions that mix D- , 4- and D- 4- dimensional quantities are forbidden in Dirac matrix chains unless you are using the t'Hooft- Veltman scheme. For every other scheme, please recheck your input expressions and ensure that all matrices, spinors and tensors are purely D- dimensional. You might want to use FCGetDimensions[exp] to find the offending terms and fix them by hand or employ ChangeDimension[exp,D] to convert the whole expression to D- dimensions. If you explicitly intend to use the t'Hooft- Veltman scheme, please activate it via FCSetDiracGammaScheme["BMHV"].

$Aborted

```
FCSetDiracGammaScheme["BMHV"]

DiracSimplify[GAE[\[Mu]] . GAD[\[Mu]]]
```

BMHV

$D - 4$

```
FCSetDiracGammaScheme["NDR"]
```

NDR

2.42 GSE

GSE[p] can be used as input for a $D-4$-dimensional $\gamma \cdot p = \gamma^\mu p_\mu$ and is transformed into **DiracGamma[Momentum[p, D-4], D-4]** by **FeynCalcInternal** (**FCI**). **GSE[p, q, ...]** is a short form for **GSE[p].GSE[q]. ...**.

2.42.1 See also

Overview, DiracGamma, GA, GAD, GSD.

2.42.2 Examples

```
GSE[p]
```

$$\hat{\gamma} \cdot \hat{p}$$

```
GSE[p] // FCI // StandardForm

(*DiracGamma[Momentum[p, -4 + D], -4 + D]*)
```

```
GSE[p, q, r, s]
```

$$(\hat{\gamma} \cdot \hat{p}) . (\hat{\gamma} \cdot \hat{q}) . (\hat{\gamma} \cdot \hat{r}) . (\hat{\gamma} \cdot \hat{s})$$

```
GSE[p, q, r, s] // StandardForm

(*GSE[p] . GSE[q] . GSE[r] . GSE[s]*)
```

```
GSE[q] . (GSE[p] + m) . GSE[q]
```

$$(\hat{\gamma} \cdot \hat{q}) . (m + \hat{\gamma} \cdot \hat{p}) . (\hat{\gamma} \cdot \hat{q})$$

In order to use Dirac algebra with $D - 4$ dimensional objects you need to activate the t'Hooft-Veltman-Breitenlohner-Maison scheme first

```
FCSetDiracGammaScheme["NDR"];

DiracSimplify[GSE[q] . GS[q] . GSE[q]]
```

DiracTrace: Expressions that mix D-, 4- and D-4-dimensional quantities are forbidden in Dirac matrix chains unless you are using the t'Hooft-Veltman scheme. For every other scheme, please recheck your input expressions and ensure that all matrices, spinors and tensors are purely D-dimensional. You might want to use FCGetDimensions[exp] to find the offending terms and fix them by hand or employ ChangeDimension[exp,D] to convert the whole expression to D-dimensions. If you explicitly intend to use the t'Hooft-Veltman scheme, please activate it via FCSetDiracGammaScheme["BMHV"].

$Aborted

```
FCSetDiracGammaScheme["BMHV"];

DiracSimplify[GSE[q] . GS[q] . GSE[q]]
```

$$\hat{q}^2 \, (-(\bar{\gamma} \cdot \bar{q}))$$

```
FCSetDiracGammaScheme["NDR"];
```

2.43 DiracIndexDelta

`DiracIndexDelta[DiracIndex[i], DiracIndex[j]]` is the Kronecker-delta in the Dirac space with two explicit Dirac indices **`i`** and **`j`**.

2.43.1 See also

Overview, DiracChain, DCHN, DiracIndex, DIDelta, DiracChainJoin, DiracChainCombine, DiracChainExpand, DiracChainFactor.

2.43.2 Examples

```
DiracIndexDelta[DiracIndex[i], DiracIndex[j]]
```

$$\delta_{ij}$$

```
ex = DiracIndexDelta[DiracIndex[i], DiracIndex[j]]^2
```

$$\delta_{ij}^2$$

```
DiracChainJoin[ex]
```

$$4$$

```
DiracChainJoin[ex, TraceOfOne -> D]
```

$$D$$

```
ex = DiracIndexDelta[DiracIndex[i], DiracIndex[j]]
DiracIndexDelta[DiracIndex[j], DiracIndex[k]]
```

$$\delta_{ij}\delta_{jk}$$

```
DiracChainJoin[ex]
```

$$\delta_{ik}$$

```
ex = DiracIndexDelta[DiracIndex[i2], DiracIndex[i3]]
DiracIndexDelta[DiracIndex[i4], DiracIndex[i5]] DiracChain[DiracIndex[i7],
Spinor[-Momentum[q], 0, 1]] DiracChain[Spinor[Momentum[p], m, 1],
DiracIndex[i0]] DiracChain[DiracGamma[LorentzIndex[\[Mu]]], DiracIndex[i1],
DiracIndex[i2]] DiracChain[DiracGamma[LorentzIndex[\[Nu]]], DiracIndex[i5],
DiracIndex[i6]] DiracChain[m + DiracGamma[Momentum[p]], DiracIndex[i3],
DiracIndex[i4]]
```

$$\delta_{\text{i2 i3}}\delta_{\text{i4 i5}}\left(\bar{\gamma}^{\mu}\right)_{\text{i1 i2}}\left(\bar{\gamma}^{\nu}\right)_{\text{i5 i6}}\left(\varphi(-\overline{q})\right)_{\text{i7}}\left(\bar{\gamma}\cdot\overline{p}+m\right)_{\text{i3 i4}}\left(\varphi(\overline{p},m)\right)_{\text{i0}}$$

```
DiracChainJoin[ex]
```

$$\left(\varphi(-\overline{q})\right)_{\text{i7}}\left(\varphi(\overline{p},m)\right)_{\text{i0}}\left(\bar{\gamma}^{\mu}.\left(\bar{\gamma}\cdot\overline{p}+m\right).\bar{\gamma}^{\nu}\right)_{\text{i1 i6}}$$

```
DiracChainJoin[ex DIDelta[i0, i1]]
```

$$\left(\varphi(-\overline{q})\right)_{\text{i7}}\left(\varphi(\overline{p},m).\bar{\gamma}^{\mu}.\left(\bar{\gamma}\cdot\overline{p}+m\right).\bar{\gamma}^{\nu}\right)_{\text{i6}}$$

```
DiracChainJoin[ex DIDelta[i7, i6]]
```

$$\left(\varphi(\overline{p},m)\right)_{\text{i0}}\left(\bar{\gamma}^{\mu}.\left(\bar{\gamma}\cdot\overline{p}+m\right).\bar{\gamma}^{\nu}.\varphi(-\overline{q})\right)_{\text{i1}}$$

2.44 DIDelta

DIDelta[i, j] is the Kronecker-delta in the Dirac space.

DIDelta[i, j] is transformed into **DiracDelta[DiracIndex[i], DiracIndex[j]]** by **FeynCalcInternal**.

2.44.1 See also

Overview, DiracChain, DCHN, DiracIndex, DiracIndexDelta, DiracChainJoin, DiracChainExpand, DiracChainFactor.

2.44.2 Examples

```
DIDelta[i, j]
```

$$\delta_{ij}$$

```
DIDelta[i, i]

DiracChainJoin[%]
```

$$\delta_{ii}$$

$$4$$

```
DIDelta[i, j]^2

DiracChainJoin[%]
```

$$\delta_{ij}^2$$

$$4$$

```
DIDelta[i, j] DIDelta[j, k]

DiracChainJoin[%]
```

$$\delta_{ij}\delta_{jk}$$

$$\delta_{ik}$$

```
ex = DCHN[SpinorUBar[p, m], i0] DCHN[GA[\[Mu]], i1, i2] DCHN[GS[p] + m, i3,
i4] DCHN[GA[\[Nu]], i5, i6] DIDelta[i2, i3] DIDelta[i4, i5] DCHN[i7,
SpinorV[q]]
```

$$\delta_{\text{i2 i3}}\delta_{\text{i4 i5}}(v(q))_{\text{i7}}\,(\bar{\gamma}^{\mu})_{\text{i1 i2}}\,(\bar{\gamma}^{\nu})_{\text{i5 i6}}\,(\bar{u}(p,m))_{\text{i0}}\,(\bar{\gamma}\cdot\overline{p}+m)_{\text{i3 i4}}$$

```
ex // FCI // StandardForm

(*DiracChain[DiracIndex[i7], Spinor[-Momentum[q], 0, 1]]
DiracChain[Spinor[Momentum[p], m, 1], DiracIndex[i0]]
DiracChain[DiracGamma[LorentzIndex[\[Mu]]], DiracIndex[i1], DiracIndex[i2]]
DiracChain[DiracGamma[LorentzIndex[\[Nu]]], DiracIndex[i5], DiracIndex[i6]]
DiracChain[m + DiracGamma[Momentum[p]], DiracIndex[i3], DiracIndex[i4]]
DiracIndexDelta[DiracIndex[i2], DiracIndex[i3]]
DiracIndexDelta[DiracIndex[i4], DiracIndex[i5]]*)
```

```
DiracChainJoin[ex]
```

$$(\varphi(-\overline{q}))_{\text{i7}}\,(\varphi(\overline{p},m))_{\text{i0}}\,(\bar{\gamma}^{\mu}.\,(\bar{\gamma}\cdot\overline{p}+m)\,.\bar{\gamma}^{\nu})_{\,\text{i1 i6}}$$

```
DiracChainJoin[ex DIDelta[i0, i1]]
```

$$(\varphi(-\overline{q}))_{\text{i7}}\,(\varphi(\overline{p},m).\bar{\gamma}^{\mu}.\,(\bar{\gamma}\cdot\overline{p}+m)\,.\bar{\gamma}^{\nu})_{\,\text{i6}}$$

```
DiracChainJoin[ex DIDelta[i7, i6]]
```

$$(\varphi(\overline{p},m))_{\text{i0}}\,(\bar{\gamma}^{\mu}.\,(\bar{\gamma}\cdot\overline{p}+m)\,.\bar{\gamma}^{\nu}.\varphi(-\overline{q}))_{\,\text{i1}}$$

2.45 DiracSigma

`DiracSigma[a, b]` stands for $I/2(a.b - b.a)$ in 4 dimensions.

a and **b** must have head **`DiracGamma`**, **`GA`** or **`GS`**. Only antisymmetry is implemented.

2.45.1 See also

Overview, DiracSigmaExplicit.

2.45.2 Examples

```
DiracSigma[GA[\[Alpha]], GA[\[Beta]]]

DiracSigmaExplicit[%]
```

$$\sigma^{\alpha\beta}$$

$$\frac{1}{2} i \left(\bar{\gamma}^{\alpha}.\bar{\gamma}^{\beta} - \bar{\gamma}^{\beta}.\bar{\gamma}^{\alpha}\right)$$

```
DiracSigma[GA[\[Beta]], GA[\[Alpha]]]
```

$$-\sigma^{\alpha\beta}$$

```
DiracSigma[GS[p], GS[q]]

DiracSigmaExplicit[%]
```

$$\sigma^{pq}$$

$$\frac{1}{2} i \left((\bar{\gamma} \cdot \overline{p}) . (\bar{\gamma} \cdot \overline{q}) - (\bar{\gamma} \cdot \overline{q}) . (\bar{\gamma} \cdot \overline{p})\right)$$

The antisymmetry property is built-in

```
DiracSigma[GA[\[Alpha]], GA[\[Alpha]]]
```

$$0$$

2.46 DOT

`DOT[a, b, ...]` is the FeynCalc function for non-commutative multiplication. By default it is set to the Mathematica Dot function. This can in principle be disabled by setting **`DOT=.`**, but then non-commutative products should to be entered like DOT[GA[mu], m + GS[p], GA[mu]] etc.

2.46.1 See also

Overview, DotSimplify.

2.46.2 Examples

2.47 Eps

Eps[a, b, c, d] is the head of the totally antisymmetric ϵ (Levi-Civita) tensor. The **a, b, ...** may have head **LorentzIndex** or **Momentum**.

When some indices of a Levi-Civita-tensor are contracted with 4-vectors, FeynCalc suppresses explicit dummy indices by putting those vectors into the corresponding index slots. For example, $\varepsilon^{p_1 p_2 p_3 p_4}$ (accessible via **LC[][p1, p2, p3, p4]**) correspond to $\varepsilon_{\mu\nu\rho\sigma} p_1^\mu p_2^\nu p_3^\rho p_4^\sigma$.

2.47.1 See also

Overview, EpsContract, EpsEvaluate, LC, LCD.

2.47.2 Examples

```
Eps[LorentzIndex[\[Mu]], LorentzIndex[\[Nu]], LorentzIndex[\[Rho]],
LorentzIndex[\[Sigma]]]
```

$$\bar{\epsilon}^{\mu\nu\rho\sigma}$$

```
Eps[Momentum[p], LorentzIndex[\[Nu]], LorentzIndex[\[Rho]],
LorentzIndex[\[Sigma]]]
```

$$\bar{\epsilon}^{\overline{p}\nu\rho\sigma}$$

```
Eps[b, a, c, d] // StandardForm

(*Eps[b, a, c, d]*)
```

```
Eps[ExplicitLorentzIndex[0], ExplicitLorentzIndex[1],
ExplicitLorentzIndex[2],
  ExplicitLorentzIndex[3]]
```

$$\bar{\epsilon}^{0123}$$

```
Eps[LorentzIndex[\[Mu]], LorentzIndex[\[Nu]], LorentzIndex[\[Rho]],
LorentzIndex[\[Sigma]]] *
   Pair[LorentzIndex[\[Mu]], Momentum[Subscript[p, 1]]]
Pair[LorentzIndex[\[Nu]], Momentum[Subscript[p, 2]]]*
   Pair[LorentzIndex[\[Rho]], Momentum[Subscript[p, 3]]]
Pair[LorentzIndex[\[Sigma]], Momentum[Subscript[p, 4]]]

Contract[%]
```

$$\overline{p}_1{}^{\mu}\overline{p}_2{}^{\nu}\overline{p}_3{}^{\rho}\overline{p}_4{}^{\sigma}\bar{\epsilon}^{\mu\nu\rho\sigma}$$

$$\bar{\epsilon}^{\overline{p}_1\overline{p}_2\overline{p}_3\overline{p}_4}$$

```
Eps[LorentzIndex[\[Mu]], LorentzIndex[\[Nu]], LorentzIndex[\[Rho]],
LorentzIndex[\[Sigma]]]

Contract[% %]
```

$$\bar{\epsilon}^{\mu\nu\rho\sigma}$$

$$-24$$

```
Eps[LorentzIndex[\[Mu], D], LorentzIndex[\[Nu], D], LorentzIndex[\[Rho],
D], LorentzIndex[\[Sigma], D]]

Contract[% %] // Factor2
```

$$\epsilon^{\mu\nu\rho\sigma}$$

$$(1-D)(2-D)(3-D)D$$

```
ex1 = -(I/24) LCD[\[Mu], \[Nu], \[Rho], \[Alpha]] . GAD[\[Mu], \[Nu],
\[Rho], \[Alpha]] // FCI
```

$$-\frac{1}{24}i\epsilon^{\mu\nu\rho\alpha}.\gamma^{\mu}.\gamma^{\nu}.\gamma^{\rho}.\gamma^{\alpha}$$

```
ex2 = -(I/24) LCD[\[Mu]', \[Nu]', \[Rho]', \[Alpha]'] . GAD[\[Mu]', \[Nu]',
\[Rho]', \[Alpha]'] // FCI
```

$$-\frac{1}{24} i \epsilon^{\mu' \nu' \rho' \alpha'} . \gamma^{\mu'} . \gamma^{\nu'} . \gamma^{\rho'} . \gamma^{\alpha'}$$

```
DiracSimplify[ex1 . ex2] // Factor2

% /. D -> 4
```

$$-\frac{1}{24}(1-D)(2-D)(3-D)D$$

$$1$$

2.48 LC

LC[m, n, r, s] evaluates to 4-dimensional ε^{mnrs} by virtue of applying **FeynCalcInternal**.

LC[m, ...][p, ...] evaluates to 4-dimensional $\epsilon^{m \cdots \mu \cdots} p_{\mu \ldots}$ applying **FeynCalcInternal**.

When some indices of a Levi-Civita-tensor are contracted with 4-vectors, FeynCalc suppresses explicit dummy indices by putting those vectors into the corresponding index slots. For example, $\varepsilon^{p_1 p_2 p_3 p_4}$ (accessible via **LC[][p1, p2, p3, p4]**) correspond to $\varepsilon_{\mu\nu\rho\sigma} p_1^\mu p_2^\nu p_3^\rho p_4^\sigma$.

2.48.1 See also

Overview, Eps, LCD.

2.48.2 Examples

```
LC[\[Mu], \[Nu], \[Rho], \[Sigma]]
```

$$\bar{\epsilon}^{\mu\nu\rho\sigma}$$

```
LC[\[Mu], \[Nu], \[Rho], \[Sigma]] // FCI // StandardForm

(*Eps[LorentzIndex[\[Mu]], LorentzIndex[\[Nu]], LorentzIndex[\[Rho]],
LorentzIndex[\[Sigma]]]*)
```

```
LC[\[Mu], \[Nu]][p, q]
```

$$\bar{\epsilon}^{\mu\nu\overline{pq}}$$

```
LC[\[Mu], \[Nu]][p, q] // FCI // StandardForm

(*Eps[LorentzIndex[\[Mu]], LorentzIndex[\[Nu]], Momentum[p], Momentum[q]]*)
```

```
Contract[LC[\[Mu], \[Nu], \[Rho]][p] LC[\[Mu], \[Nu], \[Rho]][q]
```

$$-6\,(\overline{p}\cdot\overline{q})$$

```
LC[\[Mu], \[Nu], \[Rho], \[Sigma]] FV[Subscript[p, 1], \[Mu]]
FV[Subscript[p, 2], \[Nu]] FV[Subscript[p, 3], \[Rho]] FV[Subscript[p, 4],
\[Sigma]]

Contract[%]
```

$$\overline{p}_1{}^{\mu}\overline{p}_2{}^{\nu}\overline{p}_3{}^{\rho}\overline{p}_4{}^{\sigma}\bar{\epsilon}^{\mu\nu\rho\sigma}$$

$$\bar{\epsilon}^{\overline{p}_1\overline{p}_2\overline{p}_3\overline{p}_4}$$

2.49 LCD

`LCD[m, n, r, s]` evaluates to D-dimensional ε^{mnrs} by virtue of applying **`FeynCalcInternal`**.

`LCD[m, ...][p, ...]` evaluates to D-dimensional $\epsilon^{m\ldots\mu\ldots}p_{\mu\ldots}$ applying **`FeynCalcInternal`**.

When some indices of a Levi-Civita-tensor are contracted with 4-vectors, FeynCalc suppresses explicit dummy indices by putting those vectors into the corresponding index slots. For example, $\varepsilon^{p_1p_2p_3p_4}$ (accessible via **`LCD[][p1, p2, p3, p4]`**) correspond to $\varepsilon_{\mu\nu\rho\sigma}p_1^{\mu}p_2^{\nu}p_3^{\rho}p_4^{\sigma}$.

2.49.1 See also

Overview, Eps, LC.

2.49.2 Examples

```
LCD[\[Mu], \[Nu], \[Rho], \[Sigma]]
```

$$\epsilon^{\mu\nu\rho\sigma}$$

```
LCD[\[Mu], \[Nu], \[Rho], \[Sigma]] // FCI // StandardForm

(*Eps[LorentzIndex[\[Mu], D], LorentzIndex[\[Nu], D], LorentzIndex[\[Rho],
D], LorentzIndex[\[Sigma], D]]*)
```

```
LCD[\[Mu], \[Nu]][p, q]
```

$$\epsilon^{\mu\nu pq}$$

```
LCD[\[Mu], \[Nu]][p, q] // FCI // StandardForm

(*Eps[LorentzIndex[\[Mu], D], LorentzIndex[\[Nu], D], Momentum[p, D],
Momentum[q, D]]*)
```

```
Factor2[Contract[LCD[\[Mu], \[Nu], \[Rho]][p] LCD[\[Mu], \[Nu],
\[Rho]][q]]]
```

$$(1-D)(2-D)(3-D)(p\cdot q)$$

```
LCD[\[Mu], \[Nu], \[Rho], \[Sigma]] FVD[Subscript[p, 1], \[Mu]]
FVD[Subscript[p, 2], \[Nu]] FVD[Subscript[p, 3], \[Rho]] FVD[Subscript[p,
4], \[Sigma]]

Contract[%]
```

$$p_1{}^{\mu} p_2{}^{\nu} p_3{}^{\rho} p_4{}^{\sigma} \epsilon^{\mu\nu\rho\sigma}$$

$$\epsilon^{p_1 p_2 p_3 p_4}$$

2.50 CLC

CLC[m, n, r] evaluates to **Eps[CartesianIndex[m], CartesianIndex[n], CartesianIndex[r]]** applying **FeynCalcInternal**.

CLC[m, ...][p, ...] evaluates to **Eps[CartesianIndex[m], ..., CartesianMomentum[p], ...]** applying **FeynCalcInternal**.

When some indices of a Levi-Civita-tensor are contracted with 3-vectors, FeynCalc suppresses explicit dummy indices by putting those vectors into the corresponding index slots. For example, $\varepsilon^{p_1 p_2 p_3}$ (accessible via **CLC[][p1, p2, p3]**) correspond to $\varepsilon^{ijk} p_1^i p_2^j p_3^k$.

2.50.1 See also

Overview, LC, Eps.

2.50.2 Examples

```
CLC[i, j, k]
```

$$\bar{\epsilon}^{ijk}$$

```
CLC[i, j, k] // FCI // StandardForm

(*Eps[CartesianIndex[i], CartesianIndex[j], CartesianIndex[k]]*)
```

```
CLC[i][p, q]
```

$$\bar{\epsilon}^{i\overline{pq}}$$

```
CLC[i][p, q] // FCI // StandardForm

(*Eps[CartesianIndex[i], CartesianMomentum[p], CartesianMomentum[q]]*)
```

```
Contract[CLC[i, j, k] CLC[i, l, m]]
```

$$\bar{\delta}^{jl}\bar{\delta}^{km} - \bar{\delta}^{jm}\bar{\delta}^{kl}$$

```
CLC[i, j, k] CV[Subscript[p, 1], i] CV[Subscript[p, 2], j] CV[Subscript[p,
3], k]

Contract[%]
```

$$\overline{p}_1{}^i \overline{p}_2{}^j \overline{p}_3{}^k \bar{\epsilon}^{ijk}$$

$$\bar{\epsilon}^{\overline{p}_1 \overline{p}_2 \overline{p}_3}$$

2.51 CLCD

CLCD[m, n, r] evaluates to **Eps[CartesianIndex[m, D-1], CartesianIndex[n, D-1], CartesianIndex[r, D-1]]** applying **FeynCalcInternal**.

CLC[m, ...][p, ...] evaluates to **Eps[CartesianIndex[m, D-1], ..., CartesianMomentum[p, D-1], ...]** applying **FeynCalcInternal**.

When some indices of a Levi-Civita-tensor are contracted with 3-vectors, FeynCalc suppresses explicit dummy indices by putting those vectors into the corresponding index slots. For example, $\varepsilon^{p_1 p_2 p_3}$ (accessible via **CLCD[][p1, p2, p3]**) correspond to $\varepsilon^{ijk} p_1^i p_2^j p_3^k$.

2.51.1 See also

Overview, LCD, Eps.

2.51.2 Examples

```
CLCD[i, j, k]
```

$$\epsilon^{ijk}$$

```
CLCD[i, j, k] // FCI // StandardForm

(*Eps[CartesianIndex[i, -1 + D], CartesianIndex[j, -1 + D],
CartesianIndex[k, -1 + D]]*)
```

```
CLCD[i, j][p]
```

$$\epsilon^{ijp}$$

```
CLCD[i, j][p] // FCI // StandardForm

(*Eps[CartesianIndex[i, -1 + D], CartesianIndex[j, -1 + D],
CartesianMomentum[p, -1 + D]]*)
```

```
CLCD[i, j][p] CLCD[i, j][q] // Contract // Factor2
```

$$(2-D)(3-D)(p\cdot q)$$

```
CLCD[i, j, k] CVD[Subscript[p, 1], i] CVD[Subscript[p, 2], j]
CVD[Subscript[p, 3], k]

Contract[%]
```

$$p_1{}^i p_2{}^j p_3{}^k \epsilon^{ijk}$$

$$\epsilon^{p_1 p_2 p_3}$$

2.52 EpsilonUV

EpsilonUV denotes $(D-4)$, where D is the number of space-time dimensions.

EpsilonUV stands for a small positive number that explicitly regulates only UV divergences.

2.52.1 See also

Overview, Epsilon, EpsilonIR.

2.52.2 Examples

```
EpsilonUV
```

$$\varepsilon_{\text{UV}}$$

2.53 EpsilonIR

EpsilonIR denotes $(D-4)$, where D is the number of space-time dimensions.

EpsilonIR stands for a small negative number that explicitly regulates only IR divergences.

2.53.1 See also

Overview, Epsilon, EpsilonUV.

2.53.2 Examples

```
EpsilonIR
```

$$\varepsilon_{\text{IR}}$$

2.54 Epsilon

Epsilon is $(n-4)$, where n is the space-time dimension.

Epsilon stands for a small positive number.

2.54.1 See also

Overview, Series2.

2.54.2 Examples

```
Epsilon
```

$$\varepsilon$$

Epsilon has no functional properties, but some upvalues are changed:

```
{Re[Epsilon] > -4, Re[Epsilon] > -3, Re[Epsilon] > -2, Re[Epsilon] > -1,
Re[Epsilon] > 0}
```

$$\{\text{True}, \text{True}, \text{True}, \text{True}, \text{True}\}$$

2.55 ExplicitLorentzIndex

`ExplicitLorentzIndex[ind]` is an explicit Lorentz index, i.e., **`ind`** is an integer.

2.55.1 See also

Overview, LorentzIndex, Pair.

2.55.2 Examples

```
Pair[LorentzIndex[1], LorentzIndex[\[Mu]]]
```

$$\bar{g}^{1\mu}$$

```
Pair[LorentzIndex[1], LorentzIndex[\[Mu]]] // StandardForm

(*Pair[ExplicitLorentzIndex[1], LorentzIndex[\[Mu]]]*)
```

2.56 LorentzIndex

`LorentzIndex[mu]` denotes a 4-dimensional Lorentz index.

For other than 4 dimensions: **`LorentzIndex[mu, D]`** or **`LorentzIndex[mu]`** etc.

`LorentzIndex[mu, 4]` simplifies to **`LorentzIndex[mu]`**.

2.56.1 See also

Overview, ChangeDimension, Momentum.

2.56.2 Examples

This denotes a 4-dimensional Lorentz index.

```
LorentzIndex[\[Alpha]]
```

$$\alpha$$

An optional second argument can be given for a dimension different from 4.

```
LorentzIndex[\[Alpha], n]
```

$$\alpha$$

2.57 ExplicitDiracIndex

`ExplicitDiracIndex[ind]` is an explicit Dirac index, i.e., **`ind`** is an integer.

2.57.1 See also

Overview, DiracChain, DCHN, DiracIndex, DiracIndexDelta, DIDelta, DiracChainJoin, DiracChainCombine, DiracChainExpand, DiracChainFactor.

2.57.2 Examples

```
DCHN[GA[\[Mu]], 1, 2]
```

$$(\bar{\gamma}^{\mu})_{12}$$

```
DCHN[GA[\[Mu]], 1, 2] // FCI // StandardForm

(*DiracChain[DiracGamma[LorentzIndex[\[Mu]]], ExplicitDiracIndex[1],
ExplicitDiracIndex[2]]*)
```

2.58 DiracIndex

`DiracIndex` is the head of Dirac indices. The internal representation of a four-dimensional spinorial index **`i`** is **`DiracIndex[i]`**.

If the first argument is an integer, **`DiracIndex[i]`** turns into **`ExplicitDiracIndex[i]`**.

Dirac indices are the indices that denote the components of Dirac matrices or spinors. They should not be confused with the Lorentz indices attached to the Dirac matrices. For example in the case of γ^{μ}_{ij}, μ is a Lorentz index, while i and j are Dirac (spinorial) indices.

2.58.1 See also

Overview, DiracChain, DCHN, ExplicitDiracIndex, DiracIndexDelta, DIDelta, DiracChainJoin, DiracChainCombine, DiracChainExpand, DiracChainFactor.

2.58.2 Examples

```
DiracIndex[i]
```

$$i$$

```
DiracIndex[i] // StandardForm

(*DiracIndex[i]*)
```

```
DiracIndex[2]
```

$$2$$

```
DiracIndex[2] // StandardForm

(*ExplicitDiracIndex[2]*)
```

```
DIDelta[i, j] // FCI // StandardForm

(*DiracIndexDelta[DiracIndex[i], DiracIndex[j]]*)
```

2.59 ExplicitPauliIndex

`ExplicitPauliIndex[ind]` is an explicit Pauli index, i.e., **`ind`** is an integer.

2.59.1 See also

Overview, PauliChain, PCHN, PauliIndex, PauliIndexDelta, PIDelta, PauliChainJoin, PauliChainCombine, PauliChainExpand, PauliChainFactor.

2.59.2 Examples

```
PCHN[SI[i], 1, 2]
```

$$\left(\bar{\sigma}^{i}\right)_{12}$$

```
PCHN[SI[i], 1, 2] // FCI // StandardForm

(*PauliChain[PauliSigma[LorentzIndex[i]], ExplicitPauliIndex[1],
ExplicitPauliIndex[2]]*)
```

2.60 PauliIndex

PauliIndex is the head of Pauli indices. The internal representation of a two-dimensional spinorial index **i** is **PauliIndex[i]**.

If the first argument is an integer, **PauliIndex[i]** turns into **ExplicitPauliIndex[i]**.

Pauli indices are the indices that denote the components of Pauli matrices or spinors. They should not be confused with the Cartesian indices attached to the Pauli matrices. For example in the case of σ_{ij}^{k}, k is a Lorentz index, while i and j are Pauli (spinorial) indices.

2.60.1 See also

Overview, PauliChain, PCHN, ExplicitPauliIndex, PauliIndexDelta, DIDelta, PauliChainJoin, PauliChainCombine, PauliChainExpand, PauliChainFactor.

2.60.2 Examples

```
PauliIndex[i]
```

$$i$$

```
PauliIndex[i] // StandardForm

(*PauliIndex[i]*)
```

```
PauliIndex[2]
```

$$2$$

```
PauliIndex[2] // StandardForm

(*ExplicitPauliIndex[2]*)
```

```
PIDelta[i, j] // FCI // StandardForm

(*PauliIndexDelta[PauliIndex[i], PauliIndex[j]]*)
```

2.61 ExplicitSUNIndex

ExplicitSUNIndex[ind] is a specific $SU(N)$ index in the adjoint representation, i.e. **ind** is an integer.

2.61.1 See also

Overview, FCI, SUNDelta, SUNF, SUNIndex.

2.61.2 Examples

```
ExplicitSUNIndex[1]
```

1

2.62 SUNIndex

SUNIndex[a] is an $SU(N)$ index in the adjoint representation. If the argument is an integer, **SUNIndex[a]** turns into **ExplicitSUNIndex[a]**.

2.62.1 See also

Overview, ExplicitSUNIndex, SUNDelta, SUNF.

2.62.2 Examples

```
SUNIndex[i]
```

i

```
SUNIndex[i] // StandardForm

(*SUNIndex[i]*)
```

```
SUNIndex[2]
```

$$2$$

```
SUNIndex[2] // StandardForm

(*ExplicitSUNIndex[2]*)
```

```
SUNDelta[i, j] // FCI // StandardForm

(*SUNDelta[SUNIndex[i], SUNIndex[j]]*)
```

2.63 ExplicitSUNFIndex

`ExplicitSUNFIndex[ind]` is a specific $SU(N)$ index in the fundamental representation, i.e. **`ind`** is an integer.

2.63.1 See also

Overview, SUNIndex, SUNFIndex.

2.63.2 Examples

```
ExplicitSUNFIndex[1]
```

$$1$$

```
SUNTF[a, 1, 2]
```

$$T^a_{12}$$

```
SUNTF[a, 1, 2] // FCI // StandardForm

(*SUNTF[{SUNIndex[a]}, ExplicitSUNFIndex[1], ExplicitSUNFIndex[2]]*)
```

2.64 SUNFIndex

SUNFIndex[a] is an $SU(N)$ index in the fundamental representation. If the argument is an integer, **SUNFIndex[a]** turns into **ExplicitSUNFIndex[a]**.

2.64.1 See also

Overview, SUNIndex.

2.64.2 Examples

```
SUNFIndex[i]
```

$$i$$

```
SUNFIndex[i] // StandardForm

(*SUNFIndex[i]*)
```

```
SUNFIndex[2]
```

$$2$$

```
SUNFIndex[2] // StandardForm

(*ExplicitSUNFIndex[2]*)
```

```
SUNFDelta[i, j] // FCI // StandardForm

(*SUNFDelta[SUNFIndex[i], SUNFIndex[j]]*)
```

2.65 FAD

FAD is the FeynCalc external form of **FeynAmpDenominator** and denotes an inverse propagator.

FAD[q, q-p, ...] is $\frac{1}{q^2(q-p)^2...}$.

FAD[{q1, m}, {q1-p, m}, q2, ...] is $\frac{1}{[q1^2-m^2][(q1-p)^2-m^2]q2^2}$. Translation into FeynCalc internal form is performed by **FeynCalcInternal**.

2.65.1 See also

Overview, FAD, FCE, FCI, FeynAmpDenominator, FeynAmpDenominatorSimplify, PropagatorDenominator.

2.65.2 Examples

```
FAD[q, p - q]
```

$$\frac{1}{q^2.(p-q)^2}$$

```
FAD[p, {p - q, m}]
```

$$\frac{1}{p^2.\left((p-q)^2-m^2\right)}$$

```
FAD[{p, 0, 2}, {p - q, m, 3}]
```

$$\frac{1}{\left(p^2\right)^2.\left((p-q)^2-m^2\right)^3}$$

```
FAD[q, p - q] // FCI // StandardForm

(*FeynAmpDenominator[PropagatorDenominator[Momentum[q, D], 0],
PropagatorDenominator[Momentum[p, D] - Momentum[q, D], 0]]*)
```

```
FAD[p] FAD[p - q] // FeynAmpDenominatorCombine[#, FCE -> True] & //
StandardForm

(*FAD[p, p - q]*)
```

2.66 SFAD

SFAD[{{q1 +..., p1 . q2 +..., } {m^2, s}, n}, ...] denotes a Cartesian propagator given by $\frac{1}{[(q_1+...)^2+p_1\cdot q_2...+m^2+si\eta]^n}$, where q_1^2 and $p_1 \cdot q_2$ are Cartesian scalar products in $D-1$ dimensions.

For brevity one can also use shorter forms such as **SFAD[{q1+ ..., m^2}, ...]**, **SFAD[{q1+ ..., m^2 , n}, ...]**, **SFAD[{q1+ ..., {m^2, -1}}, ...]**, **SFAD[q1, ...]** etc.

If **s** is not explicitly specified, its value is determined by the option **EtaSign**, which has the default value **+1**.

If **n** is not explicitly specified, then the default value **1** is assumed. Translation into FeynCalcI internal form is performed by **FeynCalcInternal**, where a **SFAD** is encoded using the special head **CartesianPropagatorDenominator**.

2.66.1 See also

Overview, FAD, GFAD, CFAD.

2.66.2 Examples

```
SFAD[{{p, 0}, m^2}]
```

$$\frac{1}{(p^2 - m^2 + i\eta)}$$

```
SFAD[{{p, 0}, {m^2, -1}}]
```

$$\frac{1}{(p^2 - m^2 - i\eta)}$$

```
SFAD[{{p, 0}, {-m^2, -1}}]
```

$$\frac{1}{(p^2 + m^2 - i\eta)}$$

```
SFAD[{{0, p . q}, m^2}]
```

$$\frac{1}{(p \cdot q - m^2 + i\eta)}$$

```
SFAD[{{0, n . q}}]
```

$$\frac{1}{(n \cdot q + i\eta)}$$

2.67 CFAD

CFAD[{{q1 +..., p1 . q2 +..., } {m^2, s}, n}, ...] denotes a Cartesian propagator given by $\frac{1}{[(q_1+...)^2+p_1\cdot q_2...+m^2+si\eta]^n}$, where q_1^2 and $p_1 \cdot q_2$ are Cartesian scalar products in $D-1$ dimensions.

For brevity one can also use shorter forms such as **CFAD[{q1+ ..., m^2}, ...]**, **CFAD[{q1+ ..., m^2 , n}, ...]**, **CFAD[{q1+ ..., {m^2, -1}}, ...]**, **CFAD[q1, ...]** etc.

If **s** is not explicitly specified, its value is determined by the option **EtaSign**, which has the default value **-1**.

If **n** is not explicitly specified, then the default value **1** is assumed. Translation into FeynCalcI internal form is performed by **FeynCalcInternal**, where a **CFAD** is encoded using the special head **CartesianPropagatorDenominator**.

2.67.1 See also

Overview, FAD, SFAD, GFAD, FeynAmpDenominator.

2.67.2 Examples

```
CFAD[{{p, 0}, m^2}]
```

$$\frac{1}{(p^2+m^2-i\eta)}$$

```
FeynAmpDenominatorExplicit[%]
```

$$\frac{1}{m^2+p^2}$$

```
CFAD[{{p, 0}, {m^2, 1}}]
```

$$\frac{1}{(p^2+m^2+i\eta)}$$

```
FeynAmpDenominatorExplicit[%]
```

$$\frac{1}{m^2+p^2}$$

```
CFAD[{{p, 0}, -m^2}]
```

$$\frac{1}{(p^2 - m^2 - i\eta)}$$

```
FeynAmpDenominatorExplicit[%]
```

$$\frac{1}{p^2 - m^2}$$

```
CFAD[{{0, p . q}, m^2}]
```

$$\frac{1}{(p \cdot q + m^2 - i\eta)}$$

```
FeynAmpDenominatorExplicit[%]
```

$$\frac{1}{m^2 + p \cdot q}$$

```
CFAD[{{0, p . q}}]
```

$$\frac{1}{(p \cdot q - i\eta)}$$

```
FeynAmpDenominatorExplicit[%]
```

$$\frac{1}{p \cdot q}$$

2.68 GFAD

GFAD[{{{x, s}, n}, ...] denotes a generic propagator given by $\frac{1}{[x+si\eta]^n}$, where **x** can be an arbitrary expression. For brevity one can also use shorter forms such as **GFAD[{x, n}, ...]**, **GFAD[{x}, ...]** or **GFAD[x, ...]**.

If s is not explicitly specified, then its value is determined by the option **EtaSign**, which has the default value **+1**.

If **n** is not explicitly specified, then the default value **1** is assumed. Translation into FeynCalc internal form is performed by **FeynCalcInternal**, where a **GFAD** is encoded using the special head **GenericPropagatorDenominator**.

2.68.1 See also

Overview, FAD, SFAD, CFAD.

2.68.2 Examples

```
GFAD[2 z SPD[p1, q] SPD[p2, q] + x SPD[p1, p2]]

FeynAmpDenominatorExplicit[%]

% // FCE // StandardForm
```

$$\frac{1}{(x(\text{p1}\cdot\ \text{p2})+2z(\text{p1}\cdot q)(\text{p2}\cdot q)+i\eta)}$$

$$\frac{1}{2z(\text{p1}\cdot q)(\text{p2}\cdot q)+x(\text{p1}\cdot\ \text{p2})}$$

$$\frac{1}{x\ \text{SPD}[\text{p1},\text{p2}]+2z\ \text{SPD}[\text{p1},q]\ \text{SPD}[\text{p2},q]}$$

2.69 FeynAmpDenominator

`FeynAmpDenominator[...]` represents the inverse denominators of the propagators, i.e. **`FeynAmpDenominator[x]`** is $1/x$. Different propagator denominators are represented using special heads such as **`PropagatorDenominator`**, **`StandardPropagatorDenominator`**, **`CartesianPropagatorDenominator`** etc.

2.69.1 See also

Overview, FAD, SFAD, CFAD, GFAD, FeynAmpDenominatorSimplify.

2.69.2 Examples

The legacy way to represent standard Lorentzian propagators is to use **PropagatorDenominator**. Here the sign of the mass term is fixed to be -1 and no information on the $i\eta$- prescription is available. Furthermore, this way it is not possible to enter eikonal propagators

```
FeynAmpDenominator[PropagatorDenominator[Momentum[p, D], m]]
```

$$\frac{1}{p^2 - m^2}$$

```
FeynAmpDenominator[PropagatorDenominator[Momentum[p, D], m],
  PropagatorDenominator[Momentum[p - q, D], m]]
```

$$\frac{1}{(p^2 - m^2) \, . \, ((p - q)^2 - m^2)}$$

It is worth noting that the Euclidean mass dependence still can be introduced via a trick where the mass symbol is multiplied by the imaginary unit i

```
FeynAmpDenominator[PropagatorDenominator[Momentum[p, D], I m]]

% // FeynAmpDenominatorExplicit
```

$$\frac{1}{p^2 - -m^2}$$

$$\frac{1}{m^2 + p^2}$$

The shortcut to enter **FeynAmpDenominator**s with **PropagatorDenominator**s is **FAD**

```
FAD[p]
```

$$\frac{1}{p^2}$$

```
FAD[{p, m}]
```

$$\frac{1}{p^2 - m^2}$$

```
FAD[{p, m, 3}]
```

$$\frac{1}{(p^2 - m^2)^3}$$

```
FeynAmpDenominator[PropagatorDenominator[Momentum[p, D], m]] // FCE //
StandardForm

(*FAD[{p, m}]*)
```

Since version 9.3, a more flexible input is possible using **StandardPropagatorDenominator**

```
FeynAmpDenominator[StandardPropagatorDenominator[Momentum[p, D], 0, -m^2,
{1, 1}]]
```

$$\frac{1}{(p^2 - m^2 + i\eta)}$$

The mass term can be anything, as long as it does not depend on the loop momenta

```
FeynAmpDenominator[StandardPropagatorDenominator[Momentum[p, D], 0, m^2,
{1, 1}]]
```

$$\frac{1}{(p^2 + m^2 + i\eta)}$$

```
FeynAmpDenominator[StandardPropagatorDenominator[Momentum[p, D], 0, -m^2,
{1, 1}]]
```

$$\frac{1}{(p^2 - m^2 + i\eta)}$$

```
FeynAmpDenominator[StandardPropagatorDenominator[Momentum[p, D], 0, SPD[q,
q], {1, 1}]]
```

$$\frac{1}{(p^2 + q^2 + i\eta)}$$

One can also change the sign of $i\eta$, although currently no internal functions make use of it

```
FeynAmpDenominator[StandardPropagatorDenominator[Momentum[p, D], 0, -m^2,
{1, -1}]]
```

$$\frac{1}{(p^2 - m^2 - i\eta)}$$

The propagator may also be raised to integer or symbolic powers

```
FeynAmpDenominator[StandardPropagatorDenominator[Momentum[p, D], 0, m^2,
{3, 1}]]
```

$$\frac{1}{(p^2 + m^2 + i\eta)^3}$$

```
FeynAmpDenominator[StandardPropagatorDenominator[Momentum[p, D], 0, m^2,
{-2, 1}]]
```

$$(p^2 + m^2 + i\eta)^2$$

```
FeynAmpDenominator[StandardPropagatorDenominator[Momentum[p, D], 0, m^2,
{n, 1}]]
```

$$(p^2 + m^2 + i\eta)^{-n}$$

Eikonal propagators are fully supported

```
FeynAmpDenominator[StandardPropagatorDenominator[0, Pair[Momentum[p, D],
Momentum[q, D]],
   -m^2, {1, 1}]]
```

$$\frac{1}{(p \cdot q - m^2 + i\eta)}$$

```
FeynAmpDenominator[StandardPropagatorDenominator[0, Pair[Momentum[p, D],
Momentum[q, D]],
   0, {1, 1}]]
```

$$\frac{1}{(p \cdot q + i\eta)}$$

FeynCalc keeps trace of the signs of the scalar products in the eikonal propagators. This is where the $i\eta$-prescription may come handy

```
FeynAmpDenominator[StandardPropagatorDenominator[0, -Pair[Momentum[p, D],
Momentum[q, D]],
   0, {1, 1}]]
```

$$\frac{1}{(-p \cdot q + i\eta)}$$

```
FeynAmpDenominator[StandardPropagatorDenominator[0, Pair[Momentum[p, D],
Momentum[q, D]],
   0, {1, -1}]]
```

$$\frac{1}{(p \cdot q - i\eta)}$$

The shortcut to enter **FeynAmpDenominators** with **StandardPropagatorDenominators** is **SFAD**

```
FeynAmpDenominator[StandardPropagatorDenominator[Momentum[p, D], 0, -m^2,
{1, 1}]] // FCE //
  StandardForm

(*SFAD[{{p, 0}, {m^2, 1}, 1}]*)
```

Eikonal propagators are entered using the **Dot** (**.**) as in noncommutative products

```
FeynAmpDenominator[StandardPropagatorDenominator[0, Pair[Momentum[p, D],
      Momentum[q, D]], -m^2, {1, 1}]] // FCE // StandardForm

(*SFAD[{{0, p . q}, {m^2, 1}, 1}]*)
```

The Cartesian version of **StandardPropagatorDenominator** is called **CartesianPropagatorDenominator**. The syntax is almost the same as for **StandardPropagatorDenominator**, except that the momenta and scalar products must be Cartesian.

```
FeynAmpDenominator[CartesianPropagatorDenominator[CartesianMomentum[p, D -
1], 0, m^2,
   {1, -1}]]
```

$$\frac{1}{(p^2+m^2-i\eta)}$$

```
FeynAmpDenominator[CartesianPropagatorDenominator[0,
CartesianPair[CartesianMomentum[p,
     D - 1], CartesianMomentum[q, D - 1]], m^2, {1, -1}]]
```

$$\frac{1}{(p\cdot q+m^2-i\eta)}$$

The shortcut to enter **FeynAmpDenominators** with **CartesianPropagatorDenominators** is **CFAD**

```
FCE[FeynAmpDenominator[CartesianPropagatorDenominator[CartesianMomentum[p,
D - 1], 0, m^2,
     {1, -1}]]] // StandardForm

(*CFAD[{{p, 0}, {m^2, -1}, 1}]*)
```

To represent completely arbitrary propagators one can use **GenericPropagatorDenominator**. However, one should keep in mind that the number of useful manipulations and automatic simplifications available for such propagators is very limited.

```
FeynAmpDenominator[GenericPropagatorDenominator[x, {1, 1}]]
```

$$\frac{1}{(x+i\eta)}$$

This is a nonlinear propagator that appears in the calculation of the QCD Energy-Energy-Correlation function

```
FeynAmpDenominator[GenericPropagatorDenominator[2 z Pair[Momentum[p1, D],
Momentum[Q,
       D]] Pair[Momentum[p2, D], Momentum[Q, D]] - Pair[Momentum[p1, D],
       Momentum[p2, D]], {1, 1}]]
```

$$\frac{1}{(2z(\mathrm{p1}\cdot Q)(\mathrm{p2}\cdot Q)-\mathrm{p1}\cdot\ \mathrm{p2}+i\eta)}$$

The shortcut to enter **FeynAmpDenominator**s with **GenericPropagatorDenominator**s is **GFAD**

```
FeynAmpDenominator[GenericPropagatorDenominator[2 z Pair[Momentum[p1, D],
Momentum[Q,
         D]] Pair[Momentum[p2, D], Momentum[Q, D]] - Pair[Momentum[p1, D],
         Momentum[p2, D]], {1, 1}]] //
   FCE // StandardForm

(*GFAD[{{-SPD[p1, p2] + 2 z SPD[p1, Q] SPD[p2, Q], 1}, 1}]*)
```

2.70 FCGV

FCGV[x] is a FeynCalc global variable, i.e. a container for string variables that allows to introduce new variables without polluting the Global context of Mathematica.

Use the rule **`FCGV[s_] :> ToExpression[s]`** if you want to convert the string **`x`** to a symbol with the name **`x`**.

2.70.1 See also

Overview, FCGV.

2.70.2 Examples

```
FCGV["x"]
```

FCGV(x)

```
FCGV["x"] // InputForm
```

```
FCGV["x"]
```

2.71 FCTensor

`FCTensor` is a data type. E.g. **`DataType[R, FCTensor] = True`**.

2.71.1 See also

Overview, DataType.

2.71.2 Examples

2.72 FCVariable

FCVariable is a data type. E.g. **DataType[z, FCVariable] = True**.

2.72.1 See also

Overview, ExpandScalarProduct, DataType.

2.72.2 Examples

If we want to introduce constants **c1** and **c2**, the naive way doesn't lead to the desired result

```
SPD[c1 p1 + c2 p2, q] // ExpandScalarProduct
```

$$\text{c1 p1} \cdot q + \text{c2 p2} \cdot q$$

The solution is to declare **c1** and **c2** as **FCVariable** so that FeynCalc can distinguish them from the 4-momenta

```
DataType[c1, FCVariable] = True;

DataType[c2, FCVariable] = True;
```

```
SPD[c1 p1 + c2 p2, q] // ExpandScalarProduct
```

$$\text{c1}(\text{p1} \cdot q) + \text{c2}(\text{p2} \cdot q)$$

This works also for propagator denominators and matrices

```
FCI[SFAD[{q + c1 p1, m}]]
```

$$\frac{1}{((\text{c1 p1} + q)^2 - m + i\eta)}$$

```
FCI[SFAD[{q + c1 p1, m}]] // StandardForm

(*FeynAmpDenominator[StandardPropagatorDenominator[c1 Momentum[p1, D] +
Momentum[q, D], 0, -m, {1, 1}]]*)
```

```
GAD[\[Mu]] . (GSD[c1 p] + m) . GAD[\[Nu]] // FCI
```

$$\gamma^{\mu}.(\text{c1}\gamma \cdot p + m).\gamma^{\nu}$$

```
GAD[\[Mu]] . (GSD[c1 p] + m) . GAD[\[Nu]] // FCI // StandardForm

(*DiracGamma[LorentzIndex[\[Mu], D], D] . (m + c1 DiracGamma[Momentum[p,
D], D]) . DiracGamma[LorentzIndex[\[Nu], D], D]*)
```

```
CSI[i] . CSIS[c1 p] . CSI[j] // FCI
```

$$\overline{\sigma}^{i}.(\text{c1}\overline{\sigma} \cdot \overline{p}).\overline{\sigma}^{j}$$

```
CSI[i] . CSIS[c1 p] . CSI[j] // FCI // StandardForm

(*PauliSigma[CartesianIndex[i]] . (c1 PauliSigma[CartesianMomentum[p]]) .
PauliSigma[CartesianIndex[j]]*)
```

To undo the declarations use

```
DataType[c1, FCVariable] = False

DataType[c2, FCVariable] = False
```

False

False

2.73 FreeIndex

FreeIndex is a datatype which is recognized by **Contract**.

Possible use: **DataType[mu, FreeIndex] = True**.

2.73.1 See also

Overview, Contract, DataType.

2.73.2 Examples

2.74 GrassmannParity

GrassmannParity is a data type.

E.g. **DataType[F, GrassmannParity] = 1** declares **F** to be of bosonic type and **DataType[F, GrassmannParity] = -1** of fermionic one.

2.74.1 See also

Overview, DataType.

2.74.2 Examples

2.75 ImplicitDiracIndex

ImplicitDiracIndex is a data type. It mainly applies to names of quantum fields specifying that the corresponding field carries an implicit Dirac index.

This information can be supplied e.g. via **DataType[QuarkField, ImplicitDiracIndex] = True**, where **QuarkField** is a possible name of the relevant field.

The **ImplicitDiracIndex** property becomes relevant when simplifying noncommutative products involving **QuantumField**s via **ExpandPartialD**, **DotSimplify**.

2.75.1 See also

Overview, DataType, ImplicitSUNFIndex, ImplicitPauliIndex

2.75.2 Examples

Default (possibly unwanted) behavior

```
ex = QuantumField[AntiQuarkField] . GA[\[Mu]] . QuantumField[QuarkField]
```

$$\bar{\psi}.\bar{\gamma}^{\mu}.\psi$$

```
ExpandPartialD[ex]
```

$$\bar{\gamma}^{\mu}.\bar{\psi}.\psi$$

Now we let FeynCalc know that **AntiQuarkField** and **QuarkField** carry an implicit Dirac index that connects them to the Dirac matrix.

```
DataType[QuarkField, ImplicitDiracIndex] = True;
DataType[AntiQuarkField, ImplicitDiracIndex] = True;
```

```
ExpandPartialD[ex]
```

$$\bar{\psi}.\bar{\gamma}^{\mu}.\psi$$

2.76 ImplicitPauliIndex

ImplicitPauliIndex is a data type. It mainly applies to names of quantum fields specifying that the corresponding field carries an implicit Pauli index.

This information can be supplied e.g. via **DataType[QuarkFieldChi, ImplicitPauliIndex] = True**, where **QuarkFieldChi** is a possible name of the relevant field.

The **ImplicitDiracIndex** property becomes relevant when simplifying noncommutative products involving **QuantumField**s via **ExpandPartialD**, **DotSimplify**.

2.76.1 See also

Overview, DataType, ImplicitSUNFIndex, ImplicitDiracIndex

2.76.2 Examples

Default (possibly unwanted) behavior

```
ex = QuantumField[QuarkFieldChiDagger] . CSI[i] .
QuantumField[QuarkFieldChi]
```

$$\chi^{\dagger}.\overline{\sigma}^{i}.\chi$$

```
ExpandPartialD[ex]
```

$$\overline{\sigma}^{i}.\chi^{\dagger}.\chi$$

Now we let FeynCalc know that **QuarkFieldChiDagger** and **QuarkFieldChi** carry an implicit Pauli index that connects them to the Pauli matrix.

```
DataType[QuarkFieldChi, ImplicitPauliIndex] = True;
DataType[QuarkFieldChiDagger, ImplicitPauliIndex] = True;
```

```
ExpandPartialD[ex]
```

$$\chi^{\dagger}.\overline{\sigma}^{i}.\chi$$

2.77 ImplicitSUNFIndex

ImplicitSUNFIndex is a data type. It mainly applies to names of quantum fields specifying that the corresponding field carries an implicit $SU(N)$ index in the fundamental representation.

This information can be supplied e.g. via **DataType[QuarkField, ImplicitSUNFIndex] = True**, where **QuarkField** is a possible name of the relevant field.

The **ImplicitSUNFIndex** property becomes relevant when simplifying noncommutative products involving **QuantumField**s via **ExpandPartialD**, **DotSimplify**.

2.77.1 See also

Overview, DataType, ImplicitDiracIndex, ImplicitPauliIndex

2.77.2 Examples

Default (possibly unwanted) behavior

```
ex = QuantumField[AntiQuarkField] . SUNT[a] . QuantumField[QuarkField]
```

$$\bar{\psi}.T^{a}.\psi$$

```
DotSimplify[ex]
```

$$\bar{\psi}.\psi T^{a}$$

```
ExpandPartialD[ex]
```

$$\bar{\psi}.\psi T^{a}$$

Now we let FeynCalc know that **AntiQuarkField** and **QuarkField** carry an implicit color index that connects them to the color matrix.

```
DataType[QuarkField, ImplicitSUNFIndex] = True;
DataType[AntiQuarkField, ImplicitSUNFIndex] = True;
```

```
DotSimplify[ex]
```

$$\bar{\psi}.T^a.\psi$$

```
ExpandPartialD[ex]
```

$$\bar{\psi}.T^a.\psi$$

2.78 NegativeInteger

NegativeInteger is a data type. E.g. **DataType[n, NegativeInteger]** can be set to **True**.

2.78.1 See also

Overview, DataType.

2.78.2 Examples

2.79 NonCommutative

NonCommutative is a data type which may be used, e.g., as **DataType[x, NonCommutative] = True**.

2.79.1 See also

Overview, DataType, DeclareNonCommutative.

2.79.2 Examples

2.80 PositiveInteger

PositiveInteger is a data type. E.g. **DataType[OPEm, PositiveInteger]** gives **True**.

2.80.1 See also

Overview, DataType.

2.80.2 Examples

2.81 PositiveNumber

PositiveNumber is a data type. E.g. **DataType[Epsilon, PositiveNumber] = True** (by default).

2.81.1 See also

Overview, DataType.

2.81.2 Examples

2.82 FUNCTION

FUNCTION[exp, string] is a head of an expression to be declared a function (of type **String**), if used in **Write2**.

2.82.1 See also

Overview, Write2.

2.82.2 Examples

2.83 DCHN

DCHN[x, i, j] is a chain of Dirac matrices x and is transformed into **DiracChain[FCI[x], DiracIndex[i], DiracIndex[j]]** by **FeynCalcInternal**.

2.83.1 See also

Overview, DiracChain, DCHN, DiracIndex, DiracIndexDelta, DiracChainJoin, DiracChainExpand, DiracChainFactor.

2.83.2 Examples

A standalone Dirac matrix with open Dirac indices

```
DCHN[GAD[\[Mu]], i, j]
```

$$(\gamma^{\mu})_{ij}$$

A chain of Dirac matrices with open Dirac indices

```
DCHN[GAD[\[Mu]] . GAD[\[Nu]], i, j]
```

$$(\gamma^{\mu}.\gamma^{\nu})_{ij}$$

A single $\bar{u}$ spinor with an open Dirac index

```
DCHN[SpinorUBar[p, m], i]
```

$$(\bar{u}(p,m))_i$$

A single $\bar{v}$ spinor with an open Dirac index

```
DCHN[SpinorVBar[p, m], i]
```

$$(\bar{v}(p,m))_i$$

A single u spinor with an open Dirac index

```
DCHN[i, SpinorU[p, m]]
```

$$(u(p,m))_i$$

A single v spinor with an open Dirac index

```
DCHN[i, SpinorV[p, m]]
```

$$(v(p,m))_i$$

$\bar{u}$ spinor contracted with a chain of Dirac matrices

```
DCHN[GAD[\[Mu]] . GAD[\[Nu]], SpinorUBar[p, m], j]
```

$$(\bar{u}(p,m).\gamma^{\mu}.\gamma^{\nu})_{j}$$

$\bar{v}$ spinor contracted with a chain of Dirac matrices

```
DCHN[GAD[\[Mu]] . GAD[\[Nu]], SpinorVBar[p, m], j]
```

$$(\bar{v}(p,m).\gamma^{\mu}.\gamma^{\nu})_{j}$$

u spinor contracted with a chain of Dirac matrices

```
DCHN[GAD[\[Mu]] . GAD[\[Nu]], i, SpinorU[p, m]]
```

$$(\gamma^{\mu}.\gamma^{\nu}.u(p,m))_{i}$$

v spinor contracted with a chain of Dirac matrices

```
DCHN[GAD[\[Mu]] . GAD[\[Nu]], i, SpinorV[p, m]]
```

$$(\gamma^{\mu}.\gamma^{\nu}.v(p,m))_{i}$$

2.84 DiracChain

`DiracChain[x, i, j]` denotes a chain of Dirac matrices **`x`**, where the Dirac indices **`i`** and **`j`** are explicit.

2.84.1 See also

Overview, DiracChain, DCHN, DiracIndex, DiracIndexDelta, DiracChainJoin, DiracChainExpand, DiracChainFactor.

2.84.2 Examples

A standalone Dirac matrix

```
DiracChain[DiracGamma[LorentzIndex[\[Mu]]], DiracIndex[i], DiracIndex[j]]
```

$$(\bar{\gamma}^{\mu})_{ij}$$

A chain of Dirac matrices with open indices

```
DiracChain[DiracGamma[LorentzIndex[\[Mu], D], D] .
DiracGamma[LorentzIndex[\[Nu], D], D], DiracIndex[i], DiracIndex[j]]
```

$$(\gamma^{\mu}.\gamma^{\nu})_{ij}$$

A DiracChain with only two arguments denotes a spinor component

```
DiracChain[Spinor[Momentum[p], m], DiracIndex[i]]
```

$$(\varphi(\overline{p}, m))_i$$

```
DiracChain[Spinor[Momentum[-p], m], DiracIndex[i]]
```

$$(\varphi(-\overline{p}, m))_i$$

```
DiracChain[DiracIndex[i], Spinor[Momentum[p], m]]
```

$$(\varphi(\overline{p}, m))_i$$

```
DiracChain[DiracIndex[i], Spinor[Momentum[-p], m]]
```

$$(\varphi(-\overline{p}, m))_i$$

The chain may also be partially open or closed

```
DiracChain[DiracGamma[LorentzIndex[\[Mu]]] . (m + DiracGamma[Momentum[p]])
. DiracGamma[LorentzIndex[\[Nu]]], Spinor[Momentum[p], m, 1],
DiracIndex[j]]
```

$$(\varphi(\overline{p},m).\bar{\gamma}^{\mu}.(\bar{\gamma}\cdot\overline{p}+m).\bar{\gamma}^{\nu})_{j}$$

```
DiracChain[DiracGamma[LorentzIndex[\[Mu]]] . (m + DiracGamma[Momentum[p]])
. DiracGamma[LorentzIndex[\[Nu]]], DiracIndex[i], Spinor[Momentum[p], m,
1]]
```

$$(\bar{\gamma}^{\mu}.(\bar{\gamma}\cdot\overline{p}+m).\bar{\gamma}^{\nu}.\varphi(\overline{p},m))_{i}$$

```
DiracChain[DiracGamma[LorentzIndex[\[Mu]]] . (m + DiracGamma[Momentum[p]])
. DiracGamma[LorentzIndex[\[Nu]]], Spinor[Momentum[p1], m1, 1],
Spinor[Momentum[p2], m2, 1]]
```

$$\left(\varphi(\overline{\mathrm{p1}},\mathrm{m1}).\bar{\gamma}^{\mu}.(\bar{\gamma}\cdot\overline{p}+m).\bar{\gamma}^{\nu}.\varphi(\overline{\mathrm{p2}},\mathrm{m2})\right)$$

```
DiracChain[1, Spinor[Momentum[p1], m1, 1], Spinor[Momentum[p2], m2, 1]]
```

$$\left(\varphi(\overline{\mathrm{p1}},\mathrm{m1}).\varphi(\overline{\mathrm{p2}},\mathrm{m2})\right)$$

2.85 FeynAmp

`FeynAmp[q, amp]` is the head of a Feynman amplitude, where amp denotes the analytical expression for the amplitude and q is the integration variable. **`FeynAmp[q1, q2, amp]`** denotes a two-loop amplitude. **`FeynAmp`** has no functional properties and serves just as a head. There are however special typesetting rules attached.

2.85.1 See also

Overview, Amplitude.

2.85.2 Examples

This is a 1-loop gluon self-energy amplitude (ignoring factors of 2π).

```
FeynAmp[q, GV[p, \[Mu], a, q - p, \[Alpha], c, -q, \[Beta], e] GP[p - q,
\[Alpha], c, \[Rho], d] GV[-p, \[Nu], b, p - q, \[Rho], d, q, \[Sigma], f]
*
   GP[q, \[Beta], e, \[Sigma], f]]
```

$$\int d^D q \left(f^{bdf} f^{ace} \Pi^{\beta\sigma}_{ef}(q) V^{\mu\alpha\beta}(p, q-p, -q) V^{\nu\rho\sigma}(-p, p-q, q) \Pi^{\alpha\rho}_{cd}(p-q) \right)$$

This is a generic 2-loop amplitude.

```
FeynAmp[Subscript[q, 1], Subscript[q, 2], anyexpression]
```

$$\text{FeynAmp}\,(q_1, q_2, \text{anyexpression})$$

2.86 FeynAmpList

FeynAmpList[info][FeynAmp[...], FeynAmp[...], ...] is a head of a list of Feynman amplitudes. **FeynAmpList** has no functional properties and serves just as a head.

2.86.1 See also

Overview, FeynAmp.

2.86.2 Examples

2.87 FCPartialD

FCPartialD[ind] denotes a partial derivative of a field. It is an internal object that may appear only inside a **QuantumField**.

FCPartialD[LorentzIndex[mu]] denotes ∂_μ.

FCPartialD[LorentzIndex[mu , D]] denotes the D-dimensional ∂_μ.

FCPartialD[CartesianIndex[i]] denotes $\partial^i = -\nabla^i$.

If you need to specify a derivative with respect to a particular variable it also possible to use **FCPartialD[{ LorentzIndex[mu], y}]** or **FCPartialD[{CartesianIndex[i], x}]**although this notation is still somewhat experimental

2.87.1 See also

Overview, ExpandPartialD, LeftPartialD, LeftRightPartialD, RightPartialD.

2.87.2 Examples

```
QuantumField[A, {\[Mu]}] . LeftPartialD[\[Nu]]

ex = ExpandPartialD[%]
```

$$A_{\mu}.\overleftarrow{\partial}_{\nu}$$

$$(\partial_{\nu} A_{\mu})$$

```
ex // StandardForm

(*QuantumField[FCPartialD[LorentzIndex[\[Nu]]], A, LorentzIndex[\[Mu]]]*)
```

```
RightPartialD[{CartesianIndex[i], x}] . QuantumField[S, x]

ex = ExpandPartialD[%]
```

$$\vec{\partial}_{\{i,x\}}.S^{x}$$

$$\left(\partial_{\{i,x\}} S^{x}\right)$$

```
ex // StandardForm

(*QuantumField[FCPartialD[{CartesianIndex[i], x}], S, x]*)
```

FCPartialD also accepts **FCGV** symbols as arguments, which can be sometimes useful to make the final expression look nicer.

```
QuantumField[FCPartialD[FCGV["\[Del]"]], S, x]
```

$$(\nabla S^{x})$$

2.88 LeftNablaD

LeftNablaD[i] denotes $\overleftarrow{\nabla}_i$ acting to the left.

2.88.1 See also

Overview, ExpandPartialD, FCPartialD, LeftRightNablaD, RightNablaD.

2.88.2 Examples

```
QuantumField[A, LorentzIndex[\[Mu]]] . LeftNablaD[i]

ex = ExpandPartialD[%]
```

$$A_\mu .\overleftarrow{\nabla}^i$$

$$-\left(\partial_i A_\mu\right)$$

```
ex // StandardForm

(*-QuantumField[FCPartialD[CartesianIndex[i]], A, LorentzIndex[\[Mu]]]*)
```

```
StandardForm[LeftNablaD[i]]

(*LeftNablaD[CartesianIndex[i]]*)
```

```
QuantumField[A, LorentzIndex[\[Mu]]] . QuantumField[A, LorentzIndex[\[Nu]]]
. LeftNablaD[i]

ex = ExpandPartialD[%]
```

$$A_\mu .A_\nu .\overleftarrow{\nabla}^i$$

$$-A_\mu .\left(\partial_i A_\nu\right)-\left(\partial_i A_\mu\right).A_\nu$$

```
ex // StandardForm

(*-QuantumField[A, LorentzIndex[\[Mu]]] .
QuantumField[FCPartialD[CartesianIndex[i]], A, LorentzIndex[\[Nu]]] -
QuantumField[FCPartialD[CartesianIndex[i]], A, LorentzIndex[\[Mu]]] .
QuantumField[A, LorentzIndex[\[Nu]]]*)
```

2.89 LeftRightNablaD

`LeftRightNablaD[i]` denotes $\overleftrightarrow{\nabla}_i$, acting to the left and right.

`ExplicitPartialD[LeftRightNablaD[i]]` gives **`1/2 (RightNablaD[i] - LeftNablaD[i])`**.

2.89.1 See also

Overview, ExplicitPartialD, ExpandPartialD, FCPartialD, LeftNablaD, LeftRightNablaD2, RightNablaD.

2.89.2 Examples

```
LeftRightNablaD[i]

ExplicitPartialD[%]
```

$$\overleftrightarrow{\nabla}_i$$

$$\frac{1}{2}\overleftarrow{\partial}_i - \vec{\partial}_i$$

```
LeftRightNablaD[i] . QuantumField[A, LorentzIndex[\[Nu]]]

ExpandPartialD[%]
```

$$\overleftrightarrow{\nabla}_i.A_\nu$$

$$\frac{1}{2}\left(\overleftarrow{\partial}_i.A_\nu - (\partial_i A_\nu)\right)$$

```
QuantumField[A, LorentzIndex[\[Mu]]] . LeftRightNablaD[i] . QuantumField[A,
LorentzIndex[\[Rho]]]

ExpandPartialD[%]
```

$$A_\mu.\overleftrightarrow{\nabla}_i.A_\rho$$

$$\frac{1}{2}\left((\partial_i A_\mu).A_\rho - A_\mu.(\partial_i A_\rho)\right)$$

2.90 LeftRightNablaD2

`LeftRightNablaD2[mu]` denotes $\overleftrightarrow{\nabla}_i$, acting to the left and right.

`ExplicitPartialD[LeftRightNablaD2[mu]]` gives **`(RightNablaD[i] + LeftNablaD[i])`**.

2.90.1 See also

Overview, ExplicitPartialD, ExpandPartialD, FCPartialD, LeftNablaD, RightNablaD.

2.90.2 Examples

```
LeftRightNablaD2[i]

ExplicitPartialD[%]
```

$$\overleftrightarrow{\nabla}_i$$

$$-\overleftarrow{\partial}_i - \vec{\partial}_i$$

```
LeftRightNablaD2[i] . QuantumField[A, LorentzIndex[\[Nu]]]

ExpandPartialD[%]
```

$$\overleftrightarrow{\nabla}_i.A_\nu$$

$$-(\partial_i A_\nu) - \overleftarrow{\partial}_i.A_\nu$$

```
QuantumField[A, LorentzIndex[\[Mu]]] . LeftRightNablaD2[i] .
QuantumField[A, LorentzIndex[\[Rho]]]

ExpandPartialD[%]
```

$$A_\mu.\overleftrightarrow{\nabla}_i.A_\rho$$

$$-A_\mu.(\partial_i A_\rho) - (\partial_i A_\mu).A_\rho$$

2.91 LeftPartialD

`LeftPartialD[mu]` denotes $\overleftarrow{\partial}_{\mu}$ acting to the left.

2.91.1 See also

Overview, ExpandPartialD, FCPartialD, LeftRightPartialD, RightPartialD.

2.91.2 Examples

```
QuantumField[A, LorentzIndex[\[Mu]]] . LeftPartialD[\[Nu]]

ex = ExpandPartialD[%]
```

$$A_{\mu}.\overleftarrow{\partial}_{\nu}$$

$$(\partial_{\nu} A_{\mu})$$

```
ex // StandardForm

(*QuantumField[FCPartialD[LorentzIndex[\[Nu]]], A, LorentzIndex[\[Mu]]]*)
```

```
StandardForm[LeftPartialD[\[Mu]]]

(*LeftPartialD[LorentzIndex[\[Mu]]]*)
```

```
QuantumField[A, LorentzIndex[\[Mu]]] . QuantumField[A, LorentzIndex[\[Nu]]]
. LeftPartialD[\[Rho]]

ex = ExpandPartialD[%]
```

$$A_{\mu}.A_{\nu}.\overleftarrow{\partial}_{\rho}$$

$$A_{\mu}.((\partial_{\rho} A_{\nu})) + ((\partial_{\rho} A_{\mu})).A_{\nu}$$

```
ex // StandardForm

(*QuantumField[A, LorentzIndex[\[Mu]]] .
QuantumField[FCPartialD[LorentzIndex[\[Rho]]], A, LorentzIndex[\[Nu]]] +
QuantumField[FCPartialD[LorentzIndex[\[Rho]]], A, LorentzIndex[\[Mu]]] .
QuantumField[A, LorentzIndex[\[Nu]]]*)
```

2.92 LeftRightPartialD

`LeftRightPartialD[mu]` denotes $\overleftrightarrow{\partial}_{\mu}$, acting to the left and right.

`ExplicitPartialD[LeftRightPartialD[mu]]` gives **`1/2 (RightPartialD[mu] - LeftPartialD[mu])`**.

2.92.1 See also

Overview, ExplicitPartialD, ExpandPartialD, FCPartialD, LeftPartialD, LeftRightPartialD2, RightPartialD.

2.92.2 Examples

```
LeftRightPartialD[\[Mu]]

ExplicitPartialD[%]
```

$$\overleftrightarrow{\partial}_{\mu}$$

$$\frac{1}{2}\left(\vec{\partial}_{\mu} - \overleftarrow{\partial}_{\mu}\right)$$

```
LeftRightPartialD[\[Mu]] . QuantumField[A, LorentzIndex[\[Nu]]]

ExpandPartialD[%]
```

$$\overleftrightarrow{\partial}_{\mu}.A_{\nu}$$

$$\frac{1}{2}\left((\partial_{\mu}A_{\nu}) - \overleftarrow{\partial}_{\mu}.A_{\nu}\right)$$

```
QuantumField[A, LorentzIndex[\[Mu]]] . LeftRightPartialD[\[Nu]] .
QuantumField[A, LorentzIndex[\[Rho]]]

ExpandPartialD[%]
```

$$A_{\mu}.\overleftrightarrow{\partial}_{\nu}.A_{\rho}$$

$$\frac{1}{2}\left(A_{\mu}.\left((\partial_{\nu}A_{\rho})\right) - \left((\partial_{\nu}A_{\mu})\right).A_{\rho}\right)$$

2.93 LeftRightPartialD2

`LeftRightPartialD2[mu]` denotes $\overleftrightarrow{\partial}_{\mu}$, acting to the left and right.

`ExplicitPartialD[LeftRightPartialD2[mu]]` gives **`(RightPartialD[mu] + LeftPartialD[mu])`**.

2.93.1 See also

Overview, ExplicitPartialD, ExpandPartialD, FCPartialD, LeftPartialD, RightPartialD.

2.93.2 Examples

```
LeftRightPartialD2[\[Mu]]

ExplicitPartialD[%]
```

$$\overleftrightarrow{\partial}_{\mu}$$

$$\overleftarrow{\partial}_{\mu} + \vec{\partial}_{\mu}$$

```
LeftRightPartialD2[\[Mu]] . QuantumField[A, LorentzIndex[\[Nu]]]

ExpandPartialD[%]
```

$$\overleftrightarrow{\partial}_{\mu}.A_{\nu}$$

$$(\partial_{\mu} A_{\nu}) + \overleftarrow{\partial}_{\mu}.A_{\nu}$$

```
QuantumField[A, LorentzIndex[\[Mu]]] . LeftRightPartialD2[\[Nu]] .
QuantumField[A, LorentzIndex[\[Rho]]]

ExpandPartialD[%]
```

$$A_{\mu}.\overleftrightarrow{\partial}_{\nu}.A_{\rho}$$

$$A_{\mu}.((\partial_{\nu} A_{\rho})) + ((\partial_{\nu} A_{\mu})).A_{\rho}$$

2.94 RightNablaD

RightNablaD[i] denotes ∇_i, acting to the right.

2.94.1 See also

Overview, ExpandPartialD, FCPartialD, LeftNablaD.

2.94.2 Examples

```
RightNablaD[i]
```

$$\vec{\nabla}^i$$

```
RightNablaD[i] . QuantumField[A, LorentzIndex[\[Mu]]]

ex = ExpandPartialD[%]
```

$$\vec{\nabla}^i.A_\mu$$

$$-\left(\partial_i A_\mu\right)$$

```
ex // StandardForm

(*-QuantumField[FCPartialD[CartesianIndex[i]], A, LorentzIndex[\[Mu]]]*)
```

```
RightNablaD[i] // StandardForm

(*RightNablaD[CartesianIndex[i]]*)
```

2.95 RightPartialD

RightPartialD[mu] denotes ∂_μ, acting to the right.

2.95.1 See also

Overview, ExpandPartialD, FCPartialD, LeftPartialD.

2.95.2 Examples

```
RightPartialD[\[Mu]]
```

$$\vec{\partial}_{\mu}$$

```
RightPartialD[\[Mu]] . QuantumField[A, LorentzIndex[\[Mu]]]

ex = ExpandPartialD[%]
```

$$\vec{\partial}_{\mu}.A_{\mu}$$

$$(\partial_{\mu}A_{\mu})$$

```
ex // StandardForm

(*QuantumField[FCPartialD[LorentzIndex[\[Mu]]], A, LorentzIndex[\[Mu]]]*)
```

```
RightPartialD[\[Mu]] // StandardForm

(*RightPartialD[LorentzIndex[\[Mu]]]*)
```

2.96 FCTopology

`FCTopology[id, {prop1, prop2, ...}, {l1, l2, ...}, {p1, p2, ...}, {kRule1, kRule2, ...}, {}]` denotes a topology with the identifier **`id`** that is characterized by the propagators **`{prop1, prop2, ...}`**. The propagators in the list do not necessarily have to form a valid basis, i.e. the basis may also be incomplete or overdetermined. The lists **`{l1, l2, ...}`** and **`{p1, p2, ...}`** stand for the loop and external momenta respectively. Furthermore, {kRule1, kRule2, ...} denotes replacement rules for kinematic invariants.

The last argument (an empty list) is reserved for future improvements.

2.96.1 See also

Overview, FCLoopValidTopologyQ, GLI.

2.96.2 Examples

A 2-loop topology with one external momentum **`Q`**

```
FCTopology[topo1, {SFAD[p1], SFAD[p2], SFAD[Q - p1 - p2], SFAD[Q - p2],
SFAD[Q - p1]}, {p1, p2}, {Q}, {}, {}]
```

$$\text{FCTopology}\left(\text{topo1}, \left\{\frac{1}{(\text{p1}^2+i\eta)}, \frac{1}{(\text{p2}^2+i\eta)}, \frac{1}{((-\text{p1}-\text{p2}+Q)^2+i\eta)}, \frac{1}{((Q-\text{p2})^2+i\eta)}, \frac{1}{((Q-\text{p1})^2+i\eta)}\right\}, \{\text{p1}, \text{p2}\}, \{Q\}, \{\}, \{\}\right)$$

A 3-loop topology with one external momentum **Q**

```
topo = FCTopology[topo2, {SFAD[p1], SFAD[p2], SFAD[p3], SFAD[Q - p1 - p2 -
p3], SFAD[Q - p1 - p2],
    SFAD[Q - p1], SFAD[Q - p2], SFAD[p1 + p3], SFAD[p2 + p3]}, {p1, p2,
p3}, {Q}, {}, {}]
```

$$\text{FCTopology}\left(\text{topo2}, \left\{\frac{1}{(\text{p1}^2+i\eta)}, \frac{1}{(\text{p2}^2+i\eta)}, \frac{1}{(\text{p3}^2+i\eta)}, \frac{1}{((-\text{p1}-\text{p2}-\text{p3}+Q)^2+i\eta)}, \frac{1}{((-\text{p1}-\text{p2}+Q)^2+i\eta)}, \frac{1}{((Q-\text{p1})^2+i\eta)}, \frac{1}{((Q-\text{p2})^2+i\eta)}, \frac{1}{((\text{p1}+\text{p3})^2+i\eta)}, \frac{1}{((\text{p2}+\text{p3})^2+i\eta)}\right\}, \{\text{p1}, \text{p2}, \text{p3}\}, \{Q\}, \{\}, \{\}\right)$$

Use **FCLoopValidTopologyQ** to check if the syntax of the given topology is correct.

```
FCLoopValidTopologyQ[topo]
```

True

2.97 FV

FV[p, mu] is the 4-dimensional vector p^μ.

2.97.1 See also

Overview, FCE, FCI, FVD, Pair.

2.97.2 Examples

```
FV[p, \[Mu]]
```

$$\overline{p}^{\mu}$$

```
FV[p - q, \[Mu]]
```

$$(\overline{p} - \overline{q})^{\mu}$$

```
FV[p, \[Mu]] // StandardForm

(*FV[p, \[Mu]]*)
```

```
FCI[FV[p, \[Mu]]] // StandardForm

(*Pair[LorentzIndex[\[Mu]], Momentum[p]]*)
```

ExpandScalarProduct is used to expand momenta in **FV**

```
ExpandScalarProduct[FV[p - q, \[Mu]]]
```

$$\overline{p}^{\mu} - \overline{q}^{\mu}$$

2.98 FVD

FVD[p, mu] is the D-dimensional vector p with Lorentz index **mu**.

2.98.1 See also

Overview, FCE, FCI, FV, Pair.

2.98.2 Examples

```
FVD[p, \[Mu]]
```

$$p^{\mu}$$

```
FVD[p - q, \[Mu]]
```

$$(p-q)^{\mu}$$

```
FVD[p, \[Mu]] // StandardForm

(*FVD[p, \[Mu]]*)
```

```
FCI[FVD[p, \[Mu]]] // StandardForm

(*Pair[LorentzIndex[\[Mu], D], Momentum[p, D]]*)
```

There is no special function to expand momenta in **FVD**.

```
ex = ExpandScalarProduct[FVD[p - q, \[Mu]]]
```

$$p^{\mu}-q^{\mu}$$

```
ex // StandardForm

(*Pair[LorentzIndex[\[Mu], D], Momentum[p, D]] - Pair[LorentzIndex[\[Mu],
D], Momentum[q, D]]*)
```

2.99 FVE

FVE[p, mu] is the $D-4$-dimensional vector p with Lorentz index μ.

2.99.1 See also

Overview, FCE, FCI, FV, FVD, Pair.

2.99.2 Examples

```
FVE[p, \[Mu]]
```

$$\hat{p}^{\mu}$$

```
FVE[p - q, \[Mu]]
```

$$(\hat{p} - \hat{q})^{\mu}$$

```
FVE[p, \[Mu]] // StandardForm

(*FVE[p, \[Mu]]*)
```

```
FCI[FVE[p, \[Mu]]] // StandardForm

(*Pair[LorentzIndex[\[Mu], -4 + D], Momentum[p, -4 + D]]*)
```

There is no special function to expand momenta in **FVE**.

```
ex = ExpandScalarProduct[FVE[p - q, \[Mu]]]
```

$$\hat{p}^{\mu} - \hat{q}^{\mu}$$

```
ex // StandardForm

(*Pair[LorentzIndex[\[Mu], -4 + D], Momentum[p, -4 + D]] -
Pair[LorentzIndex[\[Mu], -4 + D], Momentum[q, -4 + D]]*)
```

```
Contract[FVE[p, \[Mu]] FV[q, \[Mu]]]
```

$$0$$

2.100 GaugeField

GaugeField is just a name. No functional properties are associated with it. **GaugeField** is used as default setting for the option **QuantumField** of **FieldStrength**.

2.100.1 See also

Overview, FieldStrength, QuantumField.

2.100.2 Examples

```
GaugeField
```

$$A$$

```
QuantumField[GaugeField, LorentzIndex[\[Mu]], SUNIndex[a]]
```

$$A_{\mu}^{a}$$

2.101 GaugeXi

GaugeXi is a head for gauge parameters.

2.101.1 See also

Overview, Gauge, GaugeField.

2.101.2 Examples

2.102 GluonField

GluonField is a name of a gauge field.

2.102.1 See also

Overview, GaugeField.

2.102.2 Examples

```
GluonField
```

$$A$$

```
QuantumField[GluonField, LorentzIndex[\[Mu]], SUNIndex[a]]
```

$$A_{\mu}^{a}$$

2.103 IFPD

IFPD[p, m] denotes **(p^2 - m^2)**.

2.103.1 See also

Overview, PropagatorDenominator.

2.103.2 Examples

2.104 KD

KD[i, j] is the Kronecker delta in 3 dimensions.

2.104.1 See also

Overview, CartesianPair, KDD.

2.104.2 Examples

```
KD[i, j]
```

$$\bar{\delta}^{ij}$$

```
Contract[KD[i, j] KD[i, j]]
```

$$3$$

```
KD[a, b] // StandardForm

(*KD[a, b]*)
```

```
FCI[KD[a, b]] // StandardForm

(*CartesianPair[CartesianIndex[a], CartesianIndex[b]]*)
```

```
FCE[FCI[KD[a, b]]] // StandardForm

(*KD[a, b]*)
```

2.105 KDD

KDD[i, j] is the Kronecker delta in $D - 1$ dimensions.

2.105.1 See also

Overview, CartesianPair, KD.

2.105.2 Examples

```
KDD[i, j]
```

$$\delta^{ij}$$

```
Contract[KDD[i, j] KDD[i, j]]
```

$$D - 1$$

```
KDD[a, b] // StandardForm

(*KDD[a, b]*)
```

```
FCI[KDD[a, b]] // StandardForm

(*CartesianPair[CartesianIndex[a, -1 + D], CartesianIndex[b, -1 + D]]*)
```

```
FCE[FCI[KDD[a, b]]] // StandardForm

(*KDD[a, b]*)
```

2.106 KDE

KDE[i, j] is the Kronecker delta in $D - 4$ dimensions.

2.106.1 See also

Overview, CartesianPair, KD, KDD.

2.106.2 Examples

```
KDE[i, j]
```

$$\hat{\delta}^{ij}$$

```
Contract[KDE[i, j] KDE[i, j]]
```

$$D - 4$$

```
Contract[KDE[i, j] KD[i, j]]
```

$$0$$

```
Contract[KDE[i, j] KDD[i, j]]
```

$$D - 4$$

```
KDE[i, j] // StandardForm

(*KDE[i, j]*)
```

```
FCI[KDE[i, j]] // StandardForm

(*CartesianPair[CartesianIndex[i, -4 + D], CartesianIndex[j, -4 + D]]*)
```

```
FCE[FCI[KDE[i, j]]] // StandardForm

(*KDE[i, j]*)
```

2.107 Li2

`Li2` is an abbreviation for the dilogarithm function, i.e. **`Li2 = PolyLog[2, #]&`**.

2.107.1 See also

Overview, Li3, Li4.

2.107.2 Examples

```
Li2[x]
```

$$\text{Li}_2(x)$$

```
Li2 // StandardForm

(*PolyLog[2, #1] &*)
```

```
Integrate[-Log[1 - x]/x, x]
```

$$\text{Li}_2(x)$$

2.108 Li3

Li3 is an abbreviation for the trilogarithm function, i.e. **Li3 = PolyLog[3, #]&**.

2.108.1 See also

Overview, Li2.

2.108.2 Examples

```
Li3[x]
```

$$\text{Li}_3(x)$$

```
Li3 // StandardForm

(*PolyLog[3, #1] &*)
```

```
D[Li3[x], x]
```

$$\frac{\text{Li}_2(x)}{x}$$

```
Integrate[Li3[x]/x, x]
```

$$\text{Li}_4(x)$$

2.109 Li4

Li4 is an abbreviation for the weight 4 polylogarithm function, i.e. **Li4 = PolyLog[4, #]&**.

2.109.1 See also

Overview, Li2, Li3, SimplifyPolyLog.

2.109.2 Examples

```
Li4[x]
```

$$\text{Li}_4(x)$$

```
Li4 // StandardForm

(*PolyLog[4, #1] &*)
```

```
D[Li4[x], x]
```

$$\frac{\text{Li}_3(x)}{x}$$

```
Integrate[Li3[x]/x, x]
```

$$\text{Li}_4(x)$$

2.110 Momentum

Momentum[p] is the head of a four momentum **p**.

The internal representation of a 4-dimensional p is **Momentum[p]**.

For other than 4 dimensions: **Momentum[p, dim]**.

Momentum[p, 4] simplifies to **Momentum[p]**.

2.110.1 See also

Overview, DiracGamma, Eps, LorentzIndex, MomentumExpand.

2.110.2 Examples

This is a 4-dimensional momentum.

```
Momentum[p]
```

$$\overline{p}$$

As an optional second argument the dimension must be specified if it is different from 4.

```
Momentum[p, D]
```

$$p$$

The dimension index is suppressed in the output.

```
Momentum[p, d]
```

$$p$$

```
Momentum[-q]
```

$$-\overline{q}$$

```
Momentum[-q] // StandardForm

(*-Momentum[q]*)
```

```
ex = Momentum[p - q] + Momentum[2 q]
```

$$(\overline{p} - \overline{q}) + 2\overline{q}$$

```
ex // StandardForm

(*Momentum[p - q] + 2 Momentum[q]*)
```

```
ex // MomentumExpand // StandardForm

(*Momentum[p] + Momentum[q]*)
```

```
ex // MomentumCombine // StandardForm

(*Momentum[p + q]*)
```

```
ChangeDimension[Momentum[p], d] // StandardForm

(*Momentum[p, d]*)
```

2.111 MT

MT[mu, nu] is the metric tensor in 4 dimensions.

2.111.1 See also

Overview, FeynCalcExternal, FCE, FCI, MTD, MTE.

2.111.2 Examples

```
MT[\[Alpha], \[Beta]]
```

$$\bar{g}^{\alpha\beta}$$

```
Contract[MT[\[Alpha], \[Beta]] MT[\[Alpha], \[Beta]]]
```

$$4$$

```
MT[a, b] // StandardForm

(*MT[a, b]*)
```

```
FCI[MT[a, b]] // StandardForm

(*Pair[LorentzIndex[a], LorentzIndex[b]]*)
```

```
FCE[FCI[MT[a, b]]] // StandardForm

(*MT[a, b]*)
```

2.112 MTD

MTD[mu, nu] is the metric tensor in D dimensions.

2.112.1 See also

Overview, FeynCalcExternal, FCE, FCI, MT, MTE.

2.112.2 Examples

```
MTD[\[Alpha], \[Beta]]
```

$$g^{\alpha\beta}$$

```
Contract[MTD[\[Alpha], \[Beta]] MTD[\[Alpha], \[Beta]]]
```

$$D$$

```
MTD[\[Alpha], \[Beta]] // StandardForm

(*MTD[\[Alpha], \[Beta]]*)
```

```
FCI[MTD[\[Alpha], \[Beta]]] // StandardForm

(*Pair[LorentzIndex[\[Alpha], D], LorentzIndex[\[Beta], D]]*)
```

```
FCE[FCI[MTD[\[Mu], \[Nu]]]] // StandardForm

(*MTD[\[Mu], \[Nu]]*)
```

2.113 MTE

`MTE[mu, nu]` is the metric tensor in $D-4$ dimensions.

2.113.1 See also

Overview, FeynCalcExternal, FCE, FCI, MT, MTD.

2.113.2 Examples

```
MTE[\[Alpha], \[Beta]]
```

$$\hat{g}^{\alpha\beta}$$

```
Contract[MTE[\[Alpha], \[Beta]] MTE[\[Alpha], \[Beta]]]
```

$$D-4$$

```
Contract[MTE[\[Alpha], \[Beta]] MT[\[Alpha], \[Beta]]]
```

$$0$$

```
Contract[MTE[\[Alpha], \[Beta]] MTD[\[Alpha], \[Beta]]]
```

$$D-4$$

```
MTE[\[Alpha], \[Beta]] // StandardForm

(*MTE[\[Alpha], \[Beta]]*)
```

```
MTE[\[Alpha], \[Beta]]
```

$$\hat{g}^{\alpha\beta}$$

```
FCI[MTE[\[Alpha], \[Beta]]] // StandardForm

(*Pair[LorentzIndex[\[Alpha], -4 + D], LorentzIndex[\[Beta], -4 + D]]*)
```

```
FCE[FCI[MTE[\[Mu], \[Nu]]]] // StandardForm

(*MTE[\[Mu], \[Nu]]*)
```

```
MTE[\[Mu], \[Nu]]
```

$$\hat{g}^{\mu\nu}$$

2.114 Nf

Nf denotes the number of flavors.

2.114.1 See also

Overview, Amplitude, CounterTerm, Convolute, AnomalousDimension, SplittingFunction.

2.114.2 Examples

2.115 Pair

Pair[x, y] is the head of a special pairing used in the internal representation: **x** and **y** may have heads **LorentzIndex** or **Momentum**.

If both **x** and **y** have head **LorentzIndex**, the metric tensor (e.g. $g^{\mu\nu}$) is understood.

If **x** and **y** have head **Momentum**, a scalar product (e.g. $p \cdot q$) is meant.

If one of **x** and **y** has head **LorentzIndex** and the other **Momentum**, a Lorentz vector (e.g. p^{μ}) is implied.

2.115.1 See also

Overview, FV, FVD, MT, MTD, ScalarProduct, SP, SPD.

2.115.2 Examples

This represents a 4-dimensional metric tensor

```
Pair[LorentzIndex[\[Alpha]], LorentzIndex[\[Beta]]]
```

$$\bar{g}^{\alpha\beta}$$

This is a D-dimensional metric tensor

```
Pair[LorentzIndex[\[Alpha], D], LorentzIndex[\[Beta], D]]
```

$$g^{\alpha\beta}$$

If the Lorentz indices live in different dimensions, this gets resolved according to the t'Hooft-Veltman-Breitenlohner-Maison prescription

```
Pair[LorentzIndex[\[Alpha], n - 4], LorentzIndex[\[Beta]]]
```

$$0$$

A 4-dimensional Lorentz vector

```
Pair[LorentzIndex[\[Alpha]], Momentum[p]]
```

$$\overline{p}^{\alpha}$$

A D-dimensional Lorentz vector

```
Pair[LorentzIndex[\[Alpha], D], Momentum[p, D]]
```

$$p^{\alpha}$$

4-dimensional scalar products of Lorentz vectors

```
Pair[Momentum[q], Momentum[p]]
```

$$\overline{p} \cdot \overline{q}$$

```
Pair[Momentum[p], Momentum[p]]
```

$$\overline{p}^2$$

```
Pair[Momentum[p - q], Momentum[p]]
```

$$\overline{p} \cdot (\overline{p} - \overline{q})$$

```
Pair[Momentum[p], Momentum[p]]^2
```

$$\overline{p}^4$$

```
Pair[Momentum[p], Momentum[p]]^3
```

$$\overline{p}^6$$

```
ExpandScalarProduct[Pair[Momentum[p - q], Momentum[p]]]
```

$$\overline{p}^2 - \overline{p} \cdot \overline{q}$$

```
Pair[Momentum[-q], Momentum[p]] + Pair[Momentum[q], Momentum[p]]
```

$$0$$

2.116 PCHN

PCHN[x, i, j] is a chain of Pauli matrices **x** and is transformed into **PauliChain[FCI[x], PauliIndex[i], PauliIndex[j]]** by **FeynCalcInternal**.

2.116.1 See also

Overview, PauliChain, PauliIndex, PauliIndexDelta, PauliChainJoin, PauliChainExpand, PauliChainFactor.

2.116.2 Examples

A standalone Pauli matrix with open Pauli indices

```
PCHN[CSID[a], i, j]
```

$$(\sigma^a)_{ij}$$

A chain of Pauli matrices with open Pauli indices

```
PCHN[CSID[a] . CSID[b], i, j]
```

$$\left(\sigma^a.\sigma^b\right)_{ij}$$

A single $\xi^\dagger$ spinor with an open Pauli index

```
PCHN[PauliXi[-I], i]
```

$$\left(\xi^\dagger\right)_i$$

A single $\eta^\dagger$ spinor with an open Pauli index

```
PCHN[PauliEta[-I], i]
```

$$\left(\eta^\dagger\right)_i$$

A single ξ spinor with an open Pauli index

```
PCHN[i, PauliXi[I]]
```

$$(\xi)_i$$

A single η spinor with an open Pauli index

```
PCHN[i, PauliEta[I]]
```

$$(\eta)_i$$

$\xi^\dagger$ spinor contracted with a chain of Pauli matrices

```
PCHN[CSID[a] . CSID[b], PauliXi[-I], j]
```

$$\left(\xi^{\dagger}.\sigma^{a}.\sigma^{b}\right)_{j}$$

$\eta^{\dagger}$ spinor contracted with a chain of Pauli matrices

```
PCHN[CSID[a] . CSID[b], PauliEta[-I], j]
```

$$\left(\eta^{\dagger}.\sigma^{a}.\sigma^{b}\right)_{j}$$

ξ spinor contracted with a chain of Pauli matrices

```
PCHN[CSID[a] . CSID[b], i, PauliXi[I]]
```

$$\left(\sigma^{a}.\sigma^{b}.\xi\right)_{i}$$

η spinor contracted with a chain of Pauli matrices

```
PCHN[CSID[a] . CSID[b], i, PauliEta[I]]
```

$$\left(\sigma^{a}.\sigma^{b}.\eta\right)_{i}$$

2.117 PauliChain

`PauliChain[x, i, j]` denotes a chain of Pauli matrices **`x`**, where the Pauli indices **`i`** and **`j`** are explicit.

2.117.1 See also

Overview, PCHN, PauliIndex, PauliIndexDelta, PauliChainJoin, PauliChainExpand, PauliChainFactor.

2.117.2 Examples

A standalone Pauli matrix σ^{i}_{jk}

```
PauliChain[PauliSigma[CartesianIndex[a]], PauliIndex[i], PauliIndex[j]]
```

$$(\overline{\sigma}^{a})_{ij}$$

A chain of Pauli matrices with open indices

```
PauliChain[PauliSigma[CartesianIndex[a, D - 1], D - 1] .
PauliSigma[CartesianIndex[b, D - 1], D - 1], PauliIndex[i], PauliIndex[j]]
```

$$\left(\sigma^{a}.\sigma^{b}\right)_{ij}$$

A **PauliChain** with only two arguments denotes a spinor component

```
PauliChain[PauliXi[-I], PauliIndex[i]]
```

$$\left(\xi^{\dagger}\right)_{i}$$

```
PauliChain[PauliEta[-I], PauliIndex[i]]
```

$$\left(\eta^{\dagger}\right)_{i}$$

```
PauliChain[PauliIndex[i], PauliXi[I]]
```

$$(\xi)_{i}$$

```
PauliChain[PauliIndex[i], PauliEta[I]]
```

$$(\eta)_{i}$$

The chain may also be partially open or closed

```
PauliChain[PauliSigma[CartesianIndex[a]] . (m +
PauliSigma[CartesianMomentum[p]]) . PauliSigma[CartesianIndex[b]],
PauliXi[-I], PauliIndex[j]]
```

$$\left(\xi^{\dagger}.\overline{\sigma}^{a}.\left(\overline{\sigma}\cdot\overline{p}+m\right).\overline{\sigma}^{b}\right)_{j}$$

```
PauliChain[PauliSigma[CartesianIndex[a]] . (m +
PauliSigma[CartesianMomentum[p]]) . PauliSigma[CartesianIndex[b]],
PauliIndex[i], PauliXi[I]]
```

$$\left(\overline{\sigma}^a.\left(\overline{\sigma}\cdot\overline{p}+m\right).\overline{\sigma}^b.\xi\right)_i$$

```
PauliChain[PauliSigma[CartesianIndex[a]] . (m +
PauliSigma[CartesianMomentum[p]]) . PauliSigma[CartesianIndex[b]],
PauliXi[-I], PauliEta[I]]
```

$$\left(\xi^\dagger.\overline{\sigma}^a.\left(\overline{\sigma}\cdot\overline{p}+m\right).\overline{\sigma}^b.\eta\right)$$

```
PauliChain[1, PauliXi[-I], PauliEta[I]]
```

$$\left(\xi^\dagger.\eta\right)$$

2.118 PauliEta

`PauliEta[I]` represents a two-component Pauli spinor η, while **`PauliEta[-I]`** stands for $\eta^\dagger$.

2.118.1 See also

Overview, PauliXi.

2.118.2 Examples

```
PauliEta[I]
```

$$\eta$$

```
PauliEta[-I]
```

$$\eta^\dagger$$

```
PauliEta[-I] . SIS[p] . PauliXi[I]

% // ComplexConjugate
```

$$\eta^{\dagger}.(\bar{\sigma}\cdot\overline{p}).\xi$$

$$\xi^{\dagger}.(\bar{\sigma}\cdot\overline{p}).\eta$$

2.119 PauliXi

PauliXi[I] represents a two-component Pauli spinor ξ, while **PauliXi[-I]** stands for $\xi^{\dagger}$.

2.119.1 See also

Overview, PauliEta.

2.119.2 Examples

```
PauliXi[I]
```

$$\xi$$

```
PauliXi[-I]
```

$$\xi^{\dagger}$$

```
PauliXi[-I] . SIS[p] . PauliEta[I]

% // ComplexConjugate
```

$$\xi^{\dagger}.(\bar{\sigma}\cdot\overline{p}).\eta$$

$$\eta^{\dagger}.(\bar{\sigma}\cdot\overline{p}).\xi$$

2.120 PauliIndexDelta

`PauliIndexDelta[PauliIndex[i], PauliIndex[j]]` is the Kronecker-delta in the Pauli space with two explicit Pauli indices **`i`** and **`j`**.

2.120.1 See also

Overview, PauliChain, PCHN, PauliIndex, DIDelta, PauliChainJoin, PauliChainCombine, PauliChainExpand, PauliChainFactor.

2.120.2 Examples

```
PauliIndexDelta[PauliIndex[i], PauliIndex[j]]
```

$$\delta_{ij}$$

```
PauliIndexDelta[PauliIndex[i], PauliIndex[j]]^2

PauliChainJoin[%]

PauliChainJoin[%%, TraceOfOne -> D]
```

$$\delta_{ij}^2$$

$$4$$

$$D$$

```
PauliIndexDelta[PauliIndex[i], PauliIndex[j]]
PauliIndexDelta[PauliIndex[j], PauliIndex[k]]

PauliChainJoin[%]
```

$$\delta_{ij}\delta_{jk}$$

$$\delta_{ik}$$

```
PauliChain[PauliEta[-I], PauliIndex[i0]] PIDelta[i0, i1] // FCI //
PauliChainJoin
```

$$\left(\eta^{\dagger}\right)_{\text{i1}}$$

```
PauliIndexDelta[PauliIndex[i2], PauliIndex[i3]]
PauliIndexDelta[PauliIndex[i4], PauliIndex[i5]] PauliChain[PauliIndex[i7],
PauliXi[I]] PauliChain[PauliEta[-I], PauliIndex[i0]]
PauliChain[PauliSigma[CartesianIndex[a]], PauliIndex[i1], PauliIndex[i2]]
PauliChain[PauliSigma[CartesianIndex[b]], PauliIndex[i5], PauliIndex[i6]]
PauliChain[m + PauliSigma[CartesianMomentum[p]], PauliIndex[i3],
PauliIndex[i4]]

PauliChainJoin[%]
```

$$(\xi)_{\text{i7}} \left(\eta^{\dagger}\right)_{\text{i0}} \delta_{\text{i2 i3}} \delta_{\text{i4 i5}} \left(\overline{\sigma}^{a}\right)_{\text{i1 i2}} \left(\overline{\sigma}^{b}\right)_{\text{i5 i6}} (\overline{\sigma} \cdot \overline{p} + m)_{\text{i3 i4}}$$

$$(\xi)_{\text{i7}} \left(\eta^{\dagger}\right)_{\text{i0}} \left(\overline{\sigma}^{a} . (\overline{\sigma} \cdot \overline{p} + m) . \overline{\sigma}^{b}\right)_{\text{i1 i6}}$$

```
PauliChainJoin[% PIDelta[i0, i1]]
```

$$(\xi)_{\text{i7}} \left(\eta^{\dagger} . \overline{\sigma}^{a} . (\overline{\sigma} \cdot \overline{p} + m) . \overline{\sigma}^{b}\right)_{\text{i6}}$$

```
PauliChainJoin[% PIDelta[i7, i6]]
```

$$\eta^{\dagger} . \overline{\sigma}^{a} . (\overline{\sigma} \cdot \overline{p} + m) . \overline{\sigma}^{b} . \xi$$

2.121 PIDelta

`PIDelta[i, j]` is the Kronecker-delta in the Pauli space. **`PIDelta[i, j]`** is transformed into **`PauliIndexDelta[PauliIndex[i], PauliIndex[j]]`** by FeynCalcInternal.

2.121.1 See also

Overview, PauliChain, PCHN, PauliIndex, PauliIndexDelta, PauliChainJoin, PauliChainExpand, PauliChainFactor.

2.121.2 Examples

```
PIDelta[i, j]
```

$$\delta_{ij}$$

```
PIDelta[i, i]

PauliChainJoin[%]
```

$$\delta_{ii}$$

$$4$$

```
PIDelta[i, j]^2

PauliChainJoin[%]
```

$$\delta_{ij}^2$$

$$4$$

```
PIDelta[i, j] PIDelta[j, k]

PauliChainJoin[%]
```

$$\delta_{ij}\delta_{jk}$$

$$\delta_{ik}$$

```
ex = PIDelta[i2, i3] PIDelta[i4, i5]  PCHN[i7, PauliXi[I]]
PauliChain[PauliEta[-I], PauliIndex[i0]]
PauliChain[PauliSigma[CartesianIndex[a]], PauliIndex[i1], PauliIndex[i2]]
PauliChain[PauliSigma[CartesianIndex[b]], PauliIndex[i5], PauliIndex[i6]]
PauliChain[m + PauliSigma[CartesianMomentum[p]], PauliIndex[i3],
PauliIndex[i4]]
```

$$(\xi)_{\text{i7}} \left(\eta^{\dagger}\right)_{\text{i0}} \delta_{\text{i2 i3}} \delta_{\text{i4 i5}} \left(\overline{\sigma}^{a}\right)_{\text{i1 i2}} \left(\overline{\sigma}^{b}\right)_{\text{i5 i6}} \left(\overline{\sigma} \cdot \overline{p} + m\right)_{\text{i3 i4}}$$

```
PauliChainJoin[ex]
```

$$(\xi)_{i7} \left(\eta^{\dagger}\right)_{i0} \left(\overline{\sigma}^{a}.\left(\overline{\sigma}\cdot\overline{p}+m\right).\overline{\sigma}^{b}\right)_{i1\ i6}$$

```
PauliChainJoin[ex PIDelta[i0, i1]]
```

$$(\xi)_{i7} \left(\eta^{\dagger}.\overline{\sigma}^{a}.\left(\overline{\sigma}\cdot\overline{p}+m\right).\overline{\sigma}^{b}\right)_{i6}$$

```
PauliChainJoin[% PIDelta[i7, i6]]
```

$$\eta^{\dagger}.\overline{\sigma}^{a}.\left(\overline{\sigma}\cdot\overline{p}+m\right).\overline{\sigma}^{b}.\xi$$

2.122 PauliSigma

PauliSigma[x, dim] is the internal representation of a Pauli matrix with a Lorentz or Cartesian index or a contraction of a Pauli matrix and a Lorentz or Cartesian vector.

PauliSigma[x, 3] simplifies to **PauliSigma[x]**.

2.122.1 See also

Overview, SI, CSI.

2.122.2 Examples

```
PauliSigma[LorentzIndex[\[Alpha]]]
```

$$\bar{\sigma}^{\alpha}$$

```
PauliSigma[CartesianIndex[i]]
```

$$\overline{\sigma}^{i}$$

A Pauli matrix contracted with a Lorentz or Cartesian vector is displayed as $\sigma \cdot p$

```
PauliSigma[Momentum[p]]
```

$$\bar{\sigma} \cdot \overline{p}$$

```
PauliSigma[CartesianMomentum[p]]
```

$$\overline{\sigma} \cdot \overline{p}$$

```
PauliSigma[Momentum[q]] . PauliSigma[Momentum[p - q]]
% // PauliSigmaExpand
```

$$(\bar{\sigma} \cdot \overline{q}) . (\bar{\sigma} \cdot (\overline{p} - \overline{q}))$$

$$(\bar{\sigma} \cdot \overline{q}) . (\bar{\sigma} \cdot \overline{p} - \bar{\sigma} \cdot \overline{q})$$

```
PauliSigma[CartesianMomentum[q]] . PauliSigma[CartesianMomentum[p - q]]
% // PauliSigmaExpand
```

$$(\overline{\sigma} \cdot \overline{q}) . (\overline{\sigma} \cdot (\overline{p} - \overline{q}))$$

$$(\overline{\sigma} \cdot \overline{q}) . (\overline{\sigma} \cdot \overline{p} - \overline{\sigma} \cdot \overline{q})$$

2.123 Polarization

Polarization[k] is the head of a polarization momentum with momentum **k**.

A slashed polarization vector ($\varepsilon_\mu(k)\gamma^\mu$) has to be entered as **GS[Polarization[k]]**.

Unless the option **Transversality** is set to **True**, all polarization vectors are not transverse by default.

The internal representation for a polarization vector corresponding to a boson with four momentum k is: **Momentum[Polarization[k, I]]**.

Polarization[k, -I] denotes the complex conjugate polarization.

Polarization is also an option of various functions related to the operator product expansion. The setting **0** denotes the unpolarized and **1** the polarized case.

Polarization may appear only inside **Momentum**. Outside of **Momentum** it is meaningless in FeynCalc.

The imaginary unit in the second argument of **Polarization** is used to distinguish between incoming and outgoing polarization vectors.

- **Pair[Momentum[k], Momentum[Polarization[k, I]]]** corresponds to $\varepsilon^\mu(k)$, i.e. an ingoing polarization vector
- **Pair[Momentum[k], Momentum[Polarization[k, -I]]]** corresponds to $\varepsilon^{*\mu}(k)$, i.e. an outgoing polarization vector

2.123.1 See also

Overview, PolarizationVector, PolarizationSum, DoPolarizationSums.

2.123.2 Examples

```
Polarization[k]
```

$$\text{Polarization}(k)$$

```
Polarization[k] // ComplexConjugate
```

$$\text{Polarization}(k)$$

```
GS[Polarization[k]]
```

$$\bar{\gamma} \cdot \overline{\text{Polarization}(k)}$$

```
GS[Polarization[k]] // StandardForm

(*GS[Polarization[k]]*)
```

```
Pair[Momentum[k], Momentum[Polarization[k, I]]]
```

$$\overline{k} \cdot \bar{\varepsilon}(k)$$

2.124 PolarizationVector

PolarizationVector[p, mu] denotes a 4-dimensional ingoing polarization vector $\varepsilon^{\mu}(p)$.

For the outgoing polarization vector $\varepsilon^{*\mu}(p)$ use **ComplexConjugate[PolarizationVector[p, mu]]**

To obtain a D-dimensional polarization vector, just use **ChangeDimension[vec, D]**

In the internal representation following conventions are used

- **Pair[Momentum[k], Momentum[Polarization[k, I]]]** corresponds to $\varepsilon^{\mu}(k)$, i.e. an ingoing polarization vector
- **Pair[Momentum[k], Momentum[Polarization[k, -I]]]** corresponds to $\varepsilon^{*\mu}(k)$, i.e. an outgoing polarization vector

Warning: The first argument of **PolarizationVector** should always be a standalone symbol denoting a momentum (e.g. **p**, **k1**, **q2** etc.). Never use symbols multiplied by -1 or other numbers as well as products of symbols (e.g. **-p**, **2*k**, **x*p** etc.). Doing so will inevitably lead to wrong results.

2.124.1 See also

Overview, FV, Pair, Polarization, PolarizationSum, DoPolarizationSums.

2.124.2 Examples

A polarization vector $\varepsilon^{\mu}(k)$ is a special 4-vector.

```
PolarizationVector[k, \[Mu]]
```

$$\bar{\varepsilon}^{\mu}(k)$$

```
PolarizationVector[k, \[Mu]] // StandardForm

(*Pair[LorentzIndex[\[Mu]], Momentum[Polarization[k, I]]]*)
```

```
Conjugate[PolarizationVector[k, \[Mu]]]
```

$$\bar{\varepsilon}^{*\mu}(k)$$

```
Conjugate[PolarizationVector[k, \[Mu]]] // StandardForm

(*Pair[LorentzIndex[\[Mu]], Momentum[Polarization[k, -I]]]*)
```

The transversality property is not automatic and must be explicitly activated using the option **Transversality**

```
PolarizationVector[k, \[Mu]] FV[k, \[Mu]]

Contract[%]
```

$$\overline{k}^{\mu}\bar{\varepsilon}^{\mu}(k)$$

$$\overline{k}\cdot\bar{\varepsilon}(k)$$

```
PolarizationVector[k, \[Mu], Transversality -> True] FV[k, \[Mu]]

Contract[%]
```

$$\overline{k}^{\mu}\bar{\varepsilon}^{\mu}(k)$$

$$0$$

Suppose that you are using unphysical polarization vectors for massless gauge bosons and intend to remove the unphysical degrees of freedom at a later stage using ghosts. In this case you must not use **Transversality->True**, since your polarization vectors are not transverse. Otherwise the result will be inconsistent.

Here everything is correct, we can use the gauge trick with unphysical polarization vectors.

```
FCClearScalarProducts[];

SP[k1] = 0;

SP[k2] = 0;
```

```
ex1 = SP[k1, Polarization[k1, I]] SP[k2, Polarization[k1, -I]] SP[k1,
Polarization[k2, I]]*
   SP[k2, Polarization[k2, -I]]
```

$$\left(\overline{\text{k1}}\cdot\bar{\varepsilon}(\text{k1})\right)\left(\overline{\text{k2}}\cdot\bar{\varepsilon}^*(\text{k2})\right)\left(\overline{\text{k1}}\cdot\bar{\varepsilon}(\text{k2})\right)\left(\overline{\text{k2}}\cdot\bar{\varepsilon}^*(\text{k1})\right)$$

```
ex1 // DoPolarizationSums[#, k1, 0] & // DoPolarizationSums[#, k2, 0] &
```

$$(\overline{\text{k1}}\cdot\overline{\text{k2}})^2$$

Here we erroneously set **`Transversality->True`** and consequently obtain a wrong result. In pure QED the full result (physical amplitude squared) would still come out right owing to the Ward identities, but e.g. in QCD this would not be the case.

```
ex2 = SP[k1, Polarization[k1, I, Transversality -> True]] SP[k2,
Polarization[k1, -I,
      Transversality -> True]] SP[k1, Polarization[k2, I, Transversality ->
True]] SP[k2,
     Polarization[k2, -I, Transversality -> True]] // FCI
```

$$0$$

```
ex2 // DoPolarizationSums[#, k1, 0] & // DoPolarizationSums[#, k2, 0] &
```

$$0$$

```
FCClearScalarProducts[];
```

`PolarizationVector` is a shortcut for 4-dimensional polarization vectors. Although D-dimensional polarization vectors are fully supported by FeynCalc, as of now there is no shortcut for entering such quantities. You can either use **`ChangeDimension`**

```
ChangeDimension[PolarizationVector[q, \[Mu]], D]
```

$$\varepsilon^{\mu}(q)$$

or enter such quantities directly using the **FeynCalcInternal**-notation

```
Pair[Momentum[Polarization[q, I], D], LorentzIndex[\[Mu], D]]
```

$$\varepsilon^{\mu}(q)$$

2.125 PlusDistribution

PlusDistribution[1/(1 - x)] denotes a distribution (in the sense of the "+" prescription).

2.125.1 See also

Overview, Integrate2.

2.125.2 Examples

```
PlusDistribution[1/(1 - x)]
```

$$\left(\frac{1}{1-x}\right)_+$$

```
PlusDistribution[Log[1 - x]/(1 - x)]
```

$$\left(\frac{\log(1-x)}{1-x}\right)_+$$

```
Integrate2[PlusDistribution[1/(1 - x)], {x, 0, 1}]
```

$$0$$

```
Integrate2[PlusDistribution[Log[1 - x]/(1 - x)], {x, 0, 1}]
```

$$0$$

```
Integrate2[PlusDistribution[Log[1 - x]^2/(1 - x)], {x, 0, 1}]
```

$$0$$

```
PlusDistribution[Log[x (1 - x)]/(1 - x)]
```

$$\frac{\log(x)}{1-x} + \left(\frac{\log(1-x)}{1-x}\right)_+$$

2.126 PropagatorDenominator

PropagatorDenominator[Momentum[q], m] is a factor of the denominator of a propagator. If **q** is supposed to be D-dimensional, use **PropagatorDenominator[Momentum[q, D], m]**. What is meant is $1/(q^2 - m^2)$.

PropagatorDenominator must appear inside **FeynAmpDenominator**, it is not a standalone object.

2.126.1 See also

Overview, FeynAmpDenominator, FeynAmpDenominatorExplicit.

2.126.2 Examples

```
FeynAmpDenominator[PropagatorDenominator[Momentum[p], m]]
```

$$\frac{1}{\overline{p}^2 - m^2}$$

```
FeynAmpDenominator[PropagatorDenominator[Momentum[p, D], m]]
```

$$\frac{1}{p^2 - m^2}$$

2.127 PD

PD is an abbreviation for **PropagatorDenominator**.

2.127.1 See also

Overview, PropagatorDenominator, IFPD.

2.127.2 Examples

2.128 StandardPropagatorDenominator

StandardPropagatorDenominator[propSq + ..., propEik +..., m^2, {n, s}] encodes a generic Lorentzian propagator denominator $\frac{1}{[(q_1+...)^2+q_1\cdot p_1+...+m^2+si\eta]^n}$.

propSq should be of the form **Momentum[q1, D]**, while **propEik** should look like **Pair[Momentum[q1, D], Momentum[p1, D]**.

This allows to accommodate for standard propagators of the type $1/(p^2 - m^2)$ but also for propagators encountered in manifestly Lorentz covariant effective field theories such as HQET or SCET.

StandardPropagatorDenominator is an internal object. To enter such propagators in FeynCalc you should use **SFAD**.

2.128.1 See also

Overview, PropagatorDenominator, CartesianPropagatorDenominator, GenericPropagatorDenominator, FeynAmpDenominator.

2.128.2 Examples

```
FeynAmpDenominator[StandardPropagatorDenominator[Momentum[p, D], 0, -m^2,
{1, 1}]]
```

$$\frac{1}{(p^2 - m^2 + i\eta)}$$

```
FeynAmpDenominator[StandardPropagatorDenominator[0, Pair[Momentum[p, D],
Momentum[q, D]], -m^2, {1, 1}]]
```

$$\frac{1}{(p \cdot q - m^2 + i\eta)}$$

2.129 CartesianPropagatorDenominator

CartesianPropagatorDenominator[propSq + ..., propEik + ..., m^2, {n, s}] encodes a generic Cartesian propagator denominator of the form $\frac{1}{[(q1+...)^2+q1.p1+...+m^2+s*I\,\eta]^n}$

propSq should be of the form **CartesianMomentum[q1, D - 1]**, while **propEik** should look like **CartesianPair[CartesianMomentum[q1, D - 1], CartesianMomentum[p1, D - 1]**.

CartesianPropagatorDenominator is an internal object. To enter such propagators in FeynCalc you should use **CFAD**.

2.129.1 See also

Overview, CFAD, FeynAmpDenominator.

2.129.2 Examples

Standard 3-dimensional Cartesian propagator

```
FeynAmpDenominator[CartesianPropagatorDenominator[CartesianMomentum[p, D -
1], 0, m^2, {1, -1}]]
```

$$\frac{1}{(p^2+m^2-i\eta)}$$

Here we switch the sign of the mass term

```
FeynAmpDenominator[CartesianPropagatorDenominator[CartesianMomentum[p, D -
1], 0, -m^2, {1, -1}]]
```

$$\frac{1}{(p^2-m^2-i\eta)}$$

And here also the sign of $i\eta$

```
FeynAmpDenominator[CartesianPropagatorDenominator[CartesianMomentum[p, D -
1], 0, -m^2, {1, +1}]]
```

$$\frac{1}{(p^2-m^2+i\eta)}$$

Eikonal Cartesian propagator with a residual mass term

```
FeynAmpDenominator[CartesianPropagatorDenominator[0,
   CartesianPair[CartesianMomentum[p, D - 1], CartesianMomentum[q, D - 1]],
m^2, {1, -1}]]
```

$$\frac{1}{(p \cdot q + m^2 - i\eta)}$$

2.130 GenericPropagatorDenominator

GenericPropagatorDenominator[expr, {n, s}] is a generic factor of the denominator of a propagator. Unlike **PropagatorDenominator** that is supposed to mean $1/(q^2 - m^2)$, **expr** in **GenericPropagatorDenominator** can be an arbitrary combination of **Pair**, **CartesianPair** and **TemporalPair** objects.

2.130.1 See also

Overview, PropagatorDenominator, StandardPropagatorDenominator, CartesianPropagatorDenominator.

Using **n** one can specify the power of the propagator, while **s** (**+1** or **-1**) fixes the sign of **I*eta**. **GenericPropagatorDenominator** is an internal object. To enter such propagators in FeynCalc you should use **GFAD**.

2.130.2 Examples

```
FeynAmpDenominator[GenericPropagatorDenominator[x, {1, 1}]]
```

$$\frac{1}{(x + i\eta)}$$

```
FeynAmpDenominator[GenericPropagatorDenominator[2 z Pair[Momentum[p1, D],
      Momentum[Q, D]] Pair[Momentum[p2, D], Momentum[Q, D]] -
Pair[Momentum[p1, D],
     Momentum[p2, D]], {1, 1}]]
```

$$\frac{1}{(2z(\mathrm{p1} \cdot Q)(\mathrm{p2} \cdot Q) - \mathrm{p1} \cdot \mathrm{p2} + i\eta)}$$

2.131 QuantumField

QuantumField is the head of quantized fields and their derivatives.

QuantumField[par, ftype, {lorind}, {sunind}] denotes a quantum field of type **ftype** with (possible) Lorentz-indices **lorind** and $SU(N)$ indices **sunind**. The optional first argument **par** denotes a partial derivative acting on the field.

2.131.1 See also

Overview, FeynRule, FCPartialD, ExpandPartialD.

2.131.2 Examples

This denotes a scalar field.

```
QuantumField[S]
```

$$S$$

Quark fields

```
QuantumField[AntiQuarkField]
```

$$\bar{\psi}$$

```
QuantumField[QuarkField]
```

$$\psi$$

This is a field with a Lorentz index.

```
QuantumField[B, {\[Mu]}]
```

$$B_\mu$$

Color indices should be put after the Lorentz ones.

```
QuantumField[GaugeField, {\[Mu]}, {a}] // StandardForm

(*QuantumField[GaugeField, LorentzIndex[\[Mu]], SUNIndex[a]]*)
```

A^a_Δ is a short form for $\Delta^{mu} A^a_{mu}$

```
QuantumField[A, {OPEDelta}, {a}]
```

$$A^a_\Delta$$

The first list of indices is usually interpreted as type **`LorentzIndex`**, except for **`OPEDelta`**, which gets converted to type **`Momentum`**.

```
QuantumField[A, {OPEDelta}, {a}] // StandardForm

(*QuantumField[A, Momentum[OPEDelta], SUNIndex[a]]*)
```

Derivatives of fields are denoted as follows.

```
QuantumField[FCPartialD[LorentzIndex[\[Mu]]], A, {\[Mu]}]
```

$$(\partial_\mu A_\mu)$$

```
QuantumField[FCPartialD[OPEDelta], S]
```

$$(\partial_\Delta S)$$

```
QuantumField[FCPartialD[OPEDelta], A, {OPEDelta}, {a}]
```

$$(\partial_\Delta A^a_\Delta)$$

```
QuantumField[FCPartialD[OPEDelta]^OPEm, A, {OPEDelta}, {a}]
```

$$\partial^{m A \Delta a}_\Delta$$

```
QuantumField[QuantumField[A]] === QuantumField[A]
```

True

2.132 ScaleMu

ScaleMu is the mass scale used for dimensional regularization of loop integrals.

2.132.1 See also

Overview, Epsilon, EpsilonUV, EpsilonIR, SimplifyPolyLog.

2.132.2 Examples

```
ScaleMu
```

$$\mu$$

2.133 SD

SD[i, j] denotes the $SU(N)$ Kronecker delta with color indices **i** and **j** in the adjoint representation. **SD[i, j]** is transformed into **SUNDelta[SUNIndex[i], SUNIndex[j]]** by **FeynCalcInternal**.

2.133.1 See also

Overview, SUNDelta.

2.133.2 Examples

```
SD[a, b]
```

$$\delta^{ab}$$

```
SD[a, b] // FCI // StandardForm

(*SUNDelta[SUNIndex[a], SUNIndex[b]]*)
```

```
SD[a, b] // FCE // StandardForm

(*SD[a, b]*)
```

2.134 SUNDelta

SUNDelta[a, b] is the Kronecker-delta for $SU(N)$ with color indices **a** and **b** in the adjoint representation.

2.134.1 See also

Overview, ExplicitSUNIndex, SD, SUNF, SUNIndex, SUNSimplify, Trick.

2.134.2 Examples

```
SUNDelta[SUNIndex[a], SUNIndex[b]]
```

$$\delta^{ab}$$

```
SUNDelta[SUNIndex[a], SUNIndex[b]] SUNDelta[SUNIndex[b], SUNIndex[c]]

SUNSimplify[%]
```

$$\delta^{ab}\delta^{bc}$$

$$\delta^{ac}$$

```
SUNDelta[SUNIndex[a], SUNIndex[b]] // StandardForm

(*SUNDelta[SUNIndex[a], SUNIndex[b]]*)
```

```
SUNDelta[SUNIndex[a], SUNIndex[b]] // FCI // FCE // StandardForm

(*SD[a, b]*)
```

```
SD[a, b] // FCI // StandardForm

(*SUNDelta[SUNIndex[a], SUNIndex[b]]*)
```

The arguments of **SUNDelta** may also represent explicit integer indices via the head **ExplictiSUNIndex**. The difference is that **SUNSimplify** will only sum over symbolic indices.

```
ex = SUNDelta[SUNIndex[a], ExplicitSUNIndex[2]] SUNDelta[SUNIndex[a],
SUNIndex[b]] SUNDelta[SUNIndex[c], ExplicitSUNIndex[2]] // SUNSimplify
```

$$\delta^{2b}\delta^{2c}$$

```
ex // StandardForm

(*SUNDelta[ExplicitSUNIndex[2], SUNIndex[b]] SUNDelta[ExplicitSUNIndex[2],
SUNIndex[c]]*)
```

```
SD[1, 2] // FCI // StandardForm

(*SUNDelta[ExplicitSUNIndex[1], ExplicitSUNIndex[2]]*)
```

2.135 SDF

SDF[i, j] denotes the $SU(N)$ Kronecker delta with color indices **i** and **j** in the fundamental representation. **SDF[i, j]** is transformed into **SUNFDelta[SUNFIndex[i], SUNFIndex[j]]** by **FeynCalcInternal**.

2.135.1 See also

Overview, SUNFDelta.

2.135.2 Examples

```
SDF[a, b]
```

$$\delta_{ab}$$

```
SDF[a, b] // FCI // StandardForm

(*SUNFDelta[SUNFIndex[a], SUNFIndex[b]]*)
```

```
SDF[a, b] // FCE // StandardForm

(*SDF[a, b]*)
```

2.136 SUNFDelta

SUNFDelta[a, b] is the Kronecker-delta for $SU(N)$ with color indices **a** and **b** in the fundamental representation.

2.136.1 See also

Overview, SUNDelta.

2.136.2 Examples

```
SUNFDelta[SUNFIndex[a], SUNFIndex[b]]
```

$$\delta_{ab}$$

```
SUNFDelta[SUNFIndex[a], SUNFIndex[b]] SUNFDelta[SUNFIndex[b], SUNFIndex[c]]
% // SUNSimplify
```

$$\delta_{ab}\delta_{bc}$$

$$\delta_{ac}$$

```
SUNFDelta[SUNFIndex[a], SUNFIndex[b]]^2
% // SUNSimplify
```

$$\delta_{ab}^2$$

$$C_A$$

```
SUNFDelta[SUNFIndex[a], SUNFIndex[b]] // StandardForm
(*SUNFDelta[SUNFIndex[a], SUNFIndex[b]]*)
```

```
SUNFDelta[SUNFIndex[a], SUNFIndex[b]] // FCI // FCE // StandardForm
(*SDF[a, b]*)
```

```
SDF[a, b] // FCI // StandardForm

(*SUNFDelta[SUNFIndex[a], SUNFIndex[b]]*)
```

The arguments of **SUNFDelta** may also represent explicit integer indices via the head **ExplictiSUNFIndex**. The difference is that **SUNSimplify** will only sum over symbolic indices.

```
ex = SUNFDelta[SUNFIndex[a], ExplicitSUNFIndex[2]] SUNFDelta[SUNFIndex[a],
SUNFIndex[b]] SUNFDelta[SUNFIndex[c], ExplicitSUNFIndex[2]] // SUNSimplify
```

$$\delta_{2b}\delta_{2c}$$

```
ex // StandardForm

(*SUNFDelta[ExplicitSUNFIndex[2], SUNFIndex[b]]
SUNFDelta[ExplicitSUNFIndex[2], SUNFIndex[c]]*)
```

```
SDF[1, 2] // FCI // StandardForm

(*SUNFDelta[ExplicitSUNFIndex[1], ExplicitSUNFIndex[2]]*)
```

2.137 SI

SI[mu] can be used as input for 3-dimensional σ^μ with 4-dimensional Lorentz index μ and is transformed into **PauliSigma[LorentzIndex[mu]]** by FeynCalcInternal.

2.137.1 See also

Overview, PauliSigma, SID, SIE.

2.137.2 Examples

```
SI[\[Mu]]
```

$$\bar{\sigma}^\mu$$

```
SI[\[Mu]] // FCI // StandardForm

(*PauliSigma[LorentzIndex[\[Mu]]]*)
```

```
SI[\[Mu], \[Nu]] - SI[\[Nu], \[Mu]]
```

$$\bar{\sigma}^{\mu}.\bar{\sigma}^{\nu} - \bar{\sigma}^{\nu}.\bar{\sigma}^{\mu}$$

```
SI[\[Mu], \[Nu], \[Rho], \[Sigma]]
```

$$\bar{\sigma}^{\mu}.\bar{\sigma}^{\nu}.\bar{\sigma}^{\rho}.\bar{\sigma}^{\sigma}$$

```
SI[\[Mu], \[Nu], \[Rho], \[Sigma]] // StandardForm

(*SI[\[Mu]] . SI[\[Nu]] . SI[\[Rho]] . SI[\[Sigma]]*)
```

```
SI[\[Alpha]] . (SIS[p] + m) . SI[\[Beta]]
```

$$\bar{\sigma}^{\alpha}.(\bar{\sigma}\cdot\overline{p} + m).\bar{\sigma}^{\beta}$$

2.138 SID

SID[mu] can be used as input for $D-1$-dimensional σ^{μ} with D-dimensional Lorentz index μ and is transformed into **PauliSigma[LorentzIndex[mu, D], D-1]** by **FeynCalcInternal**.

2.138.1 See also

Overview, PauliSigma, SI, SIE.

2.138.2 Examples

```
SID[\[Mu]]
```

$$\sigma^{\mu}$$

```
SID[\[Mu], \[Nu]] - SID[\[Nu], \[Mu]]
```

$$\sigma^{\mu}.\sigma^{\nu} - \sigma^{\nu}.\sigma^{\mu}$$

```
StandardForm[FCI[SID[\[Mu]]]]

(*PauliSigma[LorentzIndex[\[Mu], D], -1 + D]*)
```

```
SID[\[Mu], \[Nu], \[Rho], \[Sigma]]
```

$$\sigma^{\mu}.\sigma^{\nu}.\sigma^{\rho}.\sigma^{\sigma}$$

```
SID[\[Mu], \[Nu], \[Rho], \[Sigma]] // StandardForm

(*SID[\[Mu]] . SID[\[Nu]] . SID[\[Rho]] . SID[\[Sigma]]*)
```

```
SID[\[Alpha]] . (SISD[p] + m) . SID[\[Beta]]
```

$$\sigma^{\alpha}.(m + \sigma \cdot p).\sigma^{\beta}$$

2.139 SIE

SIE[mu] can be used as input for $D-1$-dimensional σ^{μ} with $D-4$-dimensional Lorentz index μ and is transformed into **PauliSigma[LorentzIndex[mu, D-4], D-4]** by FeynCalcInternal.

2.139.1 See also

Overview, PauliSigma, SI.

2.139.2 Examples

```
SIE[\[Mu]]
```

$$\hat{\sigma}^{\mu}$$

```
SIE[\[Mu], \[Nu]] - SIE[\[Nu], \[Mu]]
```

$$\hat{\sigma}^{\mu}.\hat{\sigma}^{\nu} - \hat{\sigma}^{\nu}.\hat{\sigma}^{\mu}$$

```
StandardForm[FCI[SIE[\[Mu]]]]

(*PauliSigma[LorentzIndex[\[Mu], -4 + D], -4 + D]*)
```

```
SIE[\[Mu], \[Nu], \[Rho], \[Sigma]]
```

$$\hat{\sigma}^{\mu}.\hat{\sigma}^{\nu}.\hat{\sigma}^{\rho}.\hat{\sigma}^{\sigma}$$

```
SIE[\[Mu], \[Nu], \[Rho], \[Sigma]] // StandardForm

(*SIE[\[Mu]] . SIE[\[Nu]] . SIE[\[Rho]] . SIE[\[Sigma]]*)
```

```
SIE[\[Alpha]] . (SISE[p] + m) . SIE[\[Beta]]
```

$$\hat{\sigma}^{\alpha}.(m+\hat{\sigma}\cdot\hat{p}).\hat{\sigma}^{\beta}$$

2.140 SIS

SIS[p] can be used as input for 3-dimensional $\sigma^{\mu} p_{\mu}$ with 4-dimensional Lorentz vector p and is transformed into **PauliSigma[Momentum[p]]** by FeynCalcInternal.

2.140.1 See also

Overview, PauliSigma, SISD.

2.140.2 Examples

```
SIS[p]
```

$$\bar{\sigma}\cdot\overline{p}$$

```
SIS[p] // FCI // StandardForm

(*PauliSigma[Momentum[p]]*)
```

```
SIS[p, q, r, s]
```

$$(\bar{\sigma}\cdot\bar{p}).(\bar{\sigma}\cdot\bar{q}).(\bar{\sigma}\cdot\bar{r}).(\bar{\sigma}\cdot\bar{s})$$

```
SIS[p, q, r, s] // StandardForm

(*SIS[p] . SIS[q] . SIS[r] . SIS[s]*)
```

```
SIS[q] . (SIS[p] + m) . SIS[q]
```

$$(\bar{\sigma}\cdot\bar{q}).(\bar{\sigma}\cdot\bar{p}+m).(\bar{\sigma}\cdot\bar{q})$$

2.141 SISD

SISD[p] can be used as input for $D-1$-dimensional $\sigma^{\mu}p_{\mu}$ with D-dimensional Lorentz vector p and is transformed into **PauliSigma[Momentum[p, D], D-1]** by **FeynCalcInternal**.

2.141.1 See also

Overview, PauliSigma, SIS.

2.141.2 Examples

```
SISD[p]
```

$$\sigma\cdot p$$

```
SISD[p] // FCI // StandardForm

(*PauliSigma[Momentum[p, D], -1 + D]*)
```

```
SISD[p, q, r, s]
```

$$(\sigma\cdot p).(\sigma\cdot q).(\sigma\cdot r).(\sigma\cdot s)$$

```
SISD[p, q, r, s] // StandardForm

(*SISD[p] . SISD[q] . SISD[r] . SISD[s]*)
```

```
SISD[q] . (SISD[p] + m) . SISD[q]
```

$$(\sigma \cdot q).(m + \sigma \cdot p).(\sigma \cdot q)$$

2.142 SISE

SISE[p] can be used as input for $D-4$-dimensional $\sigma^\mu p_\mu$ with $D-4$-dimensional Lorentz vector p and is transformed into **PauliSigma[Momentum[p, D-4], D-4]** by **FeynCalcInternal**.

2.142.1 See also

Overview, SIS, PauliSigma.

2.142.2 Examples

```
SISE[p]
```

$$\hat{\sigma} \cdot \hat{p}$$

```
SISE[p] // FCI // StandardForm

(*PauliSigma[Momentum[p, -4 + D], -4 + D]*)
```

```
SISE[p, q, r, s]
```

$$(\hat{\sigma} \cdot \hat{p}) \,.\, (\hat{\sigma} \cdot \hat{q}) \,.\, (\hat{\sigma} \cdot \hat{r}) \,.\, (\hat{\sigma} \cdot \hat{s})$$

```
SISE[p, q, r, s] // StandardForm

(*SISE[p] . SISE[q] . SISE[r] . SISE[s]*)
```

2.143 SmallDelta

SmallDelta denotes some small positive number.

2.143.1 See also

Overview, SmallEpsilon.

2.143.2 Examples

2.144 SmallEpsilon

SmallEpsilon denotes some small positive number.

2.144.1 See also

Overview, SmallDelta.

2.144.2 Examples

2.145 SmallVariable

SmallVariable[me] is the head of small (negligible) variables. This means any mass with this head can be neglected if it appears in a sum, but not as an argument of Passarino-Veltman (**PaVe**) functions or **PropagatorDenominator**.

2.145.1 See also

Overview, PaVe, PaVeReduce, PropagatorDenominator.

2.145.2 Examples

2.146 Spinor

Spinor[p, m, o] is the head of Dirac spinors. Which of the spinors u, v, $\bar{u}$ or $\bar{v}$ is understood, depends on the sign of the momentum argument **p** and the relative position of **Spinor** in the chain.

- **Spinor[Momentum[p], m]** means $\bar{u}$ if it stands at the beginning of the chain.
- **Spinor[Momentum[p], m]** means u if it stands at the end of the chain.
- **Spinor[-Momentum[p], m]** means $\bar{v}$ if it stands at the beginning of the chain.
- **Spinor[-Momentum[p], m]** means v if it stands at the end of the chain.

Spinors of fermions of mass m are normalized to have $\bar{u}u = 2m$ and $\bar{v}v = -2m$.

The optional argument **o** can be used for additional degrees of freedom. If no optional argument **o** is supplied, a **1** is substituted in.

2.146.1 See also

Overview, FermionSpinSum, DiracSimplify, SpinorU, SpinorV, SpinorUBar, SpinorVBar, SpinorUBarD, SpinorUD, SpinorVD, SpinorVBarD.

2.146.2 Examples

```
Spinor[Momentum[p]]
```

$$\varphi(\overline{p})$$

```
Spinor[Momentum[p], m]
```

$$\varphi(\overline{p}, m)$$

FeynCalc uses covariant normalization (as opposed to e.g. the normalization used in Bjorken & Drell).

```
Spinor[Momentum[p], m] . Spinor[Momentum[p], m] // DiracSimplify
```

$$2m$$

```
DiracSimplify[Spinor[-Momentum[p], m] . GS[p]]
```

$$-m\,(\varphi(-\overline{p}, m))$$

```
Spinor[Momentum[p]] // StandardForm

(*Spinor[Momentum[p], 0, 1]*)
```

```
ChangeDimension[Spinor[Momentum[p]], D] // StandardForm

(*Spinor[Momentum[p, D], 0, 1]*)
```

```
Spinor[Momentum[p], m] // StandardForm

(*Spinor[Momentum[p], m, 1]*)
```

SmallVariables are discarded by **Spinor**.

```
Spinor[Momentum[p], SmallVariable[m]] // StandardForm

(*Spinor[Momentum[p], 0, 1]*)
```

2.147 SpinorU

SpinorU[p, m] denotes a $u(p, m)$-spinor that depends on the 4-dimensional momentum p.

2.147.1 See also

Overview, Spinor, SpinorUBar, SpinorV, SpinorVBar, SpinorUBarD, SpinorUD, SpinorVD, SpinorVBarD.

2.147.2 Examples

```
SpinorU[p, m]
```

$$u(p, m)$$

```
SpinorU[p, m] // FCI // StandardForm

(*Spinor[Momentum[p], m, 1]*)
```

```
SpinorU[p]
```

$$u(p)$$

```
SpinorU[p] // FCI // StandardForm

(*Spinor[Momentum[p], 0, 1]*)
```

```
GS[p] . SpinorU[p]

DiracEquation[%]
```

$$(\bar{\gamma} \cdot \overline{p}) .u(p)$$

$$0$$

2.148 SpinorUBar

SpinorUBar[p, m] denotes a $\bar{u}(p, m)$-spinor that depends on the 4-dimensional momentum p.

2.148.1 See also

Overview, Spinor, SpinorU, SpinorV, SpinorVBar, SpinorUBarD, SpinorUD, SpinorVD, SpinorVBarD.

2.148.2 Examples

```
SpinorUBar[p, m]
```

$$\bar{u}(p, m)$$

```
SpinorUBar[p, m] // FCI // StandardForm

(*Spinor[Momentum[p], m, 1]*)
```

```
SpinorUBar[p]
```

$$\bar{u}(p)$$

```
SpinorUBar[p] // FCI // StandardForm

(*Spinor[Momentum[p], 0, 1]*)
```

```
SpinorUBar[p] . GS[p]

DiracEquation[%]
```

$$\bar{u}(p).(\bar{\gamma}\cdot\overline{p})$$

$$0$$

2.149 SpinorV

SpinorV[p, m] denotes a $v(p,m)$-spinor that depends on the 4-dimensional momentum p.

2.149.1 See also

Overview, Spinor, SpinorUBar, SpinorU, SpinorVBar, SpinorUBarD, SpinorUD, SpinorVD, SpinorVBarD.

2.149.2 Examples

```
SpinorV[p, m]
```

$$v(p,m)$$

```
SpinorV[p, m] // FCI // StandardForm

(*Spinor[-Momentum[p], m, 1]*)
```

```
SpinorV[p]
```

$$v(p)$$

```
SpinorV[p] // FCI // StandardForm

(*Spinor[-Momentum[p], 0, 1]*)
```

```
GS[p] . SpinorV[p]

DiracEquation[%]
```

$$(\bar{\gamma}\cdot\overline{p}).v(p)$$

$$0$$

2.150 SpinorVBar

SpinorVBar[p, m] denotes a $\bar{v}(p, m)$-spinor that depends on the 4-dimensional momentum p.

2.150.1 See also

Overview, Spinor, SpinorUBar, SpinorU, SpinorV, SpinorUBarD, SpinorUD, SpinorVD, SpinorVBarD.

2.150.2 Examples

```
SpinorVBar[p, m]
```

$$\bar{v}(p, m)$$

```
SpinorVBar[p, m] // FCI // StandardForm

(*Spinor[-Momentum[p], m, 1]*)
```

```
SpinorVBar[p]
```

$$\bar{v}(p)$$

```
SpinorVBar[p] // FCI // StandardForm

(*Spinor[-Momentum[p], 0, 1]*)
```

```
SpinorVBar[p] . GS[p]

DiracEquation[%]
```

$$\bar{v}(p).(\bar{\gamma} \cdot \overline{p})$$

$$0$$

2.151 SpinorUD

SpinorUD[p, m] denotes a $u(p, m)$-spinor that depends on the D-dimensional momentum p.

2.151.1 See also

Overview, Spinor, SpinorUBar, SpinorU, SpinorV, SpinorVBar, SpinorUBarD, SpinorVD, SpinorVBarD.

2.151.2 Examples

```
SpinorUD[p, m]
```

$$u(p, m)$$

```
SpinorUD[p, m] // FCI // StandardForm

(*Spinor[Momentum[p, D], m, 1]*)
```

```
SpinorUD[p]
```

$$u(p)$$

```
SpinorUD[p] // FCI // StandardForm

(*Spinor[Momentum[p, D], 0, 1]*)
```

```
GSD[p] . SpinorUD[p]

DiracEquation[%]
```

$$(\gamma \cdot p).u(p)$$

$$0$$

2.152 SpinorUBarD

SpinorUBarD[p, m] denotes a $\bar{u}(p, m)$-spinor that depends on the D-dimensional momentum p.

2.152.1 See also

Overview, Spinor, SpinorUBar, SpinorU, SpinorV, SpinorVBar, SpinorUD, SpinorVD, SpinorVBarD.

2.152.2 Examples

```
SpinorUBarD[p, m]
```

$$\bar{u}(p, m)$$

```
SpinorUBarD[p, m] // FCI // StandardForm

(*Spinor[Momentum[p, D], m, 1]*)
```

```
SpinorUBarD[p]
```

$$\bar{u}(p)$$

```
SpinorUBarD[p] // FCI // StandardForm

(*Spinor[Momentum[p, D], 0, 1]*)
```

```
SpinorUBarD[p] . GSD[p]

DiracEquation[%]
```

$$\bar{u}(p).(\gamma \cdot p)$$

$$0$$

2.153 SpinorVD

`SpinorVD[p, m]` denotes a $v(p, m)$-spinor that depends on the D-dimensional momentum p.

2.153.1 See also

Overview, Spinor, SpinorUBar, SpinorU, SpinorV, SpinorVBar, SpinorUBarD, SpinorUD, SpinorVBarD.

2.153.2 Examples

```
SpinorVD[p, m]
```

$$v(p, m)$$

```
SpinorVD[p, m] // FCI // StandardForm

(*Spinor[-Momentum[p, D], m, 1]*)
```

```
SpinorVD[p]
```

$$v(p)$$

```
SpinorVD[p] // FCI // StandardForm

(*Spinor[-Momentum[p, D], 0, 1]*)
```

```
GSD[p] . SpinorVD[p]

DiracEquation[%]
```

$$(\gamma \cdot p).v(p)$$

$$0$$

2.154 SpinorVBarD

`SpinorVBarD[p, m]` denotes a $\bar{v}(p, m)$-spinor that depends on the D-dimensional momentum p.

2.154.1 See also

Overview, Spinor, SpinorUBar, SpinorU, SpinorV, SpinorVBar, SpinorUBarD, SpinorUD, SpinorVD.

2.154.2 Examples

```
SpinorVBarD[p, m]
```

$$\bar{v}(p, m)$$

```
SpinorVBarD[p, m] // FCI // StandardForm

(*Spinor[-Momentum[p, D], m, 1]*)
```

```
SpinorVBarD[p]
```

$$\bar{v}(p)$$

```
SpinorVBarD[p] // FCI // StandardForm

(*Spinor[-Momentum[p, D], 0, 1]*)
```

```
SpinorVBarD[p] . GSD[p]

DiracEquation[%]
```

$$\bar{v}(p).(\gamma \cdot p)$$

$$0$$

2.155 SP

SP[a, b] denotes a 4-dimensional scalar product. **SP[a, b]** is transformed into **ScalarProduct[a, b]** by **FeynCalcInternal**.

SP[p] is the same as **SP[p, p]** $(= p^2)$.

2.155.1 See also

Overview, Calc, ExpandScalarProduct, ScalarProduct.

2.155.2 Examples

```
SP[p, q] + SP[q]
```

$$\overline{p} \cdot \overline{q} + \overline{q}^2$$

```
SP[p - q, q + 2 p]
```

$$(\overline{p} - \overline{q}) \cdot (2\overline{p} + \overline{q})$$

```
Calc[ SP[p - q, q + 2 p] ]
```

$$-\overline{p}\cdot\overline{q}+2\overline{p}^2-\overline{q}^2$$

```
ExpandScalarProduct[SP[p - q]]
```

$$-2\left(\overline{p}\cdot\overline{q}\right)+\overline{p}^2+\overline{q}^2$$

```
SP[a, b] // StandardForm

(*SP[a, b]*)
```

```
SP[a, b] // FCI // StandardForm

(*Pair[Momentum[a], Momentum[b]]*)
```

```
SP[a, b] // FCI // FCE // StandardForm

(*SP[a, b]*)
```

2.156 SPD

SPD[a, b] denotes a D-dimensional scalar product.

SPD[a, b] is transformed into **ScalarProduct[a, b, Dimension->D]** by **FeynCalcInternal**.

SPD[p] is the same as **SPD[p, p]** $(=p^2)$.

2.156.1 See also

Overview, PD, Calc, ExpandScalarProduct, ScalarProduct.

2.156.2 Examples

```
SPD[p, q] + SPD[q]
```

$$p\cdot q+q^2$$

```
SPD[p - q, q + 2 p]
```

$$(p-q)\cdot(2p+q)$$

```
Calc[ SPD[p - q, q + 2 p] ]
```

$$-p\cdot q+2p^2-q^2$$

```
ExpandScalarProduct[SPD[p - q]]
```

$$-2(p\cdot q)+p^2+q^2$$

```
SPD[a, b] // StandardForm

(*SPD[a, b]*)
```

```
SPD[a, b] // FCI // StandardForm

(*Pair[Momentum[a, D], Momentum[b, D]]*)
```

```
SPD[a, b] // FCI // FCE // StandardForm

(*SPD[a, b]*)
```

```
FCE[ChangeDimension[SP[p, q], D]] // StandardForm

(*SPD[p, q]*)
```

2.157 SPE

SPE[a, b] denotes a $D-4$-dimensional scalar product. **SPE[a, b]** is transformed into **Pair[Momentum[a, -4 + D], Momentum[b, -4 + D]]** by **FeynCalcInternal**.

SPE[p] is the same as **SPE[p, p]** $(= p^2)$.

2.157.1 See also

Overview, PD, Calc, ExpandScalarProduct, ScalarProduct, SPD.

2.157.2 Examples

```
SPE[p, q] + SPE[q]
```

$$\hat{p} \cdot \hat{q} + \hat{q}^2$$

```
SPE[p - q, q + 2 p]
```

$$(\hat{p} - \hat{q}) \cdot (2\hat{p} + \hat{q})$$

```
Calc[ SPE[p - q, q + 2 p] ]
```

$$-\hat{p} \cdot \hat{q} + 2\hat{p}^2 - \hat{q}^2$$

```
ExpandScalarProduct[SPE[p - q]]
```

$$-2\,(\hat{p} \cdot \hat{q}) + \hat{p}^2 + \hat{q}^2$$

```
SPE[a, b] // StandardForm

(*SPE[a, b]*)
```

```
SPE[a, b] // FCI // StandardForm

(*Pair[Momentum[a, -4 + D], Momentum[b, -4 + D]]*)
```

```
SPE[a, b] // FCI // FCE // StandardForm

(*SPE[a, b]*)
```

```
FCE[ChangeDimension[SP[p, q], D - 4]] // StandardForm

(*SPE[p, q]*)
```

2.158 StandardMatrixElement

`StandardMatrixElement[...]` is the head for matrix element abbreviations.

2.158.1 See also

Overview, OneLoop.

2.158.2 Examples

2.159 SUND

SUND[a, b, c] are the symmetric $SU(N)$ d_{abc}.

2.159.1 See also

Overview, SUNDelta, SUNF, SUNSimplify.

2.159.2 Examples

```
SUND[a, b, c]
```

$$d^{abc}$$

```
SUND[a, b, c, Explicit -> True]
```

$$2\left(\text{tr}(T^a.T^b.T^c)\right)+2\left(\text{tr}(T^b.T^a.T^c)\right)$$

```
SUND[c, a, b]
```

$$d^{abc}$$

```
SUND[a, b, b]
```

$$d^{abb}$$

```
SUNSimplify[SUND[a, b, c] SUND[a, b, c]]
```

$$-2\left(4-C_A^2\right)C_F$$

```
SUNSimplify[SUND[a, b, c] SUND[a, b, c], SUNNToCACF -> False] // Factor2
```

$$\frac{\left(1-N^2\right)\left(4-N^2\right)}{N}$$

```
SUNSimplify[SUND[a, b, c] SUND[e, b, c], SUNNToCACF -> False] // Factor2
```

$$-\frac{\left(4-N^2\right)\delta^{ae}}{N}$$

```
SUND[a, b, c] // StandardForm

(*SUND[a, b, c]*)
```

```
SUND[a, b, c] // FCI // StandardForm

(*SUND[SUNIndex[a], SUNIndex[b], SUNIndex[c]]*)
```

```
SUND[a, b, c] // FCI // FCE // StandardForm

(*SUND[a, b, c]*)
```

2.160 SUNF

SUNF[a, b, c] are the structure constants of $SU(N)$. The arguments **a, b, c** should be of symbolic type.

2.160.1 See also

Overview, SUND, SUNDelta, SUNIndex, SUNSimplify, SUNT, Trick.

2.160.2 Examples

```
SUNF[a, b, c] x + SUNF[b, a, c]

Calc[%]

SUNSimplify[%%]
```

$$x f^{abc} + f^{bac}$$

$$x f^{abc} - f^{abc}$$

$$(x-1) f^{abc}$$

```
SUNF[a, a, b]
% // Calc
```

$$f^{aab}$$

$$0$$

This is a consequence of the usual choice for the normalization of the T_a generators.

```
SUNF[a, b, c, Explicit -> True]
```

$$2i \left(\text{tr}(T^a.T^c.T^b) - \text{tr}(T^a.T^b.T^c)\right)$$

```
SUNSimplify[SUNF[a, b, c] SUNF[a, b, d]]
```

$$C_A \delta^{cd}$$

```
SUNSimplify[SUNF[a, b, c], Explicit -> True]
```

$$-2i \left(\text{tr}(T^a.T^b.T^c) - \text{tr}(T^b.T^a.T^c)\right)$$

```
SUNF[a, b, c] // StandardForm
(*SUNF[a, b, c]*)
```

```
SUNF[a, b, c] // FCI // StandardForm
(*SUNF[SUNIndex[a], SUNIndex[b], SUNIndex[c]]*)
```

```
SUNF[a, b, c] // FCI // FCE // StandardForm

(*SUNF[a, b, c]*)
```

```
SUNF[b, a, c]

% // FCI
```

$$f^{bac}$$

$$-f^{abc}$$

2.161 SUNN

SUNN denotes the number of colors. **Trick[SUNDelta[a, a]]** yields $n_c^2 - 1$.

2.161.1 See also

Overview, SUNSimplify, Trick, SUNIndex, CA, CF.

2.161.2 Examples

```
SUNSimplify[SUNDelta[SUNIndex[a], SUNIndex[a]], SUNNToCACF -> False]
```

$$N^2 - 1$$

2.162 SUNT

SUNT[a] is the $SU(N)$ T^a generator in the fundamental representation. The fundamental indices are implicit.

2.162.1 See also

Overview, CA, CF, SUND, SUNDelta, SUNF, SUNSimplify.

2.162.2 Examples

```
SUNT[a]
```

$$T^a$$

Since T^a is a noncommutative object, products have to separated by a **Dot** (**.**).

```
SUNT[a] . SUNT[b] . SUNT[c]
```

$$T^a.T^b.T^c$$

```
SUNT[a, b, c, d]
```

$$T^a.T^b.T^c.T^d$$

```
SUNSimplify[SUNT[a, b, a], SUNNToCACF -> False]
```

$$-\frac{T^b}{2N}$$

```
SUNSimplify[SUNT[a, b, b, a]]
```

$$C_F^2$$

```
SUNSimplify[SUNT[a, b, a]]
```

$$-\frac{1}{2}T^b\left(C_A-2C_F\right)$$

```
SUNSimplify[SUNT[a, b, a], SUNNToCACF -> False]
```

$$-\frac{T^b}{2N}$$

The normalization of the generators is chosen in the standard way, therefore $\mathrm{Tr}(T^aT^b)=\frac{1}{2}\delta_{ab}$

```
SUNTrace[SUNT[a, b]]
```

$$\frac{\delta^{ab}}{2}$$

In case you want T_f, you need to include a factor **2*Tf**inside the trace.

```
SUNTrace[2 Tf SUNT[a, b]]
```

$$T_f\,\delta^{ab}$$

```
SUNTrace[SUNT[a, b]] // StandardForm
```

$$\frac{1}{2}\,\text{SUNDelta}[\text{SUNIndex}[a], \text{SUNIndex}[b]]$$

```
SUNT[a] // FCI // StandardForm

(*SUNT[SUNIndex[a]]*)
```

```
SUNT[a] // FCI // FCE // StandardForm

(*SUNT[a]*)
```

2.163 SUNTF

SUNTF[{a}, i, j] is the $SU(N)$ T^a generator in the fundamental representation. The fundamental indices are explicit.

2.163.1 See also

Overview, SUNFindex, SUNT, SUNSimplify.

2.163.2 Examples

```
SUNTF[a, i, j]
```

$$T^a_{ij}$$

```
SUNTF[{a, b}, i, j]
```

$$\left(T^a T^b\right)_{ij}$$

SUNTF are c-numbers, hence they are commutative objects and do not require a dot

```
SUNTF[{a, b}, i, j] SUNTF[{c, d}, j, k]
```

$$\left(T^a T^b\right)_{ij} \left(T^c T^d\right)_{jk}$$

```
SUNTF[{a, b}, i, j] SUNTF[{c, d}, j, k] // SUNSimplify
```

$$\left(T^a T^b T^c T^d\right)_{ik}$$

A chain with closed indices is automatically converted into a trace

```
SUNTF[{a, b}, i, j] SUNTF[{c, d}, j, i] // SUNSimplify
```

$$\text{tr}\left(T^a.T^b.T^c.T^d\right)$$

```
SUNDelta[a, b] SUNTF[{a, b}, i, j] SUNTF[{c, d}, j, i] // SUNSimplify
```

$$\frac{1}{2} C_F \delta^{cd}$$

```
SUNTF[{a, b}, i, j] // FCI // StandardForm

(*SUNTF[{SUNIndex[a], SUNIndex[b]}, SUNFIndex[i], SUNFIndex[j]]*)
```

2.164 TC

TC[p] is the temporal component of a 4-vector and is transformed into **TemporalPair[TemporalMomentum[p], ExplicitLorentzIndex[0]]** by **FeynCalcInternal**.

2.164.1 See also

Overview, TemporalPair, TemporalMomentum, FCClearScalarProducts.

2.164.2 Examples

```
TC[p]
```

$$p^0$$

```
TC[p - q]
```

$$(p-q)^0$$

```
FCI[TC[p]] // StandardForm

(*TemporalPair[ExplicitLorentzIndex[0], TemporalMomentum[p]]*)
```

ExpandScalarProduct is used to expand momenta in **TC**

```
ExpandScalarProduct[TC[p - q]]
```

$$p^0 - q^0$$

```
ExpandScalarProduct[TC[p - q]] // StandardForm

(*TemporalPair[ExplicitLorentzIndex[0], TemporalMomentum[p]] -
TemporalPair[ExplicitLorentzIndex[0], TemporalMomentum[q]]*)
```

2.165 TGA

TGA[] can be used as input for γ^0 in 4 dimensions and is transformed into **DiracGamma[ExplicitLorentzIndex[0]]** by **FeynCalcInternal**.

2.165.1 See also

Overview, GA, DiracGamma.

2.165.2 Examples

```
TGA[]
```

$$\bar{\gamma}^0$$

```
TGA[] // FCI // StandardForm

(*DiracGamma[ExplicitLorentzIndex[0]]*)
```

```
TGA[] . TGA[] // DiracSimplify
```

$$1$$

2.166 TemporalMomentum

TemporalMomentum[p] is the head of the temporal component of a 4-momentum p^0. The internal representation of the temporal component p^0 is **TemporalMomentum[p]**.

TemporalMomentum may appear only inside **TemporalPair**s.

2.166.1 See also

Overview, TemporalPair, ExplicitLorentzIndex.

2.166.2 Examples

```
TemporalMomentum[p]
```

$$p$$

```
TemporalMomentum[-q] // StandardForm

(*-TemporalMomentum[q]*)
```

```
TemporalMomentum[p + q]
```

$$p + q$$

```
TemporalMomentum[p + q] // MomentumExpand // StandardForm

(*TemporalMomentum[p] + TemporalMomentum[q]*)
```

```
TemporalMomentum[p + q] // MomentumExpand // MomentumCombine //
StandardForm

(*TemporalMomentum[p + q]*)
```

2.167 TemporalPair

`TemporalPair[ExplicitLorentzIndex[0], TemporalMomentum[p]]` is a special pairing used in the internal representation to denote p^0, the temporal component of a 4-momentum p.

2.167.1 See also

Overview, TemporalPair, TC, ExplicitLorentzIndex.

2.167.2 Examples

```
TemporalPair[ExplicitLorentzIndex[0], TemporalMomentum[p]]
```

$$p^0$$

```
TemporalPair[ExplicitLorentzIndex[0], TemporalMomentum[p + q]]

% // ExpandScalarProduct
```

$$(p+q)^0$$

$$p^0 + q^0$$

2.168 Tf

`Tf` is the color factor T_f. It is $1/2$ for $SU(N)$.

2.168.1 See also

Overview, Tr2, GluonPropagator, GluonSelfEnergy.

2.168.2 Examples

2.169 Zeta2

Zeta2 denotes **Zeta[2]**.

2.169.1 See also

Overview, SimplifyPolyLog.

2.169.2 Examples

```
Zeta2
```

$$\zeta(2)$$

```
N[Zeta2]
```

$$1.64493$$

```
SimplifyPolyLog[Pi^2]
```

$$6\zeta(2)$$

```
Conjugate[Zeta2]
```

$$\zeta(2)$$

2.170 Zeta4

Zeta4 denotes **Zeta[4]**.

2.170.1 See also

Overview, Zeta2, Zeta6, Zeta8, Zeta10.

2.170.2 Examples

```
Zeta4
```

$$\zeta(4)$$

```
N[Zeta4]
```

$$1.08232$$

```
SimplifyPolyLog[Pi^4]
```

$$90\zeta(4)$$

```
Conjugate[Zeta4]
```

$$\zeta(4)$$

2.171 Zeta6

Zeta6 denotes **Zeta[6]**.

2.171.1 See also

Overview, Zeta2, Zeta4, Zeta8, Zeta10.

2.171.2 Examples

```
Zeta6
```

$$\zeta(6)$$

```
N[Zeta6]
```

$$1.01734$$

```
SimplifyPolyLog[Pi^6]
```

$$945\zeta(6)$$

```
Conjugate[Zeta6]
```

$$\zeta(6)$$

2.172 Zeta8

Zeta8 denotes **Zeta[8]**.

2.172.1 See also

Overview, Zeta2, Zeta4, Zeta6, Zeta10.

2.172.2 Examples

```
Zeta8
```

$$\zeta(8)$$

```
N[Zeta8]
```

$$1.00408$$

```
SimplifyPolyLog[Pi^8]
```

$$9450\zeta(8)$$

```
Conjugate[Zeta8]
```

$$\zeta(8)$$

2.173 Zeta10

Zeta10 denotes **Zeta[10]**.

2.173.1 See also

Overview, Zeta8, Zeta6, Zeta4, Zeta2.

2.173.2 Examples

```
Zeta10
```

$$\zeta(10)$$

```
N[Zeta[10]]
```

$$1.00099$$

```
SimplifyPolyLog[Pi^10]
```

$$93555\zeta(10)$$

```
Conjugate[Zeta10]
```

$$\zeta(10)$$

3 Basic functions

3.1 Apart1

`Apart1[expr, x]` is equivalent to **`Apart[expr, x]`**, but it fixes a Mathematica bug relevant when expr contains complex numbers.

3.1.1 See also

Overview, Apart3.

3.1.2 Examples

3.2 Apart3

`Apart3[expr, x]` is equivalent to **`Map2[Factor2, Collect2[Apart1[expr, x], x]]`**.

3.2.1 See also

Overview, Apart1.

3.2.2 Examples

3.3 Cases2

`Cases2[expr, f]` returns a list of all objects in **`expr`** with head **`f`**.

`Cases2[expr, f]` is equivalent to **`Cases2[{expr}, f[___], Infinity]//Union`**.

`Cases2[expr, f, g, ...]` or **`Cases2[expr, {f, g, ...}]`** is equivalent to **`Cases[{expr}, f[___] | g[___] ...]`**.

3.3.1 See also

Overview, Variables2.

3.3.2 Examples

```
Cases2[f[a] + f[b]^2 + f[c, d], f]
```

$$\{f(a), f(b), f(c,d)\}$$

```
Cases2[Sin[x] Sin[y - z] + g[y], Sin, g]
```

$$\{g(y), \sin(x), \sin(y-z)\}$$

```
Cases2[Sin[x] Sin[y - z] + g[x] + g[a, b, c], {Sin, g}]
```

$$\{g(x), g(a,b,c), \sin(x), \sin(y-z)\}$$

```
Cases2[GS[p] . GS[q] + SP[p, p], Dot]
```

$$\{(\bar{\gamma}\cdot\bar{p})\,.\,(\bar{\gamma}\cdot\bar{q})\}$$

3.4 Coefficient2

`Coefficient2[exp, form1, form2, ...]` is like Coefficient, but it also allows to extracts coefficients of **`form1, form2, ...`** sequentially. To specify the power in **`formi`**, write it as **`{var, pow}`**.

To keep the prefactor whose coefficient you extracted you need to set the option **`Prefactor`** to **`True`**.

3.4.1 See also

Overview, Cases2.

3.4.2 Examples

```
ex = y0 + ep y1 + a4 (1/ep x1 + x2 + x3 ep) + a4^2 (1/ep^2 z1 + 1/ep z2 +
z3 + x4 ep)
```

$$\text{a4}^2\left(\frac{\text{z1}}{\text{ep}^2}+\text{ep x4}+\frac{\text{z2}}{\text{ep}}+\text{z3}\right)+\text{a4}\left(\frac{\text{x1}}{\text{ep}}+\text{ep x3}+\text{x2}\right)+\text{ep y1}+\text{y0}$$

```
Coefficient2[ex, a4]
```

$$\frac{x1}{ep} + ep\ x3 + x2$$

```
Coefficient2[ex, a4, 2]
```

$$\frac{z1}{ep^2} + ep\ x4 + \frac{z2}{ep} + z3$$

```
Coefficient2[ex, {a4, 2}]
```

$$\frac{z1}{ep^2} + ep\ x4 + \frac{z2}{ep} + z3$$

```
Coefficient2[ex, {a4, 2}, {ep, -1}]
```

$$z2$$

```
Coefficient2[ex, {a4, 2}, ep]
```

$$x4$$

```
Coefficient2[ex, a4, 2, ep]
```

$$x4$$

```
Coefficient2[ex, {a4, 1}, {ep, 0}]
```

$$x2$$

```
Coefficient2[ex, a4, ep]
```

$$x3$$

```
Coefficient2[ex, {a4, 1}, {ep, 0}, Prefactor -> True]
```

a4 x2

3.5 Combine

Combine[expr] puts terms in a sum over a common denominator and cancels factors in the result. **Combine** is similar to **Together**, but accepts the option **Expanding** and works usually better than **Together** for polynomials involving rationals with sums in the denominator.

3.5.1 See also

Overview, Factor2.

3.5.2 Examples

```
Combine[((a - b) (c - d))/e + g]
```

$$\frac{(a-b)(c-d)+eg}{e}$$

Here the result from **Together** where the numerator is automatically expanded.

```
Together[((a - b) (c - d))/e + g]
```

$$\frac{ac-ad-bc+bd+eg}{e}$$

If the option **Expanding** is set to **True**, the result of **Combine** is the same as **Together**, but uses a slightly different algorithm.

```
Combine[((a - b) (c - d))/e + g, Expanding -> True]
```

$$\frac{ac-ad-bc+bd+eg}{e}$$

3.6 Complement1

Complement1[l1, l2] where **l1** and **l2** are lists returns a list of elements from **l1** not in**l2**. Multiple occurrences of an element in **l1** are kept and multiple occurrences of an element in **l2** are dropped if present in **l1**.

3.6.1 See also

Overview

3.6.2 Examples

```
Complement[{a, b, c, d, e, f, e}, {a, b, c, d}]
```

$$\{e, f\}$$

```
Complement1[{a, b, c, d, e, f, e}, {a, b, c, d}]
```

$$\{e, f, e\}$$

3.7 Collect2

Collect2[expr, x] collects together terms which are not free of any occurrence of **x**.

Collect2[expr, {x1, x2, ...}] (or also **Collect2[expr, x1, x2, ...]**) collects together terms which are not free of any occurrence of **x1, x2, ...**.

The coefficients are put over a common denominator. If **expr** is expanded before collecting depends on the option **Factoring**, which may be set to **Factor**, **Factor2**, or any other function, which is applied to the coefficients. If **expr** is already expanded with respect to **x** (**x1**, **x2**, ...), the option **Expanding** can be set to **False.**

3.7.1 See also

Overview, Isolate.

3.7.2 Examples

```
Collect2[t1 = a + r a + k^2 f[a] - k f[a] + x/2 - y/w, a]
```

$$(k-1)kf(a)+a(r+1)+\frac{wx-2y}{2w}$$

```
Collect2[t1, a, Factoring -> False]
```

$$\left(k^2-k\right)f(a)+a(r+1)-\frac{y}{w}+\frac{x}{2}$$

```
Collect2[t1, a, Factoring -> Factor]
```

$$(k-1)kf(a)+a(r+1)+\frac{wx-2y}{2w}$$

```
Collect2[t1, a, Factoring -> Simplify]
```

$$(k-1)kf(a)+a(r+1)-\frac{y}{w}+\frac{x}{2}$$

```
Collect2[2 a (b - a) (h - 1) - b^2 (e a - c) + b^2, {a, b}]
```

$$-2a^2(h-1)-ab^2e+2ab(h-1)+b^2(c+1)$$

```
Collect2[Expand[(a - b - c - d)^5], a, IsolateNames -> KK]

FRH[%]
```

$$a^5-5a^4\,\mathrm{KK}(19)+10a^3\,\mathrm{KK}(20)-10a^2\,\mathrm{KK}(22)+5a\,\mathrm{KK}(21)-\mathrm{KK}(23)$$

$$a^5-5a^4(b+c+d)+10a^3(b+c+d)^2-10a^2(b+c+d)^3+5a(b+c+d)^4-(b+c+d)^5$$

The option **Head** is useful for subsequent manipulations of the output

```
Collect2[Expand[(a - b - c - d)^5], a, Head -> h]
```

$$h\left(a^5\right)-5h\left(a^4\right)(b+c+d)+10h\left(a^3\right)(b+c+d)^2-10h\left(a^2\right)(b+c+d)^3+5h(a)(b+c+d)^4-(b+c+d)^5$$

```
Collect2[Expand[(a - b - c - d)^5], a, Head -> {h1, h2}]
```

$$\begin{aligned}&\text{h2}\left(1,\text{h1}\left(a^5\right)\right)+\text{h2}\left(-5(b+c+d),\text{h1}\left(a^4\right)\right)+\text{h2}\left(10(b+c+d)^2,\right.\\&\left.\text{h1}\left(a^3\right)\right)+\text{h2}\left(-10(b+c+d)^3,\text{h1}\left(a^2\right)\right)+\text{h2}\left(5(b+c+d)^4,\text{h1}(a)\right)+\text{h2}\left(-(b+c+d)^5,1\right)\end{aligned}$$

```
Collect2[Expand[(a - b - c - d)^5], a, Head -> {Identity, h2}]

Cases2[%, h2]
```

$$\begin{aligned}&\text{h2}\left(1,a^5\right)+\text{h2}\left(-5(b+c+d),a^4\right)+\text{h2}\left(10(b+c+d)^2,\right.\\&\left.a^3\right)+\text{h2}\left(-10(b+c+d)^3,a^2\right)+\text{h2}\left(5(b+c+d)^4,a\right)+\text{h2}\left(-(b+c+d)^5,1\right)\end{aligned}$$

$$\begin{aligned}&\left\{\text{h2}\left(1,a^5\right),\text{h2}\left(-5(b+c+d),a^4\right),\text{h2}\left(10(b+c+d)^2,a^3\right),\right.\\&\left.\text{h2}\left(-10(b+c+d)^3,a^2\right),\text{h2}\left(5(b+c+d)^4,a\right),\text{h2}\left(-(b+c+d)^5,1\right)\right\}\end{aligned}$$

It is possible to use different factoring functions

```
Clear[fun]

Collect2[Expand[(a - b - c)^3], a, Factoring -> fun]

% /. fun -> FactorTerms
```

$$a^3\,\text{fun}(1)+a^2\,\text{fun}(-3b-3c)+a\,\text{fun}\left(3b^2+6bc+3c^2\right)+\text{fun}\left(-b^3-3b^2c-3bc^2-c^3\right)$$

$$a^3-3a^2(b+c)+3a\left(b^2+2bc+c^2\right)-b^3-3b^2c-3bc^2-c^3$$

Another neat trick is to nest **`Collect2`** using the **`Factoring`** option

```
Collect2[Expand[((a1 + a2 + a3)^3 - (b1 + b2 + b3)^3 - (c1 + c2 +
c3)^3)^2], {a1, a2, a3},
  Factoring -> Function[x, Collect2[x, {b1, c1}]]]
```

$$
\begin{aligned}
&\mathrm{a1}^6 + 6\,\mathrm{a2}\,\mathrm{a1}^5 + 6\,\mathrm{a3}\,\mathrm{a1}^5 + 15\,\mathrm{a2}^2\,\mathrm{a1}^4 + 15\,\mathrm{a3}^2\,\mathrm{a1}^4 + 30\,\mathrm{a2}\,\mathrm{a3}\,\mathrm{a1}^4 + 20\,\mathrm{a2}^3\,\mathrm{a1}^3 + 20\,\mathrm{a3}^3\,\mathrm{a1}^3 + 60\,\mathrm{a2}\,\mathrm{a3}^2\,\mathrm{a1}^3 \\
&+ 60\,\mathrm{a2}^2\,\mathrm{a3}\,\mathrm{a1}^3 + \big(-2\,\mathrm{b1}^3 - 6(\mathrm{b2}+\mathrm{b3})\,\mathrm{b1}^2 - 6(\mathrm{b2}+\mathrm{b3})^2\,\mathrm{b1} - 2\,\mathrm{c1}^3 - 6\,\mathrm{c1}(\mathrm{c2}+\mathrm{c3})^2 - 6\,\mathrm{c1}^2(\mathrm{c2}+\mathrm{c3}) \\
&- 2(\mathrm{b2}+\mathrm{b3}+\mathrm{c2}+\mathrm{c3})\left(\mathrm{b2}^2 + 2\,\mathrm{b3}\,\mathrm{b2} - \mathrm{c2}\,\mathrm{b2} - \mathrm{c3}\,\mathrm{b2} + \mathrm{b3}^2 + \mathrm{c2}^2 + \mathrm{c3}^2 - \mathrm{b3}\,\mathrm{c2} - \mathrm{b3}\,\mathrm{c3} + 2\,\mathrm{c2}\,\mathrm{c3}\right)\big)\,\mathrm{a1}^3 \\
&+ 15\,\mathrm{a2}^4\,\mathrm{a1}^2 + 15\,\mathrm{a3}^4\,\mathrm{a1}^2 + 60\,\mathrm{a2}\,\mathrm{a3}^3\,\mathrm{a1}^2 + 90\,\mathrm{a2}^2\,\mathrm{a3}^2\,\mathrm{a1}^2 + 60\,\mathrm{a2}^3\,\mathrm{a3}\,\mathrm{a1}^2 \\
&+ \mathrm{a2}\left(-6\,\mathrm{b1}^3 - 18(\mathrm{b2}+\mathrm{b3})\,\mathrm{b1}^2 - 18(\mathrm{b2}+\mathrm{b3})^2\,\mathrm{b1} - 6\,\mathrm{c1}^3 - 18\,\mathrm{c1}(\mathrm{c2}+\mathrm{c3})^2 - 18\,\mathrm{c1}^2(\mathrm{c2}+\mathrm{c3})\right. \\
&\left. - 6(\mathrm{b2}+\mathrm{b3}+\mathrm{c2}+\mathrm{c3})\left(\mathrm{b2}^2 + 2\,\mathrm{b3}\,\mathrm{b2} - \mathrm{c2}\,\mathrm{b2} - \mathrm{c3}\,\mathrm{b2} + \mathrm{b3}^2 + \mathrm{c2}^2 + \mathrm{c3}^2 - \mathrm{b3}\,\mathrm{c2} - \mathrm{b3}\,\mathrm{c3} + 2\,\mathrm{c2}\,\mathrm{c3}\right)\right)\,\mathrm{a1}^2 \\
&+ \mathrm{a3}\left(-6\,\mathrm{b1}^3 - 18(\mathrm{b2}+\mathrm{b3})\,\mathrm{b1}^2 - 18(\mathrm{b2}+\mathrm{b3})^2\,\mathrm{b1} - 6\,\mathrm{c1}^3 - 18\,\mathrm{c1}(\mathrm{c2}+\mathrm{c3})^2 - 18\,\mathrm{c1}^2(\mathrm{c2}+\mathrm{c3})\right. \\
&\left. - 6(\mathrm{b2}+\mathrm{b3}+\mathrm{c2}+\mathrm{c3})\left(\mathrm{b2}^2 + 2\,\mathrm{b3}\,\mathrm{b2} - \mathrm{c2}\,\mathrm{b2} - \mathrm{c3}\,\mathrm{b2} + \mathrm{b3}^2 + \mathrm{c2}^2 + \mathrm{c3}^2 - \mathrm{b3}\,\mathrm{c2} - \mathrm{b3}\,\mathrm{c3} + 2\,\mathrm{c2}\,\mathrm{c3}\right)\right)\,\mathrm{a1}^2 \\
&+ 6\,\mathrm{a2}^5\,\mathrm{a1} + 6\,\mathrm{a3}^5\,\mathrm{a1} + 30\,\mathrm{a2}\,\mathrm{a3}^4\,\mathrm{a1} + 60\,\mathrm{a2}^2\,\mathrm{a3}^3\,\mathrm{a1} + 60\,\mathrm{a2}^3\,\mathrm{a3}^2\,\mathrm{a1} + 30\,\mathrm{a2}^4\,\mathrm{a3}\,\mathrm{a1} \\
&+ \mathrm{a2}\,\mathrm{a3}\left(-12\,\mathrm{b1}^3 - 36(\mathrm{b2}+\mathrm{b3})\,\mathrm{b1}^2 - 36(\mathrm{b2}+\mathrm{b3})^2\,\mathrm{b1} - 12\,\mathrm{c1}^3 - 36\,\mathrm{c1}(\mathrm{c2}+\mathrm{c3})^2 - 36\,\mathrm{c1}^2(\mathrm{c2}+\mathrm{c3})\right. \\
&\left. - 12(\mathrm{b2}+\mathrm{b3}+\mathrm{c2}+\mathrm{c3})\left(\mathrm{b2}^2 + 2\,\mathrm{b3}\,\mathrm{b2} - \mathrm{c2}\,\mathrm{b2} - \mathrm{c3}\,\mathrm{b2} + \mathrm{b3}^2 + \mathrm{c2}^2 + \mathrm{c3}^2 - \mathrm{b3}\,\mathrm{c2} - \mathrm{b3}\,\mathrm{c3} + 2\,\mathrm{c2}\,\mathrm{c3}\right)\right)\,\mathrm{a1} \\
&+ \mathrm{a2}^2\left(-6\,\mathrm{b1}^3 - 18(\mathrm{b2}+\mathrm{b3})\,\mathrm{b1}^2 - 18(\mathrm{b2}+\mathrm{b3})^2\,\mathrm{b1} - 6\,\mathrm{c1}^3 - 18\,\mathrm{c1}(\mathrm{c2}+\mathrm{c3})^2 - 18\,\mathrm{c1}^2(\mathrm{c2}+\mathrm{c3})\right. \\
&\left. - 6(\mathrm{b2}+\mathrm{b3}+\mathrm{c2}+\mathrm{c3})\left(\mathrm{b2}^2 + 2\,\mathrm{b3}\,\mathrm{b2} - \mathrm{c2}\,\mathrm{b2} - \mathrm{c3}\,\mathrm{b2} + \mathrm{b3}^2 + \mathrm{c2}^2 + \mathrm{c3}^2 - \mathrm{b3}\,\mathrm{c2} - \mathrm{b3}\,\mathrm{c3} + 2\,\mathrm{c2}\,\mathrm{c3}\right)\right)\,\mathrm{a1} \\
&+ \mathrm{a3}^2\left(-6\,\mathrm{b1}^3 - 18(\mathrm{b2}+\mathrm{b3})\,\mathrm{b1}^2 - 18(\mathrm{b2}+\mathrm{b3})^2\,\mathrm{b1} - 6\,\mathrm{c1}^3 - 18\,\mathrm{c1}(\mathrm{c2}+\mathrm{c3})^2 - 18\,\mathrm{c1}^2(\mathrm{c2}+\mathrm{c3})\right. \\
&\left. - 6(\mathrm{b2}+\mathrm{b3}+\mathrm{c2}+\mathrm{c3})\left(\mathrm{b2}^2 + 2\,\mathrm{b3}\,\mathrm{b2} - \mathrm{c2}\,\mathrm{b2} - \mathrm{c3}\,\mathrm{b2} + \mathrm{b3}^2 + \mathrm{c2}^2 + \mathrm{c3}^2 - \mathrm{b3}\,\mathrm{c2} - \mathrm{b3}\,\mathrm{c3} + 2\,\mathrm{c2}\,\mathrm{c3}\right)\right)\,\mathrm{a1} \\
&+ \mathrm{a2}^6 + \mathrm{a3}^6 + \mathrm{b1}^6 + \mathrm{c1}^6 + 6\,\mathrm{a2}\,\mathrm{a3}^5 + 15\,\mathrm{a2}^2\,\mathrm{a3}^4 + 20\,\mathrm{a2}^3\,\mathrm{a3}^3 + 2\,\mathrm{b1}^3\,\mathrm{c1}^3 \\
&+ 6\,\mathrm{b1}(\mathrm{b2}+\mathrm{b3})^2\,\mathrm{c1}^3 + 6\,\mathrm{b1}^2(\mathrm{b2}+\mathrm{b3})\,\mathrm{c1}^3 + 15\,\mathrm{a2}^4\,\mathrm{a3}^2 + 15\,\mathrm{b1}^4(\mathrm{b2}+\mathrm{b3})^2 + 15\,\mathrm{c1}^4(\mathrm{c2}+\mathrm{c3})^2 \\
&+ 6\,\mathrm{b1}^3\,\mathrm{c1}(\mathrm{c2}+\mathrm{c3})^2 + 18\,\mathrm{b1}(\mathrm{b2}+\mathrm{b3})^2\,\mathrm{c1}(\mathrm{c2}+\mathrm{c3})^2 + 18\,\mathrm{b1}^2(\mathrm{b2}+\mathrm{b3})\,\mathrm{c1}(\mathrm{c2}+\mathrm{c3})^2 \\
&+ (\mathrm{b2}+\mathrm{b3}+\mathrm{c2}+\mathrm{c3})^2\left(\mathrm{b2}^2 + 2\,\mathrm{b3}\,\mathrm{b2} - \mathrm{c2}\,\mathrm{b2} - \mathrm{c3}\,\mathrm{b2} + \mathrm{b3}^2 + \mathrm{c2}^2 + \mathrm{c3}^2 - \mathrm{b3}\,\mathrm{c2} - \mathrm{b3}\,\mathrm{c3} + 2\,\mathrm{c2}\,\mathrm{c3}\right)^2 + 6\,\mathrm{a2}^5\,\mathrm{a3} \\
&+ 6\,\mathrm{b1}^5(\mathrm{b2}+\mathrm{b3}) + 6\,\mathrm{c1}^5(\mathrm{c2}+\mathrm{c3}) + 6\,\mathrm{b1}^3\,\mathrm{c1}^2(\mathrm{c2}+\mathrm{c3}) + 18\,\mathrm{b1}(\mathrm{b2}+\mathrm{b3})^2\,\mathrm{c1}^2(\mathrm{c2}+\mathrm{c3}) + 18\,\mathrm{b1}^2(\mathrm{b2}+\mathrm{b3})\,\mathrm{c1}^2(\mathrm{c2}+\mathrm{c3}) \\
&+ 6\,\mathrm{b1}(\mathrm{b2}+\mathrm{b3})^2(\mathrm{b2}+\mathrm{b3}+\mathrm{c2}+\mathrm{c3})\left(\mathrm{b2}^2 + 2\,\mathrm{b3}\,\mathrm{b2} - \mathrm{c2}\,\mathrm{b2} - \mathrm{c3}\,\mathrm{b2} + \mathrm{b3}^2 + \mathrm{c2}^2 + \mathrm{c3}^2 - \mathrm{b3}\,\mathrm{c2} - \mathrm{b3}\,\mathrm{c3} + 2\,\mathrm{c2}\,\mathrm{c3}\right) \\
&+ 6\,\mathrm{c1}(\mathrm{c2}+\mathrm{c3})^2(\mathrm{b2}+\mathrm{b3}+\mathrm{c2}+\mathrm{c3})\left(\mathrm{b2}^2 + 2\,\mathrm{b3}\,\mathrm{b2} - \mathrm{c2}\,\mathrm{b2} - \mathrm{c3}\,\mathrm{b2} + \mathrm{b3}^2 + \mathrm{c2}^2 + \mathrm{c3}^2 - \mathrm{b3}\,\mathrm{c2} - \mathrm{b3}\,\mathrm{c3} + 2\,\mathrm{c2}\,\mathrm{c3}\right) \\
&+ 2\,\mathrm{b1}^3\left(10\,\mathrm{b2}^3 + 30\,\mathrm{b3}\,\mathrm{b2}^2 + 30\,\mathrm{b3}^2\,\mathrm{b2} + 10\,\mathrm{b3}^3 + \mathrm{c2}^3 + \mathrm{c3}^3 + 3\,\mathrm{c2}\,\mathrm{c3}^2 + 3\,\mathrm{c2}^2\,\mathrm{c3}\right) \\
&+ 3\,\mathrm{b1}^2(\mathrm{b2}+\mathrm{b3})\left(5\,\mathrm{b2}^3 + 15\,\mathrm{b3}\,\mathrm{b2}^2 + 15\,\mathrm{b3}^2\,\mathrm{b2} + 5\,\mathrm{b3}^3 + 2\,\mathrm{c2}^3 + 2\,\mathrm{c3}^3 + 6\,\mathrm{c2}\,\mathrm{c3}^2 + 6\,\mathrm{c2}^2\,\mathrm{c3}\right) \\
&+ 3\,\mathrm{c1}^2(\mathrm{c2}+\mathrm{c3})\left(2\,\mathrm{b2}^3 + 6\,\mathrm{b3}\,\mathrm{b2}^2 + 6\,\mathrm{b3}^2\,\mathrm{b2} + 2\,\mathrm{b3}^3 + 5\,\mathrm{c2}^3 + 5\,\mathrm{c3}^3 + 15\,\mathrm{c2}\,\mathrm{c3}^2 + 15\,\mathrm{c2}^2\,\mathrm{c3}\right) \\
&+ 2\,\mathrm{c1}^3\left(\mathrm{b2}^3 + 3\,\mathrm{b3}\,\mathrm{b2}^2 + 3\,\mathrm{b3}^2\,\mathrm{b2} + \mathrm{b3}^3 + 10\,\mathrm{c2}^3 + 10\,\mathrm{c3}^3 + 30\,\mathrm{c2}\,\mathrm{c3}^2 + 30\,\mathrm{c2}^2\,\mathrm{c3}\right) \\
&+ \mathrm{a2}\,\mathrm{a3}^2\left(-6\,\mathrm{b1}^3 - 18(\mathrm{b2}+\mathrm{b3})\,\mathrm{b1}^2 - 18(\mathrm{b2}+\mathrm{b3})^2\,\mathrm{b1} - 6\,\mathrm{c1}^3 - 18\,\mathrm{c1}(\mathrm{c2}+\mathrm{c3})^2 - 18\,\mathrm{c1}^2(\mathrm{c2}+\mathrm{c3})\right. \\
&\left. - 6(\mathrm{b2}+\mathrm{b3}+\mathrm{c2}+\mathrm{c3})\left(\mathrm{b2}^2 + 2\,\mathrm{b3}\,\mathrm{b2} - \mathrm{c2}\,\mathrm{b2} - \mathrm{c3}\,\mathrm{b2} + \mathrm{b3}^2 + \mathrm{c2}^2 + \mathrm{c3}^2 - \mathrm{b3}\,\mathrm{c2} - \mathrm{b3}\,\mathrm{c3} + 2\,\mathrm{c2}\,\mathrm{c3}\right)\right) \\
&+ \mathrm{a2}^2\,\mathrm{a3}\left(-6\,\mathrm{b1}^3 - 18(\mathrm{b2}+\mathrm{b3})\,\mathrm{b1}^2 - 18(\mathrm{b2}+\mathrm{b3})^2\,\mathrm{b1} - 6\,\mathrm{c1}^3 - 18\,\mathrm{c1}(\mathrm{c2}+\mathrm{c3})^2 - 18\,\mathrm{c1}^2(\mathrm{c2}+\mathrm{c3})\right. \\
&\left. - 6(\mathrm{b2}+\mathrm{b3}+\mathrm{c2}+\mathrm{c3})\left(\mathrm{b2}^2 + 2\,\mathrm{b3}\,\mathrm{b2} - \mathrm{c2}\,\mathrm{b2} - \mathrm{c3}\,\mathrm{b2} + \mathrm{b3}^2 + \mathrm{c2}^2 + \mathrm{c3}^2 - \mathrm{b3}\,\mathrm{c2} - \mathrm{b3}\,\mathrm{c3} + 2\,\mathrm{c2}\,\mathrm{c3}\right)\right) \\
&+ \mathrm{a2}^3\left(-2\,\mathrm{b1}^3 - 6(\mathrm{b2}+\mathrm{b3})\,\mathrm{b1}^2 - 6(\mathrm{b2}+\mathrm{b3})^2\,\mathrm{b1} - 2\,\mathrm{c1}^3 - 6\,\mathrm{c1}(\mathrm{c2}+\mathrm{c3})^2 - 6\,\mathrm{c1}^2(\mathrm{c2}+\mathrm{c3})\right. \\
&\left. - 2(\mathrm{b2}+\mathrm{b3}+\mathrm{c2}+\mathrm{c3})\left(\mathrm{b2}^2 + 2\,\mathrm{b3}\,\mathrm{b2} - \mathrm{c2}\,\mathrm{b2} - \mathrm{c3}\,\mathrm{b2} + \mathrm{b3}^2 + \mathrm{c2}^2 + \mathrm{c3}^2 - \mathrm{b3}\,\mathrm{c2} - \mathrm{b3}\,\mathrm{c3} + 2\,\mathrm{c2}\,\mathrm{c3}\right)\right) \\
&+ \mathrm{a3}^3\left(-2\,\mathrm{b1}^3 - 6(\mathrm{b2}+\mathrm{b3})\,\mathrm{b1}^2 - 6(\mathrm{b2}+\mathrm{b3})^2\,\mathrm{b1} - 2\,\mathrm{c1}^3 - 6\,\mathrm{c1}(\mathrm{c2}+\mathrm{c3})^2 - 6\,\mathrm{c1}^2(\mathrm{c2}+\mathrm{c3})\right. \\
&\left. - 2(\mathrm{b2}+\mathrm{b3}+\mathrm{c2}+\mathrm{c3})\left(\mathrm{b2}^2 + 2\,\mathrm{b3}\,\mathrm{b2} - \mathrm{c2}\,\mathrm{b2} - \mathrm{c3}\,\mathrm{b2} + \mathrm{b3}^2 + \mathrm{c2}^2 + \mathrm{c3}^2 - \mathrm{b3}\,\mathrm{c2} - \mathrm{b3}\,\mathrm{c3} + 2\,\mathrm{c2}\,\mathrm{c3}\right)\right)
\end{aligned}
$$

The options **`IsolateFast`** allows to save some time when Isolating prefactors, provided that no factoring is involved.

```
ClearAll[h, g, a, b, c];

exp = Sum[h[i], {i, 1, 200000}]*a + Sum[g[i], {i, 1, 200000}]*b + Sum[j[i],
{i, 1, 200000}]*c;
```

```
AbsoluteTiming[Collect2[exp, {a, b, c}, Factoring -> False, IsolateNames ->
KK, Expanding -> False]]
```

$$\{2.28593, a\ \mathrm{KK}(28) + b\ \mathrm{KK}(29) + c\ \mathrm{KK}(27)\}$$

```
AbsoluteTiming[Collect2[exp, {a, b, c}, Factoring -> False, IsolateNames ->
KK, IsolateFast -> True, Expanding -> False]]
```

$$\{1.38613, a\ \mathrm{KK}(28) + b\ \mathrm{KK}(29) + c\ \mathrm{KK}(27)\}$$

```
ClearAll[exp]
```

3.8 Collect3

`Collect3[expr, {x, y, ...}]` collects terms involving the same powers of monomials x^{n_1}, y^{n_2}, ...

The option **`Factor`** can bet set to **`True`** or **`False`**, which factors the coefficients.

The option **`Head`** (default **`Plus`**) determines the applied function to the list of monomials multiplied by their coefficients.

3.8.1 See also

Overview, Collect2, Isolate.

3.8.2 Examples

```
Collect3[2 a (b - a) (h - 1) - b^2 (e a - c) + b^2, {a, b}]
```

$$-2a^2h + 2a^2 - ab^2e + 2abh - 2ab + b^2c + b^2$$

```
Collect3[Expand[(a - b - c - d)^5], {a}]
```

$$\begin{aligned}
&a^5 - 5a^4b - 5a^4c - 5a^4d + 10a^3b^2 + 20a^3bc + 20a^3bd + 10a^3c^2 + 20a^3cd + 10a^3d^2 \\
&\quad - 10a^2b^3 - 30a^2b^2c - 30a^2b^2d - 30a^2bc^2 - 60a^2bcd - 30a^2bd^2 - 10a^2c^3 - 30a^2c^2d \\
&\quad - 30a^2cd^2 - 10a^2d^3 + 5ab^4 + 20ab^3c + 20ab^3d + 30ab^2c^2 + 60ab^2cd + 30ab^2d^2 + 20abc^3 \\
&\quad + 60abc^2d + 60abcd^2 + 20abd^3 + 5ac^4 + 20ac^3d + 30ac^2d^2 + 20acd^3 + 5ad^4 - b^5 \\
&\quad - 5b^4c - 5b^4d - 10b^3c^2 - 20b^3cd - 10b^3d^2 - 10b^2c^3 - 30b^2c^2d - 30b^2cd^2 - 10b^2d^3 \\
&\quad - 5bc^4 - 20bc^3d - 30bc^2d^2 - 20bcd^3 - 5bd^4 - c^5 - 5c^4d - 10c^3d^2 - 10c^2d^3 - 5cd^4 - d^5
\end{aligned}$$

3.9 DataType

DataType[exp, type] = True defines the object **exp** to have data-type **type**.

DataType[exp1, exp2, ..., type] defines the objects **exp1, exp2, ...** to have data-type **type**.

The default setting is **DataType[__, _] := False**.

To assign a certain data-type, do, e.g., **DataType[x, PositiveInteger] = True**. Currently used **DataTypes**:

- **NonCommutative**
- **PositiveInteger**
- **NegativeInteger**
- **PositiveNumber**
- **FreeIndex**
- **GrassmannParity**
- **FCTensor**
- **ImplicitDiracIndex**
- **ImplicitPauliIndex**
- **ImplicitSUNFIndex**

If loaded, PHI adds the **DataTypes**: **UMatrix**, **UScalar**.

3.9.1 See also

Overview, DeclareNonCommutative, NonCommutative, PositiveInteger, NegativeInteger, PositiveNumber, FreeIndex, GrassmannParity, FCTensor, ImplicitDiracIndex, ImplicitPauliIndex, ImplicitSUNFIndex.

3.9.2 Examples

NonCommutative is just a data-type.

```
DataType[f, g, NonCommutative] = True;

t = f . g - g . (2 a) . f
```

$$f.g - g.(2a).f$$

Since **f** and **g** have **DataType NonCommutative**, the function **DotSimplify** extracts only **a** out of the noncommutative product.

```
DotSimplify[t]
```

$$f.g - 2ag.f$$

```
DataType[m, odd] = DataType[a, even] = True;

ptest1[x_] := x /. (-1)^n_ /; DataType[n, odd] :> -1;

ptest2[x_] := x /. (-1)^n_ /; DataType[n, even] :> 1;

t = (-1)^m + (-1)^a + (-1)^z
```

$$(-1)^a + (-1)^m + (-1)^z$$

```
ptest1[t]

ptest2[%]
```

$$(-1)^a + (-1)^z - 1$$

$$(-1)^z$$

```
Clear[ptest1, ptest2, t, a, m];
```

```
DataType[m, integer] = True;

f[x_] := x /. {(-1)^p_ /; DataType[p, integer] :> 1};
```

```
test = (-1)^m + (-1)^n x
```

$$(-1)^m + (-1)^n x$$

```
f[test]
```

$$(-1)^n x + 1$$

```
Clear[f, test];

DataType[f, g, NonCommutative] = False;

DataType[m, odd] = DataType[a, even] = False;
```

Certain FeynCalc objects have **DataType PositiveInteger** set to **True**.

```
DataType[OPEm, PositiveInteger]
```

True

PowerSimplify uses the DataType information.

```
PowerSimplify[ (-1)^(2 OPEm)]
```

1

```
PowerSimplify[ (- SO[q])^OPEm]
```

$$(\Delta \cdot q)^m e^{2i\pi m\left\lfloor -\frac{\arg(\Delta\cdot q)}{2\pi}\right\rfloor + i\pi m}$$

3.10 Expand2

Expand2[exp, x] expands all sums containing **x**.

Expand2[exp, {x1, x2, ...}] expands all sums containing **x1, x2, ...**.

3.10.1 See also

Overview, ExpandAll2.

3.10.2 Examples

```
Expand2[(x1 + x2 + x3) (2 x1 + 3 x2) + (y1 + y2 + y3) (2 y1 + 3 y2)]
```

$$2\,\text{x1}^2 + 5\,\text{x1}\,\text{x2} + 2\,\text{x1}\,\text{x3} + 3\,\text{x2}^2 + 3\,\text{x2}\,\text{x3} + 2\,\text{y1}^2 + 5\,\text{y1}\,\text{y2} + 2\,\text{y1}\,\text{y3} + 3\,\text{y2}^2 + 3\,\text{y2}\,\text{y3}$$

```
Expand2[(x1 + x2 + x3) (2 x1 + 3 x2) + (y1 + y2 + y3) (2 y1 + 3 y2), {y1,
y2}]
```

$$(2\,x1 + 3\,x2)(x1 + x2 + x3) + 2\,y1^2 + 5\,y1\,y2 + 2\,y1\,y3 + 3\,y2^2 + 3\,y2\,y3$$

```
Expand2[(x1 + x2 + x3) (2 x1 + 3 x2) + (y1 + y2 + y3) (2 y1 + 3 y2), {x1,
x2}]
```

$$2\,x1^2 + 5\,x1\,x2 + 2\,x1\,x3 + 3\,x2^2 + 3\,x2\,x3 + (2\,y1 + 3\,y2)(y1 + y2 + y3)$$

3.11 ExpandAll2

`ExpandAll2[exp]` is similar to **`ExpandAll`**, but much faster on simple structures.

3.11.1 See also

Overview

3.11.2 Examples

Benchmark against the standard ExpandAll

```
exp = Sum[p[i], {i, 1, 100}] Sum[q[i], {i, 1, 1000}];
```

```
AbsoluteTiming[res1 = ExpandAll[exp];]
```

$$\{0.679579, \text{Null}\}$$

```
AbsoluteTiming[res2 = ExpandAll2[exp];]
```

$$\{0.250011, \text{Null}\}$$

```
res1 === res2
```

True

```
ClearAll[exp, res1, res2]
```

3.12 Explicit

Explicit[exp] inserts explicit expressions of **GluonVertex**, **Twist2GluonOperator**, **SUNF** etc. in **exp**.

To rewrite the $SU(N)$ structure constants in terms of traces, please set the corresponding options **SUNF** or **SUND** to **True**.

Explicit is also an option for **FieldStrength**, **GluonVertex**, **SUNF**, **Twist2GluonOperator** etc. If set to **True** the full form of the operator is inserted.

3.12.1 See also

Overview, GluonVertex, Twist2GluonOperator.

3.12.2 Examples

```
gv = GluonVertex[p, \[Mu], a, q, \[Nu], b, r, \[Rho], c]
```

$$f^{abc} V^{\mu\nu\rho}(p,q,r)$$

```
Explicit[gv]
```

$$g_s f^{abc} \left(g^{\mu\nu} \left(p^\rho - q^\rho\right) + g^{\mu\rho} \left(r^\nu - p^\nu\right) + g^{\nu\rho} \left(q^\mu - r^\mu\right)\right)$$

```
Explicit[gv, SUNF -> True]
```

$$2ig_s \left(\text{tr}\left(T^a.T^c.T^b\right) - \text{tr}\left(T^a.T^b.T^c\right)\right) \left(g^{\mu\nu} \left(p^\rho - q^\rho\right) + g^{\mu\rho} \left(r^\nu - p^\nu\right) + g^{\nu\rho} \left(q^\mu - r^\mu\right)\right)$$

```
Twist2GluonOperator[p, \[Mu], a, \[Nu], b]

Explicit[%]
```

$$\frac{1}{2}\left((-1)^m+1\right)\delta^{ab}\left(O^{\text{G2}}_{\mu\nu}(p)\right)$$

$$\frac{1}{2}\left((-1)^m+1\right)\delta^{ab}(\Delta\cdot p)^{m-2}\left(g^{\mu\nu}(\Delta\cdot p)^2+p^2\Delta^\mu\Delta^\nu-(\Delta\cdot p)\left(\Delta^\nu p^\mu+\Delta^\mu p^\nu\right)\right)$$

```
FieldStrength[\[Mu], \[Nu], a]
```

```
Explicit[%]
```

$$F^a_{\mu\nu}$$

$$g_s f^{\text{ab19 c20}}A^{\text{b19}}_\mu.A^{\text{c20}}_\nu+\left(\partial_\mu A^a_\nu\right)-\left(\partial_\nu A^a_\mu\right)$$

```
Explicit[SUNF[a, b, c]]
```

$$f^{abc}$$

```
Explicit[SUNF[a, b, c], SUNF -> True]
```

$$2i\left(\text{tr}\left(T^a.T^c.T^b\right)-\text{tr}\left(T^a.T^b.T^c\right)\right)$$

```
Explicit[SUND[a, b, c]]
```

$$d^{abc}$$

```
Explicit[SUND[a, b, c], SUND -> True]
```

$$2\ \text{tr}\left(T^a.T^b.T^c\right)+2\ \text{tr}\left(T^b.T^a.T^c\right)$$

3.13 Factor1

`Factor1[poly]` factorizes common terms in the summands of poly. It uses basically **`PolynomialGCD`**.

3.13.1 See also

Overview, Factor2.

3.13.2 Examples

```
(a - x) (b - x)
```

```
{Factor1[%], Factor[%]}
```

$$(a - x)(b - x)$$

$$\{(a - x)(b - x), -((a - x)(x - b))\}$$

```
ex = Expand[(a - b) (a + b)]
```

$$a^2 - b^2$$

```
Factor[ex]
```

$$(a - b)(a + b)$$

```
Factor1[ex]
```

$$a^2 - b^2$$

3.14 Factor2

Factor2[poly] factors a polynomial in a standard way.

Factor2 works sometimes better than Factor on polynomials involving rationals with sums in the denominator.

Factor2 uses **Factor** internally and is in general slower than **Factor**. There are four possible settings of the option **Method** (**0**,**1**,**2**,**3**). In general, **Factor** will work faster than **Factor2**.

3.14.1 See also

Overview, Collect2.

3.14.2 Examples

```
(a - x) (b - x)
```

```
{Factor2[%], Factor[%]}
```

$$(a-x)(b-x)$$

$$\{(a-x)(b-x), -((a-x)(x-b))\}$$

```
ex = Expand[(a - b) (a + b)]
```

$$a^2 - b^2$$

```
Factor[ex]
```

$$(a-b)(a+b)$$

```
Factor2[ex]
```

$$a^2 - b^2$$

```
Factor2[ex, FactorFull -> True]
```

$$(a-b)(a+b)$$

3.15 Factor3

`Factor3[exp]` factors a rational function **`exp`** over the field of complex numbers.

`Factor3` is primarily meant to be used on matrices from differential equations and Feynman parametric representations of loop integrals. Its main goal is to rewrite all denominators such, that they can be integrated in terms of HPLs or GPLs (when possible).

To avoid performance bottlenecks, in the case of rational functions only the denominator will be factored by default. This can be changed by setting the option **`Numerator`** to **`True`**.

3.15.1 See also

Overview, FCIntegratePolyLogs.

3.15.2 Examples

```
Factor3[(1 - 4 x) (1 + 3 y)]
```

$$-12\left(x-\frac{1}{4}\right)\left(y+\frac{1}{3}\right)$$

```
Factor3[16*(1 - 2*eps)^2*x^2]
```

$$64\left(\text{eps}-\frac{1}{2}\right)^2 x^2$$

```
Factor3[2*(32904490323 + 164521613783*eps + 1256744*eps^2)*(11 - 5*eps -
47*eps^2 + 44*eps^3)]
```

$$\begin{aligned}&110593472\left(\text{eps}+\frac{1}{264}\left(1-i\sqrt{3}\right)\sqrt[3]{-137143+198i\sqrt{122615}}\right.\\&\left.+\frac{2869\left(1+i\sqrt{3}\right)}{264\sqrt[3]{-137143+198i\sqrt{122615}}}-\frac{47}{132}\right)\left(\text{eps}+\frac{1}{264}\left(1+i\sqrt{3}\right)\sqrt[3]{-137143+198i\sqrt{122615}}\right.\\&\left.+\frac{2869\left(1-i\sqrt{3}\right)}{264\sqrt[3]{-137143+198i\sqrt{122615}}}-\frac{47}{132}\right)\left(\text{eps}-\frac{628374}{-1570927-\sqrt{2467796558401}}\right)\left(\text{eps}\right.\\&\left.-\frac{104729\left(-1570927-\sqrt{2467796558401}\right)}{2513488}\right)\left(\text{eps}\right.\\&\left.+\frac{1}{132}\left(-47-\frac{2869}{\sqrt[3]{-137143+198i\sqrt{122615}}}-\sqrt[3]{-137143+198i\sqrt{122615}}\right)\right)\end{aligned}$$

```
mat = {{(2 - 2*eps)/x, 0, 0, 0, 0}, {0, (2 - 2*eps)/(2*x), 0, 0, 0},
    {0, (-2 + 2*eps)/(x - 4*x^2), (6 - 2*(4 - 2*eps))/(1 - 4*x), 0, 0},
    {(-2 + 2*eps)/(x - 4*x^2), 0, 0, (2 - 2*eps + 4*(5 - 2*(4 -
    2*eps))*x)/(2*(1 -
          4*x)*x), 0},  {(2 - 2*eps)^2/(16*(1 - x)*x^2), -1/8*(2 -
          2*eps)^2/((1 - x)*x^2),
     0, 0, -((7 - 2*(4 - 2*eps) - 13*x + 4*(4 - 2*eps)*x)/(2*x - 2*x^2))}};
```

```
Factor3[mat]
```

$$\begin{pmatrix} \frac{2-2\,\text{eps}}{x} & 0 & 0 & 0 & 0 \\ 0 & \frac{2-2\,\text{eps}}{2x} & 0 & 0 & 0 \\ 0 & -\frac{2\,\text{eps}-2}{4\left(x-\frac{1}{4}\right)x} & -\frac{6-2(4-2\,\text{eps})}{4\left(x-\frac{1}{4}\right)} & 0 & 0 \\ -\frac{2\,\text{eps}-2}{4\left(x-\frac{1}{4}\right)x} & 0 & 0 & -\frac{4(5-2(4-2\,\text{eps}))x-2\,\text{eps}+2}{8\left(x-\frac{1}{4}\right)x} & 0 \\ -\frac{(2-2\,\text{eps})^2}{16(x-1)x^2} & \frac{(2-2\,\text{eps})^2}{8(x-1)x^2} & 0 & 0 & -\frac{-4(4-2\,\text{eps})x+2(4-2\,\text{eps})+13x-7}{2(x-1)x} \end{pmatrix}$$

3.16 FactorList2

FactorList2[exp] is similar to **FactorList** except that it correctly handles symbolic exponents.

3.16.1 See also

Overview, Factor2.

3.16.2 Examples

```
FactorList2[(x[1] x[2] + x[1] x[3] + x[2] x[3])^(-3 + 3 ep)/(x[1]^2 x[2] +
x[1]^2 x[3])^(-1 + 2 ep)]
```

$$\begin{pmatrix} 1 & 1 \\ x(1)^2(x(2)+x(3)) & -2\,\text{ep} \\ x(1)x(2)+x(3)x(2)+x(1)x(3) & 3(\text{ep}-1) \\ x(1) & 2 \\ x(2)+x(3) & 1 \end{pmatrix}$$

3.17 FC

FC changes the output format to **FeynCalcForm**. To change to **InputForm** use **FI**.

3.17.1 See also

Overview, FeynCalcForm, FI, FeynCalcExternal, FeynCalcInternal.

3.17.2 Examples

```
FI

{DiracGamma[5], DiracGamma[Momentum[p]]}
```

```
{DiracGamma[5], DiracGamma[Momentum[p]]}
```

```
FC

{DiracGamma[5], DiracGamma[Momentum[p]]}
```

$$\{\bar{\gamma}^5, \bar{\gamma} \cdot \bar{p}\}$$

3.18 FCAbbreviate

FCAbbreviate[exp, {q1, q2, ...}, {p1, p2, ...}] introduces abbreivations for scalar products of external momenta, **SMP**-symbols and other variables that are present in the expression. Functions (**LeafCount > 1**) are not supported. The main purpose is to simplify the export of FeynCalc expressions to other software tools that might not provide the richness of Mathematica's syntax. The result is returned as a list of replacement rules for scalar products, **SMP**s and all other variables present.

3.18.1 See also

Overview, ScalarProduct, SMP.

3.18.2 Examples

```
(a + I b)^2

FCAbbreviate[%, {}, {}]
```

$$(a + ib)^2$$

$$\{\{\}, \{\}, \{a \to \text{var1}, b \to \text{var2}\}\}$$

```
SPD[p, k] FAD[{q, SMP["m_e"]}, {q + p, m}]

FCAbbreviate[%, {q}, {p, k}, Head -> spd]
```

$$\frac{k \cdot p}{(q^2 - m_e^2) \cdot ((p+q)^2 - m^2)}$$

$$\{\{\text{spd}(k,k) \to \text{sp1}, \text{spd}(k,p) \to \text{sp2}, \text{spd}(p,p) \to \text{sp3}\}, \{m_e \to \text{sm1}\}, \{m \to \text{var1}\}\}$$

```
FCClearScalarProducts[];

SPD[p1, p1] = 0;

SPD[p2, p2] = 0;

SPD[p3, p3] = 0;

SPD[p1, p2] = s/2; SPD[p1, p3] = -(s + t)/2; SPD[p2, p3] = t/2;

SPD[p2, p3] FAD[q, q - p1 - p2, q - p1 - p2 - p3]

FCAbbreviate[%, {q}, {p1, p2, p3}, Head -> spd]
```

$$\frac{t}{2q^2.(-\text{p1} - \text{p2} + q)^2.(-\text{p1} - \text{p2} - \text{p3} + q)^2}$$

$$\left\{\left\{\text{spd}(\text{p1},\text{p1}) \to 0, \text{spd}(\text{p1},\text{p2}) \to \frac{\text{var1}}{2}, \text{spd}(\text{p1},\text{p3}) \to \frac{1}{2}(-\text{var1} - \text{var2}), \text{spd}(\text{p2},\right.\right.$$
$$\left.\left.\text{p2}) \to 0, \text{spd}(\text{p2},\text{p3}) \to \frac{\text{var2}}{2}, \text{spd}(\text{p3},\text{p3}) \to 0\right\}, \{\}, \{s \to \text{var1}, t \to \text{var2}\}\right\}$$

```
FCClearScalarProducts[]
```

3.19 FCAntiSymmetrize

`FCAntiSymmetrize[expr, {a1, a2, ...}]` antisymmetrizes **`expr`** with respect to the variables **`a1, a2, ...`**.

3.19.1 See also

Overview, FCSymmetrize.

3.19.2 Examples

```
FCAntiSymmetrize[f[a, b], {a, b}]
```

$$\frac{1}{2}(f(a,b)-f(b,a))$$

```
FCAntiSymmetrize[f[x, y, z], {x, y, z}]
```

$$\frac{1}{6}(f(x,y,z)-f(x,z,y)-f(y,x,z)+f(y,z,x)+f(z,x,y)-f(z,y,x))$$

3.20 FCDeclareHeader

FCDeclareHeader is an internal FeynCalc function to declare objects inside an .m file in the same manner as it is done in the JLink package. It may be used by FeynCalc addons.

3.20.1 See also

Overview

3.20.2 Examples

3.21 FCPrint

FCPrint[level, x] outputs **Print[x]** if the value of **$VeryVerbose** is larger than level.

3.21.1 See also

Overview, $VeryVerbose.

3.21.2 Examples

```
FCPrint[1, "Test"]
```

```
FCPrint[0, "Test"]
```

Test

3.22 FCReloadAddOns

FCReloadAddOns[{addons}] is an auxiliary function that attempts to reload FeynCalc addons without restarting the kernel.

It is intended to be a helper tool for FeynCalc developers, which allows one to debug/improve add-ons and test the results on the fly. Depending on the complexity of the given add-on, there might also be severe side effects.

The function is not meant to be invoked by the normal users.

3.22.1 See also

Overview, FCReloadFunctionFromFile

3.22.2 Examples

3.23 FCReloadFunctionFromFile

FCReloadFunctionFromFile[function, path] is an auxiliary function that attempts to remove all the definitions of the given FeynCalc function and then reload them from the specified file.

It is intended to be a helper tool for FeynCalc developers, which allows one to debug/improve internal functions and test the results without restarting the kernel. Depending on the complexity of the given function, there might also be unknown side effects.

The function is not meant to be invoked by the normal users.

3.23.1 See also

Overview, FCReloadAddOns.

3.23.2 Examples

3.24 FCDuplicateFreeQ

FCDuplicateFreeQ[list] yields **True** if list contains no duplicates and **False** otherwise.

FCDuplicateFreeQ[list, test] uses test to determine whether two objects should be considered duplicates.

FCDuplicateFreeQ returns the same results as the standard **DuplicateFreeQ**. The only reason for introducing **FCDuplicateFreeQ** is that **DuplicateFreeQ** is not available in Mathematica 8 and 9, which are still supported by FeynCalc.

3.24.1 See also

Overview, FCSubsetQ.

3.24.2 Examples

```
FCDuplicateFreeQ[{a, b, c}]
```

True

```
FCDuplicateFreeQ[{a, b, c, a}]
```

False

```
FCDuplicateFreeQ[{{a, b}, {a, c}}]
```

True

```
FCDuplicateFreeQ[{{a, b}, {a, c}}, Function[{x, y}, First[x] === First[y]]]
```

False

3.25 FCClearCache

`FCClearCache[func]` removes existing cached values for the function **`func`** that were introduced by **`FCUseCache`**.

To remove all existing cache values use **`FCClearCache[All]`**.

3.25.1 See also

Overview, FCUseCache, FCShowCache.

3.25.2 Examples

3.26 FCMemoryAvailable

`FCMemoryAvailable` is an option of **`MemSet`**. It can be set to an integer **`n`**, where **`n`** is the available amount of main memory in MiB. The default setting is **`$FCMemoryAvailable`**.

3.26.1 See also

Overview, MemSet, $FCMemoryAvailable.

3.26.2 Examples

3.27 FCShowCache

FCShowCache[func] shows existing cached values for the function **func**, that were introduced by **FCUseCache**.

3.27.1 See also

Overview, FCUseCache.

3.27.2 Examples

3.28 FCUseCache

FCUseCache[func, {arg1, ...}, {opt1...}] evaluates **func[arg1, ..., opt1, ...]** and caches the result such that the next evaluation of same expressions occurs almost immediately. This caching also takes into account **DownValues** and global variables that enter into evaluation of func.

For example, **ExpandScalarProduct** can't be naively cached, because its result depends on the **DownValues** of **Pair** and **ScalarProduct**, which may be changed multiple times during the session by setting and erasing values of scalar products. With **FCUseCache**, however, caching will work properly, as **FCUseCache** knows the dependence on **ExpandScalarProduct** on those DownValues. For all this to work, a function should be explicitly white-listed in FCUseCache.

3.28.1 See also

Overview, FCShowCache.

3.28.2 Examples

3.29 FCCheckSyntax

FCCheckSyntax[exp] attempts to detect mistakes and inconsistencies in the user input. The function returns the original expression but will abort the evaluation if it thinks that the input is incorrect. Notice that false positives are possible and it is not guaranteed that the input which passes **FCCheckSyntax** is indeed fully correct.

FCCheckSyntax is also an option for several FeynCalc routines. If set to **True**, those functions will try to check the syntax of the input expressions to detect possible inconsistencies. However, on large expressions such checks may cost a lot of performance, which is why this option is set to **False** by default.

3.29.1 See also

Overview

3.29.2 Examples

Typical mistake, using **Times** instead of **Dot** in noncommutative products

```
FCCheckSyntax[GA[mu]*GA[nu]]
```

FCCheckSyntax: Error! FCCheckSyntax has found an inconsistency in your input expression and must abort the evaluation. The problem reads: Commutative products of DiracGamma in $\bar{\gamma}^{mu}\bar{\gamma}^{nu}$

$Aborted

Another common mistake, Einstein summation convention is violated

```
FCCheckSyntax[FV[p, \[Mu]] FV[q, \[Mu]] FV[r, \[Mu]]]
```

FCCheckSyntax: Error! FCCheckSyntax has found an inconsistency in your input expression and must abort the evaluation. The problem reads: More than two repeating indices in Pair[LorentzIndex[μ], Momentum[p]]*Pair[LorentzIndex[μ], Momentum[q]]*Pair[LorentzIndex[μ], Momentum[r]]

$Aborted

3.30 FCCheckVersion

FCCheckVersion[major, minor, build] checks if the current version of FeynCalc is larger or equal than **marjor.minor.build**. For example, **FCCheckVersion[9, 3, 0]** will generate a warning (when running with the frontend) or quit kernel (when running without the frontend) if the loaded FeynCalc version is older than 9.3.0.

Notice that this function is available only since FeynCalc 9.3.

3.30.1 See also

Overview, $FeynCalcVersion.

3.30.2 Examples

```
FCCheckVersion[8, 2, 0]
```

```
(*FCCheckVersion[15,2,0]*)
```

3.31 FCCompareResults

FCCompareResults[{res1, res2, ...}, {res1Known, res2Known, ...}] compares the given list of expression **{res1, res2, ...}** to the list of expressions **{res1Known, res2Known, ...}** that represents the correct results. This is handy for checking both intermediate and final results of calculations, where you know what should come out at the end.

3.31.1 See also

Overview, FCMatchSolve.

3.31.2 Examples

```
FCCompareResults[{4, 4}, {2^2, 8/2}]
```

Check of the results: The results agree.

True

```
FCCompareResults[{3, 5}, {2^2, 8/2}]
```

Check of the results: The results disagree.

False

3.32 FCCompareNumbers

FCCompareNumbers[x, y] compares two purely numerical or semi-numerical expressions **x** and **y** and returns the number of agreeing significant digits calculated from the relative differences.

3.32.1 See also

Overview

3.32.2 Examples

These two numbers disagree in their 6th significant digit

```
FCCompareNumbers[5.123542342 + 1.23145 I, 5.123542324 + 1.23146 I]
```

FCCompareNumbers: Minimal number of significant digits to agree in: 6

FCCompareNumbers: Chop is set to 1.*^-10

FCCompareNumbers: No number is set to 0. by Chop at this stage.

FCGV(I) FCGV(CommonDigits)(5.09042)

Requiring agreement only in the first 5 significant digits returns 0

```
FCCompareNumbers[5.123542342 + 1.23145 I, 5.123542324 + 1.23146 I,
DigitCount -> 5]
```

FCCompareNumbers: Minimal number of significant digits to agree in: 5

FCCompareNumbers: Chop is set to 1.*^-10

FCCompareNumbers: No number is set to 0. by Chop at this stage.

0

Here even the first significant digit doesn't agree (obviously)

```
FCCompareNumbers[5, 6]
```

FCCompareNumbers: Minimal number of significant digits to agree in: 6

FCCompareNumbers: Chop is set to 1.*^-10

FCCompareNumbers: No number is set to 0. by Chop at this stage.

FCGV(CommonDigits)(0.778151)

```
lhs = (0. + 0.*I) - (0.20132103165327941 -
0.0004343444331339924 6*I)*coeff0^2*parX^2 +
(0.047227066764317975)*coeff0^2*parX^2*parY -
(0.00005403882927314103)*coeff0^2*parX^2*parY^2 + (1.4588597782189382*^-6 -
4.06569606476957*^-13*I)*coeff0^2*parX^2*parY^3 + (0.03841797609570242 +
0.00002840373351615344 6*I)*coeff0^2*parX^2*parZ
```

$$(0.038418 + 0.0000284037i)\ \text{coeffO}^2\ \text{parX}^2\ \text{parZ} - (0.201321 - 0.000434344i)\ \text{coeffO}^2\ \text{parX}^2 + (0. + 0.i) + (1.4588597782189382\text{*}\text{\^{}}\text{-}6 - 4.06569606476957\text{*}\text{\^{}}\text{-}13i)\ \text{coeffO}^2\ \text{parX}^2\ \text{parY}^3 - 0.0000540388\ \text{coeffO}^2\ \text{parX}^2\ \text{parY}^2 + 0.0472271\ \text{coeffO}^2\ \text{parX}^2\ \text{parY}$$

```
rhs = (-0.20132103165327922 + 0.0004343444331339952*I)*coeff0^2*parX^2 +
(0.0472270672349811)*coeff0^2*parX^2*parY -
(0.00005403887000187252)*coeff0^2*parX^2*parY^2 +
1.4588601127764193*^-6*coeff0^2*parX^2*parY^3 + (0.038417976095702376 +
0.00002840373351615353 7*I)*coeff0^2*parX^2*parZ
```

$$(0.038418 + 0.0000284037i)\ \text{coeffO}^2\ \text{parX}^2\ \text{parZ} - (0.201321 - 0.000434344i)\ \text{coeffO}^2\ \text{parX}^2 + 1.4588601127764193\text{*}\text{\^{}}\text{-}6\ \text{coeffO}^2\ \text{parX}^2\ \text{parY}^3 - 0.0000540389\ \text{coeffO}^2\ \text{parX}^2\ \text{parY}^2 + 0.0472271\ \text{coeffO}^2\ \text{parX}^2\ \text{parY}$$

Here the two above expressions agree in their first 6 significant digits. Notice that the number of the size 10^{-13} is treated as a numerical fluctuation and consequently removed by **Chop**

```
FCCompareNumbers[lhs, rhs]
```

FCCompareNumbers: Minimal number of significant digits to agree in: 6

FCCompareNumbers: Chop is set to 1.*^-10

FCCompareNumbers: Following numbers on the l.h.s. are set to 0. by Chop: {−4.06569606476957*^-13}

0

The application of **Chop** can be of course disabled

```
FCCompareNumbers[lhs, rhs, Chop -> False]
```

FCCompareNumbers: Minimal number of significant digits to agree in: 6

$$\text{coeffO}^2\ \text{parX}^2\ \text{parY}^3\ \text{FCGV(I)}\ \text{FCGV(Unmatched)}(-4.06569606476957*\text{\^{}}\text{-}13)$$

3.33 FCDisableTraditionalFormOutput

FCDisableTraditionalFormOutput[] sets the output format of the current FrontEnd to **StandardForm**. The setting is not persistent, such that it does not influence any subsequent Mathematica FrontEnd sessions.

3.33.1 See also

Overview, $FCTraditionalFormOutput, FCDisableTraditionalFormOutput.

3.33.2 Examples

```
FCDisableTraditionalFormOutput[]

FV[p, \[Mu]]
```

$$\overline{p}^{\mu}$$

```
FCEnableTraditionalFormOutput[]

FV[p, \[Mu]]
```

$$\overline{p}^{\mu}$$

3.34 FCEnableTraditionalFormOutput

FCEnableTraditionalFormOutput[] sets the output format of the current FrontEnd to **TraditionalForm**. The setting is not persistent, such that it does not influence any subsequent Mathematica FrontEnd sessions.

3.34.1 See also

Overview, $FCTraditionalFormOutput, FCDisableTraditionalFormOutput.

3.34.2 Examples

```
FCDisableTraditionalFormOutput[]

FV[p, \[Mu]]
```

$$\overline{p}^{\mu}$$

```
FCEnableTraditionalFormOutput[]

FV[p, \[Mu]]
```

$$\overline{p}^{\mu}$$

3.35 FCFactorOut

FCFactorOut[exp, pref] factors out **pref** out of **exp**. This is often needed to bring **exp** into a particular form that Mathematica refuses to give.

3.35.1 See also

Overview, Collect2.

3.35.2 Examples

```
FCFactorOut[(a + 3 b), 3 b]
```

$$3b\left(\frac{a}{3b}+1\right)$$

```
FCFactorOut[(a + 3 b), 3 b, Head -> hold]
```

$$3b\ \text{hold}\left(\frac{a}{3b}+1\right)$$

FCFactorOut is also an option of **Collect2**

```
x^2 + 6 y

Collect2[%, {x, y}, FCFactorOut -> 3]
```

$$x^2 + 6y$$

$$3\left(\frac{x^2}{3} + 2y\right)$$

3.36 FCFilePatch

FCFilePatch[input, output, rp] replaces the patterns given by **rp** in the file **input** and writes the modified version to **output**. Here **rp** should be a list of the form **{string -> string, ...}**.

3.36.1 See also

Overview, FAPatch.

3.36.2 Examples

3.37 FCGetNotebookDirectory

FCGetNotebookDirectory[] is a convenience function that returns the directory in which the current notebook or .m file is located. It also works when the FrontEnd is not available.

3.37.1 See also

Overview

3.37.2 Examples

```
FCGetNotebookDirectory[]
```

/media/Data/Projects/VS/FeynCalc/FeynCalc/Documentation/Mathematica/Shared/Tools/

3.38 FCHighlight

FCHighlight[exp, {{symbol1, color1}, {symbol2, color2}, ...}] highlights the given set of symbols in the output using **Style** and the provided colors. This works only in the frontend and alters the input expression in such a way, that it cannot be processed further (because of the introduced **Style** heads).

3.38.1 See also

Overview

3.38.2 Examples

```
FCHighlight[Expand[(a + b + c)^2], {{a, Red}, {b, Blue}}]
```

$$2ab + 2ac + a^2 + 2bc + b^2 + c^2$$

3.39 FCE

FCE[exp] translates **exp** from the internal FeynCalc representation to a short form.

FCE is equivalent to **FeynCalcExternal**.

3.39.1 See also

Overview, FeynCalcExternal, FCI, FeynCalcInternal.

3.39.2 Examples

```
ex = FCE[{DiracGamma[5], DiracGamma[Momentum[p]]}]
```

$$\{\bar{\gamma}^5, \bar{\gamma} \cdot \bar{p}\}$$

```
ex // StandardForm
```

```
(*{GA[5], GS[p]}*)
```

```
ex = {GA[\[Mu]], GAD[\[Rho]], GS[p], SP[p, q], MT[\[Alpha], \[Beta]], FV[p,
\[Mu]]}
```

$$\left\{\bar{\gamma}^{\mu}, \gamma^{\rho}, \bar{\gamma} \cdot \overline{p}, \overline{p} \cdot \overline{q}, \bar{g}^{\alpha\beta}, \overline{p}^{\mu}\right\}$$

```
ex // StandardForm

(*{GA[\[Mu]], GAD[\[Rho]], GS[p], SP[p, q], MT[\[Alpha], \[Beta]], FV[p,
\[Mu]]}*)
```

```
ex // FCI // StandardForm

(*{DiracGamma[LorentzIndex[\[Mu]]], DiracGamma[LorentzIndex[\[Rho], D], D],
DiracGamma[Momentum[p]], Pair[Momentum[p], Momentum[q]],
Pair[LorentzIndex[\[Alpha]], LorentzIndex[\[Beta]]],
Pair[LorentzIndex[\[Mu]], Momentum[p]]}*)
```

```
FCE[ex] // StandardForm

(*{GA[\[Mu]], GAD[\[Rho]], GS[p], SP[p, q], MT[\[Alpha], \[Beta]], FV[p,
\[Mu]]}*)
```

3.40 FeynCalcExternal

`FeynCalcExternal[exp]` translates exp from the internal FeynCalc representation to a shorthand form.

3.40.1 See also

Overview, FeynCalcInternal.

3.40.2 Examples

```
FeynCalcExternal[DiracGamma[5]]
```

$$\bar{\gamma}^5$$

```
FeynCalcExternal[DiracGamma[5]] // StandardForm

(*GA[5]*)
```

```
ex = {GA[\[Mu]], GAD[\[Rho]], GS[p], SP[p, q], MT[\[Alpha], \[Beta]], FV[p,
\[Mu]]}
```

$$\{\bar{\gamma}^{\mu}, \gamma^{\rho}, \bar{\gamma} \cdot \overline{p}, \overline{p} \cdot \overline{q}, \bar{g}^{\alpha\beta}, \overline{p}^{\mu}\}$$

```
ex // StandardForm

(*{GA[\[Mu]], GAD[\[Rho]], GS[p], SP[p, q], MT[\[Alpha], \[Beta]], FV[p,
\[Mu]]}*)
```

```
ex // FeynCalcInternal
```

$$\{\bar{\gamma}^{\mu}, \gamma^{\rho}, \bar{\gamma} \cdot \overline{p}, \overline{p} \cdot \overline{q}, \bar{g}^{\alpha\beta}, \overline{p}^{\mu}\}$$

```
ex // FeynCalcInternal // StandardForm

(*{DiracGamma[LorentzIndex[\[Mu]]], DiracGamma[LorentzIndex[\[Rho], D], D],
DiracGamma[Momentum[p]], Pair[Momentum[p], Momentum[q]],
Pair[LorentzIndex[\[Alpha]], LorentzIndex[\[Beta]]],
Pair[LorentzIndex[\[Mu]], Momentum[p]]}*)
```

```
ex // FeynCalcInternal // FeynCalcExternal // StandardForm

(*{GA[\[Mu]], GAD[\[Rho]], GS[p], SP[p, q], MT[\[Alpha], \[Beta]], FV[p,
\[Mu]]}*)
```

3.41 FCF

FCF[exp] is a short form for FeynCalcForm[exp].

3.41.1 See also

Overview, FeynCalcForm.

3.41.2 Examples

3.42 FeynCalcForm

FeynCalcForm[expr] changes the printed output to a an easy-to-read form. It allows a readable output also when running a terminal based Mathematica session. Whether the result of **FeynCalcForm[expr]** is displayed or not, depends on the setting of **$PrePrint**.

$PrePrint = FeynCalcForm forces displaying everything after applying **FeynCalcForm**. In order to change to the normal (internal) Mathematica OutputForm, do: **$PrePrint=.**.

3.42.1 See also

Overview, FC, FeynCalcExternal, FeynCalcInternal.

3.42.2 Examples

This is the normal notebook display:

```
SUNTrace[SUNT[a] . SUNT[b] . SUNT[c]]
```

$$\text{tr}(T^a.T^b.T^c)$$

This is the shorthand (terminal) display (easy-to-read form):

```
$PrePrint = FeynCalcForm;

SetOptions[$FrontEndSession, Evaluate[(Options[$FrontEndSession,
"CommonDefaultFormatTypes"] /. ("Output" -> _) -> ("Output"
->OutputForm))[[1]]]];

SUNTrace[SUNT[a] . SUNT[b] . SUNT[c]]
```

$$\text{tr}(T^a.T^b.T^c)$$

Reset to normal notebook display:

```
$PrePrint =.;

SetOptions[$FrontEndSession, Evaluate[(Options[$FrontEndSession,
"CommonDefaultFormatTypes"] /. ("Output" -> _) -> ("Output"
->TraditionalForm))[[1]]]];

SUNTrace[SUNT[a] . SUNT[b] . SUNT[c]]
```

$$\text{tr}(T^a.T^b.T^c)$$

3.43 FCI

FCI[exp] translates exp into the internal FeynCalc (datatype-)representation.

FCI is equivalent to **FeynCalcInternal**.

3.43.1 See also

Overview, FeynCalcExternal, FeynCalcInternal, FCE.

3.43.2 Examples

```
ex = {GA[\[Mu]], GAD[\[Rho]], GS[p], SP[p, q], MT[\[Alpha], \[Beta]], FV[p,
\[Mu]]}
```

$$\left\{\bar{\gamma}^{\mu}, \gamma^{\rho}, \bar{\gamma}\cdot\overline{p}, \overline{p}\cdot\overline{q}, \bar{g}^{\alpha\beta}, \overline{p}^{\mu}\right\}$$

```
ex // StandardForm

(*{GA[\[Mu]], GAD[\[Rho]], GS[p], SP[p, q], MT[\[Alpha], \[Beta]], FV[p,
\[Mu]]}*)
```

```
ex // FCI
```

$$\left\{\bar{\gamma}^{\mu}, \gamma^{\rho}, \bar{\gamma}\cdot\overline{p}, \overline{p}\cdot\overline{q}, \bar{g}^{\alpha\beta}, \overline{p}^{\mu}\right\}$$

```
ex // FCI // StandardForm

(*{DiracGamma[LorentzIndex[\[Mu]]], DiracGamma[LorentzIndex[\[Rho], D], D],
DiracGamma[Momentum[p]], Pair[Momentum[p], Momentum[q]],
Pair[LorentzIndex[\[Alpha]], LorentzIndex[\[Beta]]],
Pair[LorentzIndex[\[Mu]], Momentum[p]]}*)
```

```
ex // FCE
```

$$\left\{\bar{\gamma}^{\mu}, \gamma^{\rho}, \bar{\gamma}\cdot\overline{p}, \overline{p}\cdot\overline{q}, \bar{g}^{\alpha\beta}, \overline{p}^{\mu}\right\}$$

```
ex // FCE // StandardForm

(*{GA[\[Mu]], GAD[\[Rho]], GS[p], SP[p, q], MT[\[Alpha], \[Beta]], FV[p,
\[Mu]]}*)
```

3.44 FeynCalcInternal

`FeynCalcInternal[exp]` translates **`exp`** into the internal FeynCalc (abstract data-type) representation.

3.44.1 See also

Overview, FeynCalcExternal, FCI, FCE.

3.44.2 Examples

```
ex = {GA[\[Mu]], GAD[\[Rho]], GS[p], SP[p, q], MT[\[Alpha], \[Beta]], FV[p,
\[Mu]]}
```

$$\left\{\bar{\gamma}^{\mu}, \gamma^{\rho}, \bar{\gamma}\cdot\overline{p}, \overline{p}\cdot\overline{q}, \bar{g}^{\alpha\beta}, \overline{p}^{\mu}\right\}$$

```
ex // StandardForm

(*{GA[\[Mu]], GAD[\[Rho]], GS[p], SP[p, q], MT[\[Alpha], \[Beta]], FV[p,
\[Mu]]}*)
```

```
ex // FeynCalcInternal
```

$$\left\{\bar{\gamma}^{\mu}, \gamma^{\rho}, \bar{\gamma}\cdot\overline{p}, \overline{p}\cdot\overline{q}, \bar{g}^{\alpha\beta}, \overline{p}^{\mu}\right\}$$

```
ex // StandardForm

(*{GA[\[Mu]], GAD[\[Rho]], GS[p], SP[p, q], MT[\[Alpha], \[Beta]], FV[p,
\[Mu]]}*)
```

```
FeynCalcExternal[ex] // StandardForm

(*{GA[\[Mu]], GAD[\[Rho]], GS[p], SP[p, q], MT[\[Alpha], \[Beta]], FV[p,
\[Mu]]}*)
```

```
ex = FCI[{SD[a, b], SUND[a, b, c], SUNF[a, b, c], FAD[q], LC[\[Mu], \[Nu],
\[Rho], \[Sigma]]}]
```

$$\left\{\delta^{ab}, d^{abc}, f^{abc}, \frac{1}{q^2}, \bar{\epsilon}^{\mu\nu\rho\sigma}\right\}$$

```
ex // StandardForm

(*{SUNDelta[SUNIndex[a], SUNIndex[b]], SUND[SUNIndex[a], SUNIndex[b],
SUNIndex[c]], SUNF[SUNIndex[a], SUNIndex[b], SUNIndex[c]],
FeynAmpDenominator[PropagatorDenominator[Momentum[q, D], 0]],
Eps[LorentzIndex[\[Mu]], LorentzIndex[\[Nu]], LorentzIndex[\[Rho]],
LorentzIndex[\[Sigma]]]}*)
```

3.45 FCMakeIndex

FCMakeIndex[str1, str2, head] generates an index with the given head out of the string **str1** and **str2**. For example, **FCMakeIndex["Lor", "1", LorentzIndex]** yields **LorentzIndex[Lor1]**. The second argument can also be an integer. **FCMakeIndex** is useful for converting the output of different diagram generators such as FeynArts or QGAF into the FeynCalc notation. It uses memoization to improve the performance.

3.45.1 See also

Overview, FCMakeSymbols.

3.45.2 Examples

```
FCMakeIndex["Lor", "1"]
```

Lor1

```
FCMakeIndex["Lor", "1"] // StandardForm

(*Lor1*)
```

```
FCMakeIndex["Lor", {3, 1, 4}, LorentzIndex]
```

{Lor3, Lor1, Lor4}

```
FCMakeIndex["Lor", {3, 1, 4}, LorentzIndex] // StandardForm

(*{LorentzIndex[Lor3], LorentzIndex[Lor1], LorentzIndex[Lor4]}*)
```

```
FCMakeIndex["Sun", {"a", 1, -4}]
```

{Suna, Sun1, SunMinus4}

```
FCMakeIndex["Sun", {"a", 1, -4}] // StandardForm

(*{Suna, Sun1, SunMinus4}*)
```

3.46 FCMakeSymbols

FCMakeSymbols[name, range, type] generates a list or a sequence of symbols (depending on the value of type) by attaching elements of the list range to name.

For example, **FCMakeSymbols[mu, Range[1, 3], List]** returns **{mu1, mu2, mu3}**.

3.46.1 See also

Overview, FCMakeIndex.

3.46.2 Examples

```
FCMakeSymbols[a, Range[1, 4], List]
```

$$\{\text{a1}, \text{a2}, \text{a3}, \text{a4}\}$$

```
f[FCMakeSymbols[a, Range[1, 4], Sequence]]
```

$$f(\text{a1}, \text{a2}, \text{a3}, \text{a4})$$

```
f[FCMakeSymbols[a, {1, 3}, Sequence]]
```

$$f(\text{a1}, \text{a3})$$

3.47 FCMatchSolve

FCMatchSolve[expr, {notvar1, notvar2, ...}] assumes that **expr** is a sum that must vanish term-wise and converts it to a system of linear equations. The function automatically determines which variables to solve for, excluding **notvar1, notvar2, ...** from the list.

FCMatchSolve can also handle overdetermined systems of equations. This function is useful e.g. for determining renormalization constants or matching coefficients, where looking at each term separately and determining the values of the constants/coefficients by hand is too tedious.

The input (say a sum or a difference of amplitudes) should be prepared using **Collect2** by collecting w.r.t distinct objects, e.g. matrix elements or coupling constants so that each term must vanish separately.

3.47.1 See also

Overview, Collect2, Solve2, Solve3.

3.47.2 Examples

```
FCMatchSolve[-1/8*(tauPref*(-128 + 64*nc + 160*nc^2 - 8*nc*zz14 + 4*nc*zz44
-
         16*evFlag[4, 3, 1] + 8*nc*evFlag[4, 3, 1] - 16*evFlag[5, 3, 1] +
         8*nc*evFlag[5, 3, 1] +
         nc*evFlag[9, 3, 1] - 2*evFlag[10, 3, 1] + nc^2*evFlag[10, 3,
1])*OP[Q])/nc +
   (tauPref*(-96*nc - 96*nc^2 + 4*nc*zz24 - 4*nc*zz44 - 16*evFlag[4, 3, 1]
+
        24*nc*evFlag[4, 3, 1] + 16*nc^2*evFlag[4, 3, 1] + 16*evFlag[5, 3,
        1] -
        8*nc*evFlag[5, 3, 1] - nc*evFlag[9, 3, 1] + nc*evFlag[9, 4, 1] +
        2*evFlag[10, 3, 1] -
        nc^2*evFlag[10, 3, 1] - 2*evFlag[10, 4, 1] + nc^2*evFlag[10, 4,
1])*OP[QS])/(4*nc),
  {OP[_], nc, evFlag[__], tauPref}]
```

FCMatchSolve: Solving for: $\{zz14, zz24, zz44\}$

FCMatchSolve: A solution exists.

$$\Big\{zz24 \to \frac{1}{4\,nc}\left(-16\,nc^2\,evFlag(4,3,1) - nc^2\,evFlag(10,4,1) - 32\,nc\,evFlag(4,3,1) - nc\,evFlag(9,4,1) + 32\,evFlag(4,3,1) + 2\,evFlag(10,4,1) - 64\,nc^2 + 8\,nc\,zz14 + 32\,nc + 128\right),\ zz44 \to \frac{1}{4\,nc}\left(nc^2(-evFlag(10,3,1)) - 8\,nc\,evFlag(4,3,1) - 8\,nc\,evFlag(5,3,1) - nc\,evFlag(9,3,1) + 16\,evFlag(4,3,1) + 16\,evFlag(5,3,1) + 2\,evFlag(10,3,1) - 160\,nc^2 + 8\,nc\,zz14 - 64\,nc + 128\right)\Big\}$$

3.48 FCPatternFreeQ

FCPatternFreeQ[{exp}] yields **True** if **{exp}** does not contain any pattern objects, e.g. **Pattern**, **Blank**, **BlankSequence** and **BlankNullSequence**.

FCPatternFreeQ[{exp}, {h1, h2, ...}] checks that in addition to the pattern objects, no heads **h1, h2, ...** are present.

3.48.1 See also

Overview, FreeQ2.

3.48.2 Examples

```
FCPatternFreeQ[{a}]
```

True

```
FCPatternFreeQ[{a_}]
```

False

```
FCPatternFreeQ[{g[x]}, {g}]
```

False

3.49 FCProgressBar

FCProgressBar[text, i, total] is a simple auxiliary function that can be used to display the progress of a certain evaluation, e.g. mapping a list of integrals to some function. Here i is the number of the current step while total denotes the overall number of steps.

3.49.1 See also

Overview

3.49.2 Examples

A simple usage example

```
Table[FCProgressBar["Calculating integral ", i, 10], {i, 1, 10}];
```

Calculating integral 1 / 10 Calculating integral 2 / 10 Calculating integral 3 / 10 Calculating integral 4 / 10 Calculating integral 5 / 10 Calculating integral 6 / 10 Calculating integral 7 / 10 Calculating integral 8 / 10 Calculating integral 9 / 10 Calculating integral 10 / 10

3.50 FCReplaceAll

FCReplaceAll[exp, ru1, ...] is like **ReplaceAll**, but it also allows to apply multiple replacement rules sequentially. Instead of doing **exp /. ru1 /. ru2 /. ru3** one can just write **FCReplaceAll[exp, ru1, ru2, ru3]**.

3.50.1 See also

Overview, FCReplaceRepeated.

3.50.2 Examples

```
FCReplaceAll[a, a -> b]
```

$$b$$

```
FCReplaceAll[a c, {a -> b, c -> d}]
```

$$bd$$

```
FCReplaceAll[a c, a -> b, c -> d, d -> e, b -> f]
```

$$ef$$

3.51 FCReorderList

`FCReorderList[li, ord]` reorders the list **`li`** according to the given ordering **`ord`**.

3.51.1 See also

Overview

3.51.2 Examples

```
myList = Table[mat[i], {i, 1, 23}]
```

$$\{\text{mat}(1), \text{mat}(2), \text{mat}(3), \text{mat}(4), \text{mat}(5), \text{mat}(6), \text{mat}(7), \text{mat}(8), \text{mat}(9), \text{mat}(10), \text{mat}(11), \text{mat}(12), \text{mat}(13), \text{mat}(14), \text{mat}(15), \text{mat}(16), \text{mat}(17), \text{mat}(18), \text{mat}(19), \text{mat}(20), \text{mat}(21), \text{mat}(22), \text{mat}(23)\}$$

```
FCReorderList[myList, {{1, 10}, 23, {11, 20}, 22, 21}]
```

$$\{\text{mat}(1), \text{mat}(2), \text{mat}(3), \text{mat}(4), \text{mat}(5), \text{mat}(6), \text{mat}(7), \text{mat}(8), \text{mat}(9), \text{mat}(10), \text{mat}(23), \text{mat}(11), \text{mat}(12), \text{mat}(13), \text{mat}(14), \text{mat}(15), \text{mat}(16), \text{mat}(17), \text{mat}(18), \text{mat}(19), \text{mat}(20), \text{mat}(22), \text{mat}(21)\}$$

3.52 FCReplaceRepeated

FCReplaceRepeated[exp, ru1, ...] is like **ReplaceRepeated**, but it also allows to apply multiple replacement rules sequentially.

Instead of doing **exp //. ru1 //. ru2 //. ru3** one can just write **FCReplaceRepeated[exp, ru1, ru2, ru3]**.

3.52.1 See also

Overview, FCReplaceAll.

3.52.2 Examples

```
FCReplaceRepeated[a, a -> b]
```

$$b$$

```
FCReplaceRepeated[a c, {a -> b, c -> d}]
```

$$bd$$

```
FCReplaceRepeated[a c, a -> b, c -> d]
```

$$bd$$

```
FCReplaceRepeated[a c, a -> b, c -> d, d -> e, b -> f]
```

$$ef$$

3.53 FCShowReferenceCard

FCShowReferenceCard[{name}] shows the reference card that corresponds to **"name"**. Reference cards are stored in Tables/ReferenceCards inside the FeynCalc main directory. **FCShowReferenceCard[]** lists available reference cards.

3.53.1 See also

Overview

3.53.2 Examples

```
FCShowReferenceCard[]
```

(FeynArts)

```
FCShowReferenceCard[{"FeynArts"}]
```

class	self - conj.	indices	members	mass
F[1] (neutrinos)	no	Generation	F[1, {1}] ν_e F[1, {2}] ν_μ F[1, {3}] ν_τ	0 0 0
F[2] (massive leptons)	no	Generation	F[2, {1}] e F[2, {2}] μ F[2, {3}] τ	ME MM ML
F[3] (up- type quarks)	no	Generation Color	F[3, {1, o}] u F[3, {2, o}] c F[3, {3, o}] t	MU MC MT
F[4] (down- type quarks)	no	Generation Color	F[4, {1, o}] d F[4, {2, o}] s F[4, {3, o}] b	MD MS MB
V[1] V[2] V[3] V[4] (mixing field)	yes yes no yes		V[1] γ V[2] Z V[3] W^- V[4] γ- Z	0 MZ MW MAZ
S[1] S[2] S[3]	yes yes no		S[1] H S[2] G^0 S[3] G^-	MH MG0 MGp
U[1] U[2] U[3] U[4]	no no no no		U[1] u_γ U[2] u_Z U[3] u_- U[4] u_+	0 MZ MW MW
SV[2] (mixing field) SV[3] (mixing field)	yes no		SV[2] G^0- Z SV[3] G^- - W^-	MZ MW
The following fields are avaialble via the SMQCD extension:				
V[5] U[5]	yes no	Gluon Gluon	V[5, {i}] g_i U[5, {i}] u_{g_i}	0 0
Comments: V[4] is commented out by default in SM.mod; SV[2] and SV[3] must be enabled with $SVMixing = True.				

3.54 FCSplit

FCSplit[exp, {v1, v2, ...}] splits expr into pieces that are free of any occurrence of **v1, v2, ...** and pieces that contain those variables. This works both on sums and products. The output is provided in the form of a two element list. One can recover the original expression by applying **Total** to that list.

3.54.1 See also

Overview, FCProductSplit.

3.54.2 Examples

```
FCSplit[(a + b)^2, {a}]
```

$$\{b^2, a^2 + 2ab\}$$

```
FCSplit[(a + b + c)^2, {a, b}]
```

$$\{c^2, a^2 + 2ab + 2ac + b^2 + 2bc\}$$

3.55 FCProductSplit

FCProductSplit[exp, {v1, v2, ...}] splits expr into pieces that are free of any occurrence of **v1, v2, ...** and pieces that contain those variables. This works both only for products. The output is provided in the form of a two element list. One can recover the original expression by applying **Times** to that list.

3.55.1 See also

Overview, FCSplit.

3.55.2 Examples

```
FCProductSplit[c^2, {a}]
```

$$\{c^2, 1\}$$

```
FCProductSplit[a^2*b, {a}]
```

$$\{b, a^2\}$$

```
FCProductSplit[(a^2 + b)*b*(c + d), {a, c}]
```

$$\left\{b, \left(a^2 + b\right)(c + d)\right\}$$

3.56 PartitHead

PartitHead[expr, h] returns a list **{ex1, h[ex2]}** with **ex1** free of expressions with head **h**, and **h[ex2]** having head **h**.

3.56.1 See also

Overview, FCSplit.

3.56.2 Examples

3.57 SelectFree

SelectFree[expr, a, b, ...] is equivalent to **Select[expr, FreeQ2[#, {a, b, ...}]&]**, except the special cases: **SelectFree[a, b]** returns **a** and **SelectFree[a, a]** returns 1 (where **a** is not a product or a sum).

3.57.1 See also

Overview, FreeQ2, SelectNotFree.

3.57.2 Examples

```
SelectFree[a + b + f[a] + d, a]
```

$$b + d$$

```
SelectFree[x y, x]
```

$$y$$

```
SelectFree[2 x y z f[x], {x, y}]
```

$$2z$$

```
SelectFree[a, b]
```

$$a$$

```
SelectFree[a, a]
```

$$1$$

```
SelectFree[1, c]
```

$$1$$

```
SelectFree[f[x], x]
```

$$1$$

3.58 SelectNotFree

SelectNotFree[expr, x] returns that part of **expr** which is not free of any occurrence of **x**.

SelectNotFree[expr, a, b, ...] is equivalent to **Select[expr, !FreeQ2[#, {a, b, ...}]&]**, except the special cases: **SelectNotFree[a, b]** returns **1** and **SelectNotFree[a, a]** returns **a** (where **a** is not a product or a sum).

3.58.1 See also

Overview, FreeQ2, SelectFree.

3.58.2 Examples

```
SelectNotFree[a + b + f[a], a]
```

$$f(a) + a$$

```
SelectNotFree[2 x y f[x] z, {x, y}]
```

$$xyf(x)$$

```
SelectNotFree[a, b]
```

$$1$$

```
SelectNotFree[a + x, b]
```

$$0$$

```
SelectNotFree[a, a]
```

$$a$$

```
SelectNotFree[1, c]
```

$$1$$

```
SelectNotFree[f[x], x]
```

$$f(x)$$

3.59 SelectFree2

`SelectFree2[expr, a, b, ...]` is similar to **`SelectFree`** but it also differs from the latter in several respects.

If **`expr`** is a list, **`SelectFree2`** behaves exactly the same way as **`SelectFree`**.

If **`expr`** is not a list, **`SelectFree2`** first expands the expression w.r.t. the arguments via **`Expand2`**.

Furthermore, **`SelectFree2[a, b]`** returns **`a`** and **`SelectFree2[a, a]`** returns **`0`**. This differs from the behavior of **`SelectFree`** but is consistent with the naive expectations when applying the function to a sum of terms.

3.59.1 See also

Overview, FreeQ2, SelectFree, SelectNotFree, SelectNotFree2.

3.59.2 Examples

Note the difference between SelectFree and SelectFree2

```
SelectFree[(a + b) c, b]
```

$$c$$

```
SelectFree2[(a + b) c, b]
```

$$ac$$

```
SelectFree[a, b]
```

$$a$$

```
SelectFree2[a, b]
```

$$a$$

```
SelectFree[a, a]
```

$$1$$

```
SelectFree2[a, a]
```

$$0$$

When there are hidden zeros, **SelectFree2** obviously works better

```
SelectFree[(a - b + c)^2 - (a^2 - 2 a b + 2 a c + b^2 - 2 b c + c^2),a]
```

$$-b^2 + 2bc - c^2$$

```
SelectFree2[(a - b + c)^2 - (a^2 - 2 a b + 2 a c + b^2 - 2 b c + c^2),a]
```

$$0$$

3.60 SelectNotFree2

`SelectNotFree2[expr, a, b, ...]` is similar to **`SelectNotFree`** but it also differs from the latter in several respects.

If **`expr`** is a list, **`SelectNotFree2`** behaves exactly the same way as **`SelectNotFree`**.

If **`expr`** is not a list, **`SelectNotFree2`** first expands the expression w.r.t. the arguments via **`Expand2`**.

Furthermore, **`SelectNotFree2[a, b]`** returns **`0`**. This differs from the behavior of **`SelectFree`** but is consistent with the naive expectations when applying the function to a sum of terms.

3.60.1 See also

Overview, FreeQ2, SelectFree, SelectNotFree, SelectFree2.

3.60.2 Examples

Note the difference between SelectNotFree and SelectNotFree2

```
SelectNotFree[(a + b) c, b]
```

$$a + b$$

```
SelectNotFree2[(a + b) c, b]
```

$$bc$$

```
SelectNotFree[a, b]
```

$$1$$

```
SelectNotFree2[a, b]
```

$$0$$

Here the behavior is identical

```
SelectNotFree[a, a]
```

$$a$$

```
SelectNotFree2[a, a]
```

$$a$$

When there are hidden zeros, **SelectNotFree2** obviously works better

```
SelectNotFree[(a - b + c)^2 - (a^2 - 2 a b + 2 a c + b^2 - 2 b c + c^2), a]
```

$$-a^2 + (a - b + c)^2 + 2ab - 2ac$$

```
SelectNotFree2[(a - b + c)^2 - (a^2 - 2 a b + 2 a c + b^2 - 2 b c + c^2),
a]
```

$$0$$

3.61 SelectSplit

SelectSplit[l, p] constructs list of mutually exclusive subsets from **l** in which every element **li** satisfies a criterion **pj[li]** with **pj** from **p** and appends the subset of remaining unmatched elements.

3.61.1 See also

Overview, SelectFree.

3.61.2 Examples

```
SelectSplit[{a^2, b^3, c^4, d^5, e^6, f + g, h^4}, {MatchQ[#, _^2] &,
MatchQ[#, _^4] &, FreeQ[#, Power] &}]
```

$$\{\{a^2\}, \{c^4, h^4\}, \{f + g\}, \{b^3, d^5, e^6\}\}$$

```
SelectSplit[{a^2, b^3, c^4, d^5, e^6, f + g, h^4}, {FreeQ[#, Plus] &,
FreeQ[#, Power] &}]
```

$$\left\{\left\{a^2, b^3, c^4, d^5, e^6, h^4\right\}, \{f + g\}, \{\}\right\}$$

3.62 FCSubsetQ

FCSubsetQ[list1, list2] yields **True** if **list2** is a subset of **list1** and **False** otherwise. It returns the same results as the standard **SubsetQ**. The only reason for introducing **FCSubsetQ** is that **SubsetQ** is not available in Mathematica 8 and 9, which are still supported by FeynCalc.

3.62.1 See also

Overview, FCDuplicateFreeQ.

3.62.2 Examples

```
FCSubsetQ[{a, b, c, d}, {a, d, e}]
```

False

```
FCSubsetQ[{a, b, c, d}, {a, d}]
```

True

3.63 FCSymmetrize

FCSymmetrize[expr, {a1, a2, ...}] symmetrizes expr with respect to the variables **a1, a2, ...**.

3.63.1 See also

Overview, FCAntiSymmetrize.

3.63.2 Examples

```
FCSymmetrize[f[a, b], {a, b}]
```

$$\frac{1}{2}(f(a,b)+f(b,a))$$

```
FCSymmetrize[f[x, y, z], {x, y, z}]
```

$$\frac{1}{6}(f(x,y,z)+f(x,z,y)+f(y,x,z)+f(y,z,x)+f(z,x,y)+f(z,y,x))$$

3.64 FeynCalcHowToCite

FeynCalcHowToCite[] lists publications that should be cited when mentioning FeynCalc in scientific works.

3.64.1 See also

Overview

3.64.2 Examples

```
FeynCalcHowToCite[]
```


- V. Shtabovenko, R. Mertig and F. Orellana, Comput.Phys.Commun. 256 (2020) 107478, arXiv:2001.04407.

- V. Shtabovenko, R. Mertig and F. Orellana, Comput.Phys.Commun. 207 (2016) 432-444, arXiv:1601.01167.

- R. Mertig, M. Böhm, and A. Denner, Comput. Phys. Commun. 64 (1991) 345-359.

3.65 FI

FI changes the output format to **InputForm**. This is useful to see the internal representation of FeynCalc objects. To change back to FeynCalcForm use FC.

3.65.1 See also

Overview, FeynCalcForm, FC, FeynCalcExternal, FeynCalcInternal.

3.65.2 Examples

3.66 FreeQ2

FreeQ2[expr, {form1, form2, ...}] yields **True** if **expr** does not contain any occurrence of **form1, form2, ...** and **False** otherwise.

FreeQ2[expr, form] is the same as **FreeQ[expr, form]**.

3.66.1 See also

Overview, SelectFree, SelectNotFree.

3.66.2 Examples

```
FreeQ2[x + f[x] + y, {a, x}]
```

False

```
FreeQ2[x + f[x] + y, {a, b}]
```

True

```
FreeQ2[x, y]
```

True

```
FreeQ2[f[x], f]
```

False

3.67 FRH

FRH[exp_] corresponds to **FixedPoint[ReleaseHold, exp]**, i.e. **FRH** removes all **HoldForm** and **Hold** in **exp**.

3.67.1 See also

Overview, Isolate.

3.67.2 Examples

```
Hold[1 - 1 - Hold[2 - 2]]
```

$$\text{Hold}[-\text{Hold}[2-2]+1-1]$$

```
FRH[%]
```

$$0$$

```
Isolate[ToRadicals[Solve[x^3 - x - 1 == 0]], x, IsolateNames -> KK]
```

$$\{\{x \to \text{KK}(21)\}, \{x \to \text{KK}(24)\}, \{x \to \text{KK}(25)\}\}$$

```
FRH[%]
```

$$\left\{\left\{x \to \frac{1}{3}\sqrt[3]{\frac{27}{2}-\frac{3\sqrt{69}}{2}}+\frac{\sqrt[3]{\frac{1}{2}\left(9+\sqrt{69}\right)}}{3^{2/3}}\right\},\right.$$
$$\left\{x \to -\frac{1}{6}\left(1-i\sqrt{3}\right)\sqrt[3]{\frac{27}{2}-\frac{3\sqrt{69}}{2}}-\frac{\left(1+i\sqrt{3}\right)\sqrt[3]{\frac{1}{2}\left(9+\sqrt{69}\right)}}{2\,3^{2/3}}\right\},$$
$$\left.\left\{x \to -\frac{1}{6}\left(1+i\sqrt{3}\right)\sqrt[3]{\frac{27}{2}-\frac{3\sqrt{69}}{2}}-\frac{\left(1-i\sqrt{3}\right)\sqrt[3]{\frac{1}{2}\left(9+\sqrt{69}\right)}}{2\,3^{2/3}}\right\}\right\}$$

3.68 ILimit

`ILimit[exp, a -> b]` checks functions specified by the option FunctionLimits and takes the limit **`a->b`** of these functions only if it is finite. For the rest of the expression **`exp`**, the limit is taken.

3.68.1 See also

Overview, FunctionLimits.

3.68.2 Examples

3.69 MLimit

`MLimit[expr, lims]` takes multiple limits of **`expr`** using the limits **`lims`**.

3.69.1 See also

Overview

3.69.2 Examples

```
MLimit[y Log[y] + Sin[x - 1]/(x - 1), {x -> 1, y -> 0}]
```

1

3.70 Isolate

`Isolate[expr]` substitutes abbreviations **`KK[i]`** for all **`Plus[...]`** (sub-sums) in **`expr`**. The inserted **`KK[i]`** have head **`HoldForm`**. **`Isolate[expr, varlist]`** substitutes **`KK[i]`** for all subsums in **`expr`** which are free of any occurrence of a member of the list **`varlist`**. Instead of **`KK`** any other head or a list of names of the abbreviations may be specified with the option **`IsolateNames`**.

3.70.1 See also

Overview, IsolateNames, Collect2.

3.70.2 Examples

```
t0 = Isolate[a + b]
```

KK(19)

```
t1 = Isolate[(a + b) f + (c + d) f + e, f]
```

$$e + f\ \mathrm{KK}(19) + f\ \mathrm{KK}(20)$$

```
StandardForm[t1]
```

$$e + f\ \mathrm{KK}[19] + f\ \mathrm{KK}[20]$$

```
{t0, t1, ReleaseHold[t1]}
```

$$\{\mathrm{KK}(19), e + f\ \mathrm{KK}(19) + f\ \mathrm{KK}(20), f(a+b) + f(c+d) + e\}$$

```
Isolate[a[z] (b + c (y + z)) + d[z] (y + z), {a, d}, IsolateNames -> fF]
```

$$\mathrm{fF}(22)a(z) + \mathrm{fF}(21)d(z)$$

```
Information[fF]
```

Symbol

Global`fF

Definitions

$$fF[21] = y + z$$

$$fF[22] = b + c\, fF[21]$$

Full Name Global`fF

```
Global`fF
```

fF

```
fF[26] = y + z
```

$y + z$

```
fF[27] = b + c HoldForm[fF[26]]
```

$$b + c\ \mathrm{fF}(26)$$

```
Isolate[a - b - c - d - e, IsolateNames -> l, IsolateSplit -> 15]
```

$$l(24)$$

```
Clear[t0, t1, l, fF]
```

3.71 KK

KK[i] is the default setting of **IsolateNames**, which is the head of abbreviations used by **Isolate**. A **KK[i]** returned by **Isolate** is given in **HoldForm** and can be recovered by **ReleaseHold[KK[i]]**.

3.71.1 See also

Overview, Isolate, IsolateNames.

3.71.2 Examples

3.72 Map2

Map2[f, exp] is equivalent to **Map** if **Nterms[exp] > 0**, otherwise **Map2[f, exp]** gives **f[exp]**.

3.72.1 See also

Overview, NTerms.

3.72.2 Examples

```
Map2[f, a - b]
```

$$f(a) + f(-b)$$

```
Map2[f, x]
```

$$f(x)$$

```
Map2[f, {a, b, c}]
```

$$f(\{a, b, c\})$$

```
Map2[f, 1]
```

$$f(1)$$

3.73 MemSet

MemSet[f[x_], body] is like **f[x_] := f[x] = body**, but depending on the value of the setting of **FCMemoryAvailable -> memorycut** (**memorycut** - **MemoryInUse[]/10^6**)

MemSet[f[x_], body] may evaluate as **f[x_] := body**.

3.73.1 See also

Overview, FCMemoryAvailable.

3.73.2 Examples

3.74 NTerms

NTerms[x] is equivalent to **Length** if **x** is a sum; otherwise **NTerms[x]** returns **1**, except **NTerms[0] -> 0**.

3.74.1 See also

Overview

3.74.2 Examples

```
NTerms[a - b]
```

$$2$$

```
NTerms[a b c]
```

$$1$$

```
NTerms[9]
```

$$1$$

```
NTerms[0]
```

$$0$$

3.75 NumericalFactor

`NumericalFactor[expr]` gives the overall numerical factor of **`expr`**.

3.75.1 See also

Overview, FCFactorOut.

3.75.2 Examples

3.76 NumericQ1

`NumericQ1[x, {a, b, ..}]` is like **`NumericQ`**, but assumes that **`{a, b, ..}`** are numeric quantities.

3.76.1 See also

Overview

3.76.2 Examples

```
NumericQ[3 a + Log[b] + c^2]
```

False

```
NumericQ1[3 a + Log[b] + c^2, {}]
```

False

```
NumericQ1[3 a + Log[b] + c^2, {a, b, c}]
```

True

3.77 Power2

Power2[x, y] represents **x^y**. Sometimes **Power2** is more useful than the Mathematica **Power**. **Power2[-a, b]** simplifies to **(-1)^b Power2[a, b]** (if no **Epsilon** is in **b** . . .).

3.77.1 See also

Overview, PowerFactor.

3.77.2 Examples

```
Power[-a, b]
```

$$(-a)^b$$

```
Power2[-a, b]
```

$$(-1)^b a^b$$

3.78 PowerFactor

PowerFactor[exp] replaces **x^a y^a** with **(x y)^a**.

3.78.1 See also

Overview, PowerSimplify.

3.78.2 Examples

```
x^a y^a
```

```
PowerFactor[%]
```

$$x^a y^a$$

$$(xy)^a$$

3.79 PowerSimplify

PowerSimplify[exp] simplifies **(-x)^a** to **(-1)^a x^a** and **(y-x)^n** to **(-1)^n (x-y)^n** thus assuming that the exponent is an integer (even if it is symbolic).

Furthermore, **(-1)^(a+n)** and **I^(a+n)** are expanded and **(I)^(2 m) -> (-1)^m and (-1)^(n_Integer?EvenQ m) -> 1** and **(-1)^(n_Integer?OddQ m) -> (-1)^m** for **n** even and odd respectively and **(-1)^(-n) -> (-1)^n** and **Exp[I m Pi] -> (-1)^m**.

3.79.1 See also

Overview, DataType, OPEm.

3.79.2 Examples

```
PowerSimplify[(-1)^(2 OPEm)]
```

$$1$$

```
PowerSimplify[(-1)^(OPEm + 2)]
```

$$(-1)^m$$

```
PowerSimplify[(-1)^(OPEm - 2)]
```

$$(-1)^m$$

```
PowerSimplify[I^(2 OPEm)]
```

$$(-1)^m$$

3.80 XYT

XYT[exp, x, y] transforms **(x y)^m** away.

3.80.1 See also

Overview

3.80.2 Examples

3.81 Series2

Series2 performs a series expansion around **0**. **Series2** is (up to the **Gamma**-bug in Mathematica versions smaller than 5.0) equivalent to **Series**, except that it applies **Normal** on the result and has an option **FinalSubstitutions**.

Series2[f, e, n] is equivalent to **Series2[f, {e, 0, n}]**.

3.81.1 See also

Overview, Series3.

3.81.2 Examples

```
Series2[(x (1 - x))^(\[Delta]/2), \[Delta], 1]
```

$$\frac{1}{2}\delta \log(1-x) + \frac{1}{2}\delta \log(x) + 1$$

```
Series2[Gamma[x], x, 1]
```

$$\frac{1}{2}\zeta(2)x + \frac{1}{x}$$

```
Series[Gamma[x], {x, 0, 1}]
```

$$\frac{1}{x} - \gamma + \frac{1}{12}\left(6\gamma^2 + \pi^2\right)x + O\left(x^2\right)$$

```
Series2[Gamma[x], x, 2]
```

$$-\frac{x^2\zeta(3)}{3} + \frac{1}{2}\zeta(2)x + \frac{1}{x}$$

```
Series2[Gamma[x], x, 2, FinalSubstitutions -> {}] // FullSimplify
```

$$\frac{1}{6}\left(-3\gamma\left(\zeta(2)x^2 + 2\right) - 2x^2\zeta(3) - \gamma^3x^2 + 3\zeta(2)x + 3\gamma^2x + \frac{6}{x}\right)$$

```
Series[Gamma[x], {x, 0, If[$VersionNumber < 5, 4, 2]}] // Normal // Expand
// FullSimplify
```

$$\frac{1}{12}\left(-2\gamma^3x^2 - \gamma\left(\pi^2x^2 + 12\right) + x\left(\pi^2 - 4x\zeta(3)\right) + 6\gamma^2x + \frac{12}{x}\right)$$

There is a table of expansions of special hypergeometric functions.

```
Series2[HypergeometricPFQ[{1, OPEm - 1, Epsilon/2 + OPEm}, {OPEm, OPEm +
Epsilon}, 1], Epsilon, 1]
```

$$-\frac{2}{\varepsilon} + \frac{2m}{\varepsilon} + \frac{1}{2}\varepsilon m\psi^{(1)}(m) - \frac{\varepsilon\psi^{(1)}(m)}{2} + 1$$

```
Series2[HypergeometricPFQ[{1, OPEm, Epsilon/2 + OPEm}, {1 + OPEm, Epsilon +
OPEm},  1], Epsilon, 1]
```

$$\frac{1}{4}\varepsilon\zeta(2)m + \frac{2m}{\varepsilon} + \frac{1}{4}\varepsilon m\psi^{(0)}(m)^2 + \frac{3}{4}\varepsilon m\psi^{(1)}(m) - \frac{1}{2}\varepsilon mS_{11}(m-1)$$

```
Hypergeometric2F1[1, Epsilon, 1 + 2 Epsilon, x]

Series2[%, Epsilon, 3]
```

$$ {}_2F_1(1, \varepsilon; 2\varepsilon + 1; x) $$

$$ \begin{aligned} &-2\varepsilon^2\zeta(2) + 2\varepsilon^3 \operatorname{Li}_3(1-x) + 2\varepsilon^2 \operatorname{Li}_2(1-x) - 4\varepsilon^3 \operatorname{Li}_2(1-x)\log(x) - 4\varepsilon^3 S_{12}(1-x) \\ &- 2\varepsilon^3\zeta(2)\log(1-x) + 4\varepsilon^3\zeta(2)\log(x) - \frac{1}{6}\varepsilon^3\log^3(1-x) - 2\varepsilon^3\log(1-x)\log^2(x) \\ &+ \varepsilon^3\log^2(1-x)\log(x) - \frac{1}{2}\varepsilon^2\log^2(1-x) + 2\varepsilon^2\log(1-x)\log(x) - \varepsilon\log(1-x) + 2\varepsilon^3\zeta(3) + 1 \end{aligned} $$

There are over 100 more special expansions of ${}_2F_1$ tabulated in **`Series2.m`**. The interested user can consult the source code (search for HYPERLIST).

3.82 Series3

`Series3` performs a series expansion around **`0`**. **`Series3`** is equivalent to **`Series`**, except that it applies **`Normal`** on the result and that some **`Series`** bugs are fixed.

`Series3[f, e, n]` is equivalent to **`Series3[f, {e, 0, n}]`**.

3.82.1 See also

Overview, Series2.

3.82.2 Examples

```
Series3[(x (1 - x))^(\[Delta]/2), \[Delta], 1]
```

$$ \frac{1}{2}\delta\log((1-x)x) + 1 $$

```
Series3[Gamma[x], x, 1] // FullSimplify
```

$$ \frac{1}{x} - \gamma + 1 $$

3.83 SetStandardMatrixElements

SetStandardMatrixElements[{sm1 -> abb1}, {sm2 -> abb2}, ...] sets abbreviations **abb1, abb2, ...** for matrix elements **sm1, sm2, ...**.

SetStandardMatrixElements[{sm1 -> abb1}, {sm2 -> abb2}, ..., cons]. Set abbreviations **abb1, abb2, ...** for matrix elements **sm1, sm2, ...** using energy-momentum conservation **cons**, e.g. **k2 -> p1 + p2 - k1**

3.83.1 See also

Overview, OneLoop.

3.83.2 Examples

3.84 Solve2

Solve2 is equivalent to **Solve**, except that it works only for linear equations (and returns just a list) and accepts the options **Factoring** and **FinalSubstitutions**.

Solve2 uses the "high school algorithm" and factors intermediate results. Therefore it can be drastically more useful than **Solve**.

3.84.1 See also

Overview, Solve3.

3.84.2 Examples

```
Solve2[{2 x == b - w/2, y - d == p}, {x, y}]
```

$$\left\{x \to \frac{1}{4}(2b - w), y \to d + p\right\}$$

If no equation sign is given the polynomials are supposed to be 0.

```
Solve2[x + y, x]
```

$$\{x \to -y\}$$

```
Solve2[x + y, x, FinalSubstitutions -> {y -> h}]
```

$$\{x \to -h\}$$

```
Solve2[{2 x == b - w/2, y - d == p}, {x, y}, Factoring -> Expand]
```

$$\left\{x \to \frac{b}{2} - \frac{w}{4}, y \to d + p\right\}$$

```
Solve[{2 x == b - w/2, y - d == p}, {x, y}]
```

$$\left\{\left\{x \to \frac{1}{4}(2b - w), y \to d + p\right\}\right\}$$

3.85 Solve3

Solve3 is equivalent to **Solve**, except that it works only for linear equations (and returns just a list) and uses the "high school algorithm.

Sometimes it is better than **Solve** for systems involving rational polynomials.

3.85.1 See also

Overview, Solve2.

3.85.2 Examples

```
Solve3[{2 x == b - w/2, y - d == p}, {x, y}]
```

$$\left\{x \to \frac{1}{4}(2b - w), y \to d + p\right\}$$

3.86 SumP

SumP[k, m] is $2^{k-1} \sum_{i=1}^{2m} \left(1 + (-1)^i\right) / i^k$.

3.86.1 See also

Overview, SumS, SumT.

3.86.2 Examples

```
SumP[1, m - 1]
```

$$S_1^{\prime(m-1)}$$

```
SumP[2, m - 1]
```

$$S_2^{\prime(m-1)}$$

```
SumP[1, m]
```

$$S_1^{\prime(m)}$$

```
SumP[1, 4]
```

$$\frac{25}{12}$$

```
Explicit[SumP[1, n/2]]

% /. n -> 8
```

$$\frac{1}{2}\left(1-(-1)^n\right) S_1\left(\frac{n-1}{2}\right)+\frac{1}{2}\left((-1)^n+1\right) S_1\left(\frac{n}{2}\right)$$

$$\frac{25}{12}$$

3.87 SumS

SumS[1, m] is the harmonic number $S_1(m) = \sum_{i=1}^{m} i^{-1}$.

SumS[1, 1, m] is $\sum_{i=1}^{m} S_1(i)/i$.

SumS[k, l, m] is $\sum_{i=1}^{m} S_l(i)/i^k$.

SumS[r, n] represents **Sum[Sign[r]^i/i^Abs[r], {i, 1, n}]**.

SumS[r, s, n] is **Sum[Sign[r]^k/k^Abs[r] Sign[s]^j/j^Abs[s], {k, 1, n}, {j, 1, k}]** etc.

3.87.1 See also

Overview, SumP, SumT.

3.87.2 Examples

```
SumS[1, m - 1]
```

$$S_1(m-1)$$

```
SumS[2, m - 1]
```

$$S_2(m-1)$$

```
SumS[-1, m]
```

$$S_{-1}(m)$$

```
SumS[1, m, Reduce -> True]
```

$$S_1(m-1)+\frac{1}{m}$$

```
SumS[3, m + 2, Reduce -> True]
```

$$S_3(m+1)+\frac{1}{(m+2)^3}$$

```
SetOptions[SumS, Reduce -> True];
SumS[3, m + 2]
```

$$\frac{1}{m^3}+S_3(m-1)+\frac{1}{(m+1)^3}+\frac{1}{(m+2)^3}$$

```
SetOptions[SumS, Reduce -> False];
SumS[1, 4]
```

$$\frac{25}{12}$$

```
SumS[1, 2, m - 1]
```

$$S_{12}(m-1)$$

```
SumS[1, 1, 1, 11]
```

$$\frac{31276937512951}{4260000729600}$$

```
SumS[-1, 4]
```

$$-\frac{7}{12}$$

```
SumT[1, 4]
```

$$-\frac{7}{12}$$

3.88 SumT

SumT[1, m] is the alternative harmonic number $\sum_{i=1}^{m}(-1)\hat{}i/i$

SumT[r, n] represents **Sum[(-1)^i/i^r, {i, 1, n}]**

SumT[r, s, n] is **Sum[1/k^r (-1)^j/j^s, {k, 1, n}, {j, 1, k}]**.

3.88.1 See also

Overview, SumP, SumS.

3.88.2 Examples

```
SumT[1, m - 1]
```

$$\tilde{S}_1(m-1)$$

```
SumT[2, m - 1]
```

$$\tilde{S}_2(m-1)$$

```
SumT[1, m]
```

$$\tilde{S}_1(m)$$

```
SumT[1, m, Reduce -> True]
```

$$\tilde{S}_1(m-1) + \frac{(-1)^m}{m}$$

```
SumT[1, 4]
```

$$-\frac{7}{12}$$

```
SumT[1, 2, m - 1]
```

$$\tilde{S}_{12}(m-1)$$

```
SumT[1, 2, 42]
```

$$-\frac{3898795869705501336048986429870362142961015213868392 7}{1051212166070237840531600496448376108087919052800000 0}$$

```
SumT[1, 4]
```

$$-\frac{7}{12}$$

```
SumS[-1, 4]
```

$$-\frac{7}{12}$$

```
SumT[1, 2, 12]
```

$$-\frac{57561743656913}{21300003648000}$$

```
SumS[1, -2, 42]
```

$$-\frac{3898795869705501336048986429870362142961015213868392 7}{10512121660702378405316004964483761080879190528000000}$$

```
Array[SumT, 6]
```

$$\left\{-1, -\frac{5}{8}, -\frac{179}{216}, -\frac{1207}{1728}, -\frac{170603}{216000}, -\frac{155903}{216000}\right\}$$

```
Array[SumS[-2, 1, #1] &, 6]
```

$$\left\{-1, -\frac{5}{8}, -\frac{179}{216}, -\frac{1207}{1728}, -\frac{170603}{216000}, -\frac{155903}{216000}\right\}$$

3.89 TimedIntegrate

`TimedIntegrate[exp, vars]` is like **`Integrate`**, but stops after the number of seconds specified by the option **`Timing`**. Options of **`Integrate`** can be given and are passed on.

3.89.1 See also

Overview

3.89.2 Examples

This should reach to be done

```
TimedIntegrate[Log[x^5], {x, 0, 1}, Timing -> 1]
```

$$-5$$

This shouldn't

```
TimedIntegrate[Log[Cos[x^5]], {x, 0, 1}, Timing -> 10, Integrate -> int]
```

$$\text{int}\left(\log\left(\cos\left(x^5\right)\right), \{x, 0, 1\}, \text{Assumptions} \to \varepsilon > 0\right)$$

3.90 TBox

`TBox[a, b, ...]` produces a **`RowBox[{a, b, ...}]`** where **`a, b, ...`** are boxed in **`TraditionalForm`**.

`TBox` is used internally by FeynCalc to produce the typeset output in **`TraditionalForm`**.

3.90.1 See also

Overview

3.90.2 Examples

```
TBox[a + b]
% // DisplayForm
```

$$\text{FormBox}[\text{RowBox}[\{\text{a}, +, \text{b}\}], \text{TraditionalForm}]$$

$$a + b$$

3.91 TypesettingExplicitLorentzIndex

`TypesettingExplicitLorentzIndex` determines the **`TraditionalForm`** typesetting of explicit Lorentz indices.

3.91.1 See also

Overview.

3.91.2 Examples

Current setting

```
TypesettingExplicitLorentzIndex
% // InputForm
```

$$\text{FeynCalc`SharedObjects`Private`x} \to \text{FeynCalc`SharedObjects`Private`x}$$

```
Function[FeynCalc`SharedObjects`Private`x,
 FeynCalc`SharedObjects`Private`x]
```

Make explicit Lorentz indices look red

```
TypesettingExplicitLorentzIndex = Function[x, Style[x, Red]];

4 M^2 u FV[k, 0]^2 - 4 M^2 u FV[k, 3]^2 - 4 M SP[k, k] - 2 M u FV[k, 0]
FV[k, 3]^2 + 4 M u FV[k, 0] FV[k, 2] - u^2 FV[k, 2]^2
```

$$-4M^2u\left(\overline{k}^3\right)^2 - 2k^0Mu\left(\overline{k}^3\right)^2 + 4k^0Mu\overline{k}^2 - 4M\overline{k}^2 - u^2\left(\overline{k}^2\right)^2 + 4\left(k^0\right)^2M^2u$$

Back to the standard settings

```
TypesettingExplicitLorentzIndex = Function[x, x]
```

$$x \to x$$

3.92 $TypesettingDim4

The string value of $TypesettingDim4 determines which symbols will be displayed above 4-dimensional momenta, Dirac matrices, metric tensors and polarization vectors. This concerns only typesetting in the **`TraditionalForm`** output and doesn't change the physical behavior of those objects.

3.92.1 See also

Overview, $TypesettingDimD, $TypesettingDimE.

3.92.2 Examples

```
$TypesettingDim4
```

–

3.93 $TypesettingDimD

The string value of **$TypesettingDimD** determines which symbols will be displayed above D-dimensional momenta, Dirac matrices, metric tensors and polarization vectors. This concerns only typesetting in the **TraditionalForm** output and doesn't change the physical behavior of those objects.

3.93.1 See also

Overview, $TypesettingDim4, $TypesettingDimE.

3.93.2 Examples

```
$TypesettingDimD
```

3.94 $TypesettingDimE

The string value of **$TypesettingDimE** determines which symbols will be displayed above $(D-4)$-dimensional momenta, Dirac matrices, metric tensors and polarization vectors. This concerns only typesetting in the **TraditionalForm** output and doesn't change the physical behavior of those objects.

3.94.1 See also

Overview, $TypesettingDim4, $TypesettingDimD.

3.94.2 Examples

```
$TypesettingDimE
```

^

3.95 Variables2

`Variables2[expr]` is like **`Variables`**, but it also works on rules and equalities as well as lists thereof. **`Variables2`** always applies **`Union`** to the output.

3.95.1 See also

Overview, Cases2.

3.95.2 Examples

Some cases where **`Variables2`** is much more useful than **`Variables`**

```
Variables[{a -> x1 + y1, b -> x2 + y2}]
```

$$\{\}$$

```
Variables2[{a -> x1 + y1, b -> x2 + y2}]
```

$$\{a, b, \text{x1}, \text{x2}, \text{y1}, \text{y2}\}$$

```
Variables[a + b == c + d]
```

$$\{\}$$

```
Variables2[a + b == c + d]
```

$$\{a, b, c, d\}$$

4 Lorentz and Cartesian tensors

4.1 Amputate

`Amputate[exp, q1, q2, ...]` amputates **`Eps`** and **`DiracGamma`**. **`Amputate[exp, q1, q2, Pair->{p}]`** amputates also **`p.q1`** and **`p.q2`**; **`Pair->All`** amputates all except **`OPEDelta`**.

4.1.1 See also

Overview, DiracGamma, GA, DiracSimplify, GS, DiracTrick.

4.1.2 Examples

```
GS[p] . GS[q]
```

$$(\bar{\gamma}\cdot\overline{p})\,.\,(\bar{\gamma}\cdot\overline{q})$$

```
Amputate[%, q]
```

$$q^{\$\text{AL}\$16479(1)}\,(\bar{\gamma}\cdot\overline{p})\,.\gamma^{\$\text{AL}\$16479(1)}$$

4.2 CartesianToLorentz

`CartesianToLorentz[exp]` rewrites Cartesian tensors in form of Lorentz tensors (when possible). Using options one can specify which types of tensors should be converted.

4.2.1 See also

Overview, LorentzToCartesian.

4.2.2 Examples

```
CGS[p]
```

```
% // CartesianToLorentz
```

$$\overline{\gamma} \cdot \overline{p}$$

$$p^0 \bar{\gamma}^0 - \bar{\gamma} \cdot \overline{p}$$

```
CSP[p, q]
```

```
% // CartesianToLorentz
```

$$\overline{p} \cdot \overline{q}$$

$$p^0 q^0 - \overline{p} \cdot \overline{q}$$

4.3 CartesianPairContract

`CartesianPairContract` is like **`CartesianPair`**, but with (local) contraction properties. The function fully supports the BMHV algebra and will not expand momenta inside scalar products.

`CartesianPairContract` is an auxiliary function used in higher level FeynCalc functions that require fast contractions between multiple expressions, where **`Contract`** would be too slow.

4.3.1 See also

Overview, CartesianPair, Contract.

4.3.2 Examples

```
CartesianPair[CartesianIndex[i], CartesianMomentum[p]]
CartesianPair[CartesianIndex[i], CartesianMomentum[q]]

% /. CartesianPair -> CartesianPairContract

% /. CartesianPairContract -> CartesianPair
```

$$\overline{p}^i \overline{q}^i$$

$$\text{CartesianPairContract}\left(\overline{p}, \overline{q}\right)$$

$$\overline{p} \cdot \overline{q}$$

```
CartesianPair[CartesianIndex[i], CartesianMomentum[p]]
CartesianPair[CartesianIndex[j], CartesianMomentum[q]]
CartesianPair[CartesianIndex[i], CartesianIndex[j]]

% /. CartesianPair -> CartesianPairContract

% /. CartesianPairContract -> CartesianPair
```

$$\overline{p}^i \overline{q}^j \bar{\delta}^{ij}$$

$$\text{CartesianPairContract}\left(\overline{p}, \overline{q}\right)$$

$$\overline{p} \cdot \overline{q}$$

```
CartesianPair[CartesianIndex[i], CartesianMomentum[p + q]]
CartesianPair[CartesianIndex[i], CartesianMomentum[r + s]]

% /. CartesianPair -> CartesianPairContract

% /. CartesianPairContract -> CartesianPair
```

$$\left(\overline{p} + \overline{q}\right)^i \left(\overline{r} + \overline{s}\right)^i$$

$$\text{CartesianPairContract}\left(\overline{p} + \overline{q}, \overline{r} + \overline{s}\right)$$

$$\left(\overline{p} + \overline{q}\right) \cdot \left(\overline{r} + \overline{s}\right)$$

4.4 CartesianScalarProduct

CartesianScalarProduct[p, q] is the input for the scalar product of two Cartesian vectors **p** and **q**.

CartesianScalarProduct[p] is equivalent to **CartesianScalarProduct[p, p]**.

Expansion of sums of momenta in **CartesianScalarProduct** is done with **ExpandScalarProduct**.

Scalar products may be set, e.g. via **CartesianScalarProduct[a, b] = m^2;** but **a** and **b** may not contain sums.

CartesianScalarProduct[a] corresponds to **CartesianScalarProduct[a, a]**

Note that **ScalarProduct[a, b] = m^2** actually sets Cartesian scalar products in different dimensions specified by the value of the **SetDimensions** option.

It is highly recommended to set **CartesianScalarProduct**s before any calculation. This improves the performance of FeynCalc.

4.4.1 See also

Overview, CSP, CSPD, CSPE.

4.4.2 Examples

```
CartesianScalarProduct[p, q]
```

$$\overline{p} \cdot \overline{q}$$

```
CartesianScalarProduct[p + q, -q]
```

$$-(\overline{q} \cdot (\overline{p} + \overline{q}))$$

```
CartesianScalarProduct[p, p]
```

$$\overline{p}^2$$

```
CartesianScalarProduct[q]
```

$$\overline{q}^2$$

```
CartesianScalarProduct[p, q] // StandardForm

(*CartesianPair[CartesianMomentum[p], CartesianMomentum[q]]*)
```

```
CartesianScalarProduct[p, q, Dimension -> D - 1] // StandardForm

(*CartesianPair[CartesianMomentum[p, -1 + D], CartesianMomentum[q, -1 +
D]]*)
```

```
CartesianScalarProduct[Subscript[p, 1], Subscript[p, 2]] = s/2
```

$$\frac{s}{2}$$

```
ExpandScalarProduct[ CartesianScalarProduct[Subscript[p, 1] - q,
Subscript[p, 2] - k]]
```

$$-\overline{k}\cdot\overline{p}_1+\overline{k}\cdot\overline{q}-\overline{q}\cdot\overline{p}_2+\frac{s}{2}$$

```
Calc[ CartesianScalarProduct[Subscript[p, 1] - q, Subscript[p, 2] - k]]
```

$$-\overline{k}\cdot\overline{p}_1+\overline{k}\cdot\overline{q}-\overline{q}\cdot\overline{p}_2+\frac{s}{2}$$

```
CartesianScalarProduct[q1] = qq;
```

```
CSP[q1]
```

$$\text{qq}$$

```
FCClearScalarProducts[]
```

4.5 ChangeDimension

ChangeDimension[exp, dim] changes all **LorentzIndex** and **Momentum** symbols in **exp** to dimension **dim** (and also Levi-Civita-tensors, Dirac slashes and Dirac matrices).

Notice that the dimension of **CartesianIndex** and **CartesianMomentum** objects will be changed to **dim-1**, not **dim**.

4.5.1 See also

Overview, LorentzIndex, Momentum, DiracGamma, Eps.

4.5.2 Examples

Remember that **`LorentzIndex[mu, 4]`** is simplified to **`LorentzIndex[mu]`** and **`Momentum[p, 4]`** to **`Momentum[p]`**. Thus the following objects are defined in four dimensions.

```
{LorentzIndex[\[Mu]], Momentum[p]}

ex = ChangeDimension[%, D]
```

$$\{\mu, \overline{p}\}$$

$$\{\mu, p\}$$

```
ex // StandardForm

(*{LorentzIndex[\[Mu], D], Momentum[p, D]}*)
```

This changes all non-4-dimensional objects to 4-dimensional ones

```
ChangeDimension[%%, 4] // StandardForm

(*{LorentzIndex[\[Mu]], Momentum[p]}*)
```

Consider the following list of 4- and D-dimensional objects

```
{GA[\[Mu], \[Nu]] MT[\[Mu], \[Nu]], GAD[\[Mu], \[Nu]] MTD[\[Mu], \[Nu]]
f[D]}

DiracTrick /@ Contract /@ %

DiracTrick /@ Contract /@ ChangeDimension[%%, n]
```

$$\{\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}\bar{g}^{\mu\nu}, f(D)\gamma^{\mu}.\gamma^{\nu}g^{\mu\nu}\}$$

$$\{4, Df(D)\}$$

$$\{n, nf(D)\}$$

Any explicit occurrence of D (like in $f(D)$) is not replaced by **ChangeDimension**.

```
LC[\[Mu], \[Nu], \[Rho], \[Sigma]]

ChangeDimension[%, D]

Factor2[Contract[%^2]]
```

$$\bar{\epsilon}^{\mu\nu\rho\sigma}$$

$$\epsilon^{\mu\nu\rho\sigma}$$

$$(1-D)(2-D)(3-D)D$$

```
Contract[LC[\[Mu], \[Nu], \[Rho], \[Sigma]]^2]
```

$$-24$$

4.6 CompleteSquare

CompleteSquarep[exp, x] completes the square of a second order polynomial in the momentum x.

4.6.1 See also

Overview, ExpandScalarProduct.

4.6.2 Examples

```
CompleteSquare[4 SP[p] + SP[b, p] + c, p]
```

$$4(\frac{\overline{b}}{8}+\overline{p})^2 - \frac{\overline{b}^2}{16} + c$$

```
CompleteSquare[4 SP[p] + SP[b, p] + c, p, q]
```

$$\left\{-\frac{\overline{b}^2}{16} + 4\overline{q}^2 + c, \overline{q} \to \frac{\overline{b}}{8} + \overline{p}\right\}$$

```
ex1 = 5 SP[2 p + 3 r, p + r]
```

$$5\,((\overline{p}+\overline{r})\cdot(2\overline{p}+3\overline{r}))$$

```
ex2 = CompleteSquare[ex1, p]
```

$$10(\overline{p}+\frac{5\overline{r}}{4})^2-\frac{5\overline{r}^2}{8}$$

```
ex1 - ex2 // ScalarProductExpand // Expand
```

$$0$$

```
CompleteSquare[5 SP[2 p + 3 r, p + r], p, q]
```

$$\left\{10\overline{q}^2-\frac{5\overline{r}^2}{8},\overline{q}\to\overline{p}+\frac{5\overline{r}}{4}\right\}$$

```
SPD[a] + 2 SPD[a, b]

ex = CompleteSquare[%, a]
```

$$2(a\cdot b)+a^2$$

$$(a+b)^2-b^2$$

```
ex // StandardForm

(*-Pair[Momentum[b, D], Momentum[b, D]] + Pair[Momentum[a, D] + Momentum[b,
D], Momentum[a, D] + Momentum[b, D]]*)
```

4.7 Contract

Contract[expr] contracts pairs of Lorentz or Cartesian indices of metric tensors, vectors and (depending on the value of the option **EpsContract**) of Levi-Civita tensors in **expr**.

For contractions of Dirac matrices with each other use **DiracSimplify**.

Contract[exp1, exp2] contracts **(exp1*exp2)**, where **exp1** and **exp2** may be larger products of sums of metric tensors and 4-vectors. This can be also useful when evaluating polarization sums, where **exp2** should be the product (or expanded sum) of the polarization sums for the vector bosons.

4.7.1 See also

Overview, Pair, CartesianPair, DiracSimplify, MomentumCombine.

4.7.2 Examples

```
MT[\[Mu], \[Nu]] FV[p, \[Mu]]

Contract[%]
```

$$\overline{p}^{\mu}\bar{g}^{\mu\nu}$$

$$\overline{p}^{\nu}$$

```
FV[p, \[Mu]] GA[\[Mu]]

Contract[%]
```

$$\bar{\gamma}^{\mu}\overline{p}^{\mu}$$

$$\bar{\gamma}\cdot\overline{p}$$

The default dimension for a metric tensor is 4.

```
MT[\[Mu], \[Mu]]

Contract[%]
```

$$\bar{g}^{\mu\mu}$$

$$4$$

A quick way to enter D-dimensional metric tensors is given by **MTD**.

```
MTD[\[Mu], \[Nu]]  MTD[\[Mu], \[Nu]]
Contract[%]
```

$$(g^{\mu\nu})^2$$

$$D$$

```
FV[p, \[Mu]] FV[q, \[Mu]]
Contract[% ]
```

$$\overline{p}^{\mu}\overline{q}^{\mu}$$

$$\overline{p}\cdot\overline{q}$$

```
FV[p - q, \[Mu]] FV[a - b, \[Mu]]
Contract[%]
```

$$\left(\overline{a}-\overline{b}\right)^{\mu}\left(\overline{p}-\overline{q}\right)^{\mu}$$

$$\overline{a}\cdot\overline{p}-\overline{a}\cdot\overline{q}-\overline{b}\cdot\overline{p}+\overline{b}\cdot\overline{q}$$

```
FVD[p - q, \[Nu]] FVD[a - b, \[Nu]]
Contract[%]
```

$$(a-b)^{\nu}(p-q)^{\nu}$$

$$a\cdot p-a\cdot q-b\cdot p+b\cdot q$$

```
LC[\[Mu], \[Nu], \[Alpha], \[Sigma]] FV[p, \[Sigma]]

Contract[%]
```

$$\overline{p}^{\sigma}\bar{\epsilon}^{\mu\nu\alpha\sigma}$$

$$\bar{\epsilon}^{\alpha\mu\nu\overline{p}}$$

```
LC[\[Mu], \[Nu], \[Alpha], \[Beta]] LC[\[Mu], \[Nu], \[Alpha], \[Sigma]]

Contract[%]
```

$$\bar{\epsilon}^{\mu\nu\alpha\beta}\bar{\epsilon}^{\mu\nu\alpha\sigma}$$

$$-6\bar{g}^{\beta\sigma}$$

```
LCD[\[Mu], \[Nu], \[Alpha], \[Beta]] LCD[\[Mu], \[Nu], \[Alpha], \[Sigma]]

Contract[%] // Factor2
```

$$\epsilon^{\mu\nu\alpha\beta}\epsilon^{\mu\nu\alpha\sigma}$$

$$(1-D)(2-D)(3-D)g^{\beta\sigma}$$

Contractions of Cartesian tensors are also possible. They can live in 3, $D-1$ or $D-4$ dimensions.

```
KD[i, j] CV[p, i]

Contract[%]
```

$$\overline{p}^{i}\bar{\delta}^{ij}$$

$$\overline{p}^{j}$$

```
CV[p, i] CGA[i]
```

```
Contract[%]
```

$$\overline{\gamma}^{i}\overline{p}^{i}$$

$$\overline{\gamma} \cdot \overline{p}$$

```
KD[i, i]
```

```
Contract[%]
```

$$\bar{\delta}^{ii}$$

$$3$$

```
KD[i, j]^2
```

```
Contract[%]
```

$$(\bar{\delta}^{ij})^2$$

$$3$$

```
CV[p - q, j] CV[a - b, j]
```

```
Contract[%]
```

$$\left(\overline{a} - \overline{b}\right)^{j} \left(\overline{p} - \overline{q}\right)^{j}$$

$$(\overline{a} - \overline{b}) \cdot (\overline{p} - \overline{q})$$

```
CLC[i, j, k] CV[p, k]
```

```
Contract[%]
```

$$\overline{p}^{k}\,\overline{\epsilon}^{ijk}$$

$$\overline{\epsilon}^{ij\overline{p}}$$

```
CLC[i, j, k] CLC[i, j, l]

Contract[%]
```

$$\overline{\epsilon}^{ijk}\,\overline{\epsilon}^{ijl}$$

$$2\overline{\delta}^{kl}$$

```
CLCD[i, j, k] CLCD[i, j, l]

Contract[%] // Factor2
```

$$\epsilon^{ijk}\,\epsilon^{ijl}$$

$$(2-D)(3-D)\delta^{kl}$$

4.8 DeclareFCTensor

`DeclareFCTensor[a, b, ...]` declares **a**,**b**, ... to be tensor heads, i.e., **`DataType[a, b, ..., FCTensor]`** is set to **`True`**.

4.8.1 See also

Overview, ExpandScalarProduct, Uncontract.

4.8.2 Examples

```
ClearAll[myTens]

DeclareFCTensor[myTens]

ExpandScalarProduct[myTens[z, Momentum[a + b], Momentum[c + d]]]
```

$$\text{myTens}\,(z,\overline{a},\overline{c})+\text{myTens}\,(z,\overline{a},\overline{d})+\text{myTens}\,(z,\overline{b},\overline{c})+\text{myTens}\,(z,\overline{b},\overline{d})$$

```
UnDeclareFCTensor[myTens]
```

4.9 DummyIndexFreeQ

`DummyIndexFreeQ[exp, {head1, head2, ...}]` returns **`True`** if the expression contains dummy indices with heads **`head1`**, **`head2`**, ... and **`False`** otherwise.

As always in FeynCalc, Einstein summation convention is implicitly assumed.

The function is optimized for large expressions, i.e. it is not so good as a criterion in e.g. Select.

4.9.1 See also

Overview, FCRenameDummyIndices, Contract, FreeIndexFreeQ.

4.9.2 Examples

```
FCI[FV[p, \[Mu]] FV[q, \[Nu]]]

DummyIndexFreeQ[%, {LorentzIndex}]
```

$$\overline{p}^{\mu}\overline{q}^{\nu}$$

True

```
FCI[FV[p, \[Mu]] FV[q, \[Mu]]]

DummyIndexFreeQ[%, {LorentzIndex}]
```

$$\overline{p}^{\mu}\overline{q}^{\mu}$$

False

```
FCI[SUNT[a, b]]

DummyIndexFreeQ[%, {SUNIndex}]
```

$$T^{a}.T^{b}$$

True

```
FCI[SUNT[a, a]]
```

```
DummyIndexFreeQ[%, {SUNIndex}]
```

$$T^a.T^a$$

False

4.10 EpsContract

`EpsContract[exp]` handles contractions of two Levi-Civita tensors. It is also an option of **`Contract`** and other functions that specifies whether such contractions should be done or not.

4.10.1 See also

Overview, Eps, Contract.

4.10.2 Examples

```
LCD[\[Mu], \[Nu], \[Rho], \[Sigma]]
```

```
EpsContract[% %] // Factor2
```

$$\epsilon^{\mu\nu\rho\sigma}$$

$$(1-D)(2-D)(3-D)D$$

```
Contract[LCD[\[Mu], \[Nu], \[Rho], \[Sigma]]^2] // Factor2
```

$$(1-D)(2-D)(3-D)D$$

```
Contract[LCD[\[Mu], \[Nu], \[Rho], \[Sigma]]^2, EpsContract -> False]
```

$$(\epsilon^{\mu\nu\rho\sigma})^2$$

4.11 EpsContractFreeQ

`EpsContractFreeQ[exp]` returns **`True`** if the expression contains epsilon tensors that can be contracted with each other. The function is optimized for large expressions, i.e. it is not so good as a criterion in e.g. **`Select`**.

4.11.1 See also

Overview, Contract, EpsContract.

4.11.2 Examples

```
FCI[LC[p1, p2, p3, p4]]

EpsContractFreeQ[%]
```

$$\bar{\epsilon}^{\text{p1 p2 p3 p4}}$$

True

```
FCI[LC[p1, p2, p3, mu] LC[q1, q2, q3, q4]]

EpsContractFreeQ[%]
```

$$\bar{\epsilon}^{\text{p1 p2 p3 mu}}\,\bar{\epsilon}^{\text{q1 q2 q3 q4}}$$

False

4.12 EpsEvaluate

`EpsEvaluate[expr]` applies total antisymmetry and linearity (w.r.t. momenta) to all Levi-Civita tensors (**`Eps`**) in expr.

4.12.1 See also

Overview, Contract, Eps, LC, Trick.

4.12.2 Examples

```
Contract[LC[\[Mu], \[Nu], \[Rho], \[Sigma]] FV[p + q, \[Sigma]]] //
MomentumCombine

ex = EpsEvaluate[%]
```

$$\bar{\epsilon}^{\mu\nu\rho\overline{p}+\overline{q}}$$

$$\bar{\epsilon}^{\mu\nu\rho\overline{p}} + \bar{\epsilon}^{\mu\nu\rho\overline{q}}$$

```
ex // StandardForm

(*Eps[LorentzIndex[\[Mu]], LorentzIndex[\[Nu]], LorentzIndex[\[Rho]],
Momentum[p]] + Eps[LorentzIndex[\[Mu]], LorentzIndex[\[Nu]],
LorentzIndex[\[Rho]], Momentum[q]]*)
```

4.13 ExpandScalarProduct

`ExpandScalarProduct[expr]` expands scalar products of sums of momenta in **`expr`**.

`ExpandScalarProduct` does not use **`Expand`** on **`expr`**.

4.13.1 See also

Overview, Calc, MomentumExpand, MomentumCombine, FCVariable

4.13.2 Examples

```
SP[p1 + p2 + p3, p4 + p5 + p6]

% // ExpandScalarProduct
```

$$(\overline{\text{p1}} + \overline{\text{p2}} + \overline{\text{p3}}) \cdot (\overline{\text{p4}} + \overline{\text{p5}} + \overline{\text{p6}})$$

$$\overline{\text{p1}} \cdot \overline{\text{p4}} + \overline{\text{p1}} \cdot \overline{\text{p5}} + \overline{\text{p1}} \cdot \overline{\text{p6}} + \overline{\text{p2}} \cdot \overline{\text{p4}} + \overline{\text{p2}} \cdot \overline{\text{p5}} + \overline{\text{p2}} \cdot \overline{\text{p6}} + \overline{\text{p3}} \cdot \overline{\text{p4}} + \overline{\text{p3}} \cdot \overline{\text{p5}} + \overline{\text{p3}} \cdot \overline{\text{p6}}$$

```
SP[p, p - q]

ExpandScalarProduct[%]
```

$$\overline{p} \cdot (\overline{p} - \overline{q})$$

$$\overline{p}^2 - \overline{p} \cdot \overline{q}$$

```
FV[p - q, \[Mu]]

ExpandScalarProduct[%]
```

$$(\overline{p} - \overline{q})^{\mu}$$

$$\overline{p}^{\mu} - \overline{q}^{\mu}$$

```
SPD[p - q, q - r]

ExpandScalarProduct[%]
```

$$(p - q) \cdot (q - r)$$

$$p \cdot q - p \cdot r + q \cdot r - q^2$$

Using the option **`Momentum`** one can limit the expansion to particular momenta

```
SP[p1 + p2 + p3, p4 + p5 + p6]

ExpandScalarProduct[%, Momentum -> {p1}]
```

$$(\overline{\text{p1}} + \overline{\text{p2}} + \overline{\text{p3}}) \cdot (\overline{\text{p4}} + \overline{\text{p5}} + \overline{\text{p6}})$$

$$\overline{\text{p1}} \cdot (\overline{\text{p4}} + \overline{\text{p5}} + \overline{\text{p6}}) + (\overline{\text{p2}} + \overline{\text{p3}}) \cdot (\overline{\text{p4}} + \overline{\text{p5}} + \overline{\text{p6}})$$

By default **`ExpandScalarProduct`** does not apply linearity to Levi-Civita tensors

```
LC[\[Mu]][p1 + p2, p3 + p4, p5 + p6]

ExpandScalarProduct[%]
```

$$\bar{\epsilon}^{\mu \overline{p1}+\overline{p2}\,\overline{p3}+\overline{p4}\,\overline{p5}+\overline{p6}}$$

$$\bar{\epsilon}^{\mu \overline{p1}+\overline{p2}\,\overline{p3}+\overline{p4}\,\overline{p5}+\overline{p6}}$$

Using the option **EpsEvaluate** takes care of that

```
LC[\[Mu]][p1 + p2, p3 + p4, p5 + p6]

ExpandScalarProduct[%, EpsEvaluate -> True]
```

$$\bar{\epsilon}^{\mu \overline{p1}+\overline{p2}\,\overline{p3}+\overline{p4}\,\overline{p5}+\overline{p6}}$$

$$\bar{\epsilon}^{\mu \overline{p1}\,\overline{p3}\,\overline{p5}} + \bar{\epsilon}^{\mu \overline{p1}\,\overline{p3}\,\overline{p6}} + \bar{\epsilon}^{\mu \overline{p1}\,\overline{p4}\,\overline{p5}} + \bar{\epsilon}^{\mu \overline{p1}\,\overline{p4}\,\overline{p6}} + \bar{\epsilon}^{\mu \overline{p2}\,\overline{p3}\,\overline{p5}} + \bar{\epsilon}^{\mu \overline{p2}\,\overline{p3}\,\overline{p6}} + \bar{\epsilon}^{\mu \overline{p2}\,\overline{p4}\,\overline{p5}} + \bar{\epsilon}^{\mu \overline{p2}\,\overline{p4}\,\overline{p6}}$$

One can use the options **EpsEvaluate** and **Momentum** together

```
LC[\[Mu]][p1 + p2, p3 + p4, p5 + p6]

ExpandScalarProduct[%, EpsEvaluate -> True, Momentum -> {p1}]
```

$$\bar{\epsilon}^{\mu \overline{p1}+\overline{p2}\,\overline{p3}+\overline{p4}\,\overline{p5}+\overline{p6}}$$

$$\bar{\epsilon}^{\mu \overline{p1}\,\overline{p3}+\overline{p4}\,\overline{p5}+\overline{p6}} + \bar{\epsilon}^{\mu \overline{p2}\,\overline{p3}+\overline{p4}\,\overline{p5}+\overline{p6}}$$

Of course, the function is also applicable to Cartesian quantities

```
CSP[p1 + p2, p3 + p4]

ExpandScalarProduct[%]
```

$$(\overline{p1} + \overline{p2}) \cdot (\overline{p3} + \overline{p4})$$

$$\overline{p1} \cdot \overline{p3} + \overline{p1} \cdot \overline{p4} + \overline{p2} \cdot \overline{p3} + \overline{p2} \cdot \overline{p4}$$

```
CLC[][p1 + p2, p3 + p4, p5 + p6]

ExpandScalarProduct[%, EpsEvaluate -> True]
```

$$\bar{\epsilon}^{\overline{\text{p1}}+\overline{\text{p2}}\,\overline{\text{p3}}+\overline{\text{p4}}\,\overline{\text{p5}}+\overline{\text{p6}}}$$

$$\bar{\epsilon}^{\overline{\text{p1}}\,\overline{\text{p3}}\,\overline{\text{p5}}} + \bar{\epsilon}^{\overline{\text{p1}}\,\overline{\text{p3}}\,\overline{\text{p6}}} + \bar{\epsilon}^{\overline{\text{p1}}\,\overline{\text{p4}}\,\overline{\text{p5}}} + \bar{\epsilon}^{\overline{\text{p1}}\,\overline{\text{p4}}\,\overline{\text{p6}}} + \bar{\epsilon}^{\overline{\text{p2}}\,\overline{\text{p3}}\,\overline{\text{p5}}} + \bar{\epsilon}^{\overline{\text{p2}}\,\overline{\text{p3}}\,\overline{\text{p6}}} + \bar{\epsilon}^{\overline{\text{p2}}\,\overline{\text{p4}}\,\overline{\text{p5}}} + \bar{\epsilon}^{\overline{\text{p2}}\,\overline{\text{p4}}\,\overline{\text{p6}}}$$

Sometimes one would like to have external momenta multiplied by symbolic parameters in the propagators. In this case one should first declare the corresponding variables to be of **FCVariable** type

```
DataType[a, FCVariable] = True;
DataType[b, FCVariable] = True;
```

```
ExpandScalarProduct[SP[P, Q] /. P -> a P1 + b P2]

StandardForm[%]
```

$$a\left(\overline{\text{P1}}\cdot\overline{Q}\right) + b\left(\overline{\text{P2}}\cdot\overline{Q}\right)$$

```
(*a Pair[Momentum[P1], Momentum[Q]] + b Pair[Momentum[P2], Momentum[Q]]*)
```

4.14 FCCanonicalizeDummyIndices

FCCanonicalizeDummyIndices[expr] canonicalizes dummy indices in the expression.

Following index types are supported: **LorentzIndex**, **CartesianIndex**, **SUNIndex**, **SUNFIndex**, **DiracIndex**, **PauliIndex**

In the case of Lorentz indices the option **Momentum** provides a possibility to limit the canonicalization only to particular **Momenta**. The option **LorentzIndexNames** can be used to assign specific names to the canonicalized indices, to have say μ, ν, ρ etc. instead of some random names.

For other index types the corresponding options are called **CartesianIndexNames**, **SUNIndexNames**, **SUNFIndexNames**, **DiracIndexNames** and **PauliIndexNames**.

4.14.1 See also

Overview, FCRenameDummyIndices.

4.14.2 Examples

Canonicalization of Lorentz indices

```
FVD[q, mu] FVD[p, mu] + FVD[q, nu] FVD[p, nu] + FVD[q, si] FVD[r, si]

FCCanonicalizeDummyIndices[%] // Factor2
```

$$p^{\text{mu}} q^{\text{mu}} + p^{\text{nu}} q^{\text{nu}} + q^{\text{si}} r^{\text{si}}$$

$$q^{\text{FCGV(li191)}} \left(2 p^{\text{FCGV(li191)}} + r^{\text{FCGV(li191)}}\right)$$

```
Uncontract[SPD[q, p]^2, q, p, Pair -> All]

FCCanonicalizeDummyIndices[%, LorentzIndexNames -> {\[Mu], \[Nu]}]
```

$$p^{\$\text{AL}(\$28)} p^{\$\text{AL}(\$29)} q^{\$\text{AL}(\$28)} q^{\$\text{AL}(\$29)}$$

$$p^{\mu} p^{\nu} q^{\mu} q^{\nu}$$

Canonicalization of Cartesian indices

```
CVD[p, i] CVD[q, i] + CVD[p, j] CVD[r, j]

FCCanonicalizeDummyIndices[%] // Factor2
```

$$p^{i} q^{i} + p^{j} r^{j}$$

$$p^{\text{FCGV(ci391)}} \left(q^{\text{FCGV(ci391)}} + r^{\text{FCGV(ci391)}}\right)$$

```
CVD[p, i] CVD[q, i] + CVD[p, j] CVD[r, j]

FCCanonicalizeDummyIndices[%, CartesianIndexNames -> {a}] // Factor2
```

$$p^{i} q^{i} + p^{j} r^{j}$$

$$p^{a} \left(q^{a} + r^{a}\right)$$

Canonicalization of color indices

```
SUNT[a, b, a] + SUNT[c, b, c]
FCCanonicalizeDummyIndices[%]
```

$$T^a.T^b.T^a + T^c.T^b.T^c$$

$$2T^{\text{FCGV(sun601)}}.T^b.T^{\text{FCGV(sun601)}}$$

```
SUNT[a, b, a] + SUNT[c, b, c]
FCCanonicalizeDummyIndices[%, SUNIndexNames -> {u}]
```

$$T^a.T^b.T^a + T^c.T^b.T^c$$

$$2T^u.T^b.T^u$$

Canonicalization of Dirac indices

```
DCHN[GA[mu], i, j] DCHN[GA[nu], j, k]
FCCanonicalizeDummyIndices[%]
```

$$(\bar{\gamma}^{\text{mu}})_{ij}\,(\bar{\gamma}^{\text{nu}})_{jk}$$

$$(\bar{\gamma}^{\text{mu}})_{i\text{FCGV(di771)}}\,(\bar{\gamma}^{\text{nu}})_{\text{FCGV(di771)}k}$$

```
DCHN[GA[mu], i, j] DCHN[GA[nu], j, k]
FCCanonicalizeDummyIndices[%, DiracIndexNames -> {a}]
```

$$(\bar{\gamma}^{\text{mu}})_{ij}\,(\bar{\gamma}^{\text{nu}})_{jk}$$

$$(\bar{\gamma}^{\text{mu}})_{ia}\,(\bar{\gamma}^{\text{nu}})_{ak}$$

Canonicalization of Pauli indices

```
PCHN[CSI[a], i, j] PCHN[CSI[b], j, k]

FCCanonicalizeDummyIndices[%]
```

$$\left(\overline{\sigma}^{a}\right)_{ij}\left(\overline{\sigma}^{b}\right)_{jk}$$

$$\left(\overline{\sigma}^{a}\right)_{i\text{FCGV(pi921)}}\left(\overline{\sigma}^{b}\right)_{\text{FCGV(pi921)}k}$$

```
PCHN[CSI[a], i, j] PCHN[CSI[b], j, k]

FCCanonicalizeDummyIndices[%, PauliIndexNames -> {l}]
```

$$\left(\overline{\sigma}^{a}\right)_{ij}\left(\overline{\sigma}^{b}\right)_{jk}$$

$$\left(\overline{\sigma}^{a}\right)_{il}\left(\overline{\sigma}^{b}\right)_{lk}$$

Using the option **Head** one can specify which index heads should be canonicalized, while the rest will be ignored.

```
(QuantumField[Superscript[\[Phi], "+"], PauliIndex[k1], PauliIndex[k2],
     R, r] . QuantumField[FCPartialD[{CartesianIndex[i], r}],
     FCPartialD[{CartesianIndex[i], r}], \[Phi], PauliIndex[k2],
PauliIndex[k1], R, r])

FCCanonicalizeDummyIndices[%, CartesianIndexNames -> {j}, Head ->
{CartesianIndex}]
```

$$\phi^{+\text{k1 k2}Rr}\cdot\left(\partial_{\{i,r\}}\partial_{\{i,r\}}\phi^{\text{k2 k1}Rr}\right)$$

$$\phi^{+\text{k1 k2}Rr}\cdot\left(\partial_{\{j,r\}}\partial_{\{j,r\}}\phi^{\text{k2 k1}Rr}\right)$$

4.15 FCClearScalarProducts

FCClearScalarProducts[] removes all user-performed specific settings for ScalarProduct's.

4.15.1 See also

Overview, ScalarProduct, Pair, SP, SPD.

4.15.2 Examples

```
ScalarProduct[p, p] = m^2;

Pair[Momentum[p], Momentum[p]]
```

$$m^2$$

```
FCClearScalarProducts[]

Pair[Momentum[p], Momentum[p]]

SP[p, p]
```

$$\overline{p}^2$$

$$\overline{p}^2$$

4.16 FCGetDimensions

`FCGetDimensions[expr]` is an auxiliary function that determines the dimensions in which 4-momenta and Dirac matrices of the given expression are defined. The result is returned as a list, e.g. **`{4}`**, **`{D}`** or **`{4, D, D-4}`** etc.

This is useful if one wants to be sure that all quantities inside a particular expression are purely 4-dimensional or purely D-dimensional.

4.16.1 See also

Overview, ChangeDimension.

4.16.2 Examples

```
FCGetDimensions[GA[i]]
```

$$\{4\}$$

```
FCGetDimensions[GSD[p]]
```

$$\{D\}$$

```
FCGetDimensions[FVE[q, \[Mu]] GS[p]]
```

$$\{4, D-4\}$$

4.17 FCGetDummyIndices

`FCGetDummyIndices[exp, {head1, head2, ...}]` returns the list of dummy indices from heads **`head1`**, **`head2`**, ...

As always in FeynCalc, Einstein summation convention is implicitly assumed.

4.17.1 See also

Overview, FCRenameDummyIndices, Contract, DummyIndexFreeQ, FCGetFreeIndices, FreeIndexFreeQ.

4.17.2 Examples

```
FCI[FV[p, \[Mu]] FV[q, \[Nu]]]

FCGetDummyIndices[%, {LorentzIndex}]
```

$$\overline{p}^{\mu}\overline{q}^{\nu}$$

$$\{\}$$

```
FCI[FV[p, \[Mu]] FV[q, \[Mu]]]

FCGetDummyIndices[%, {LorentzIndex}]
```

$$\overline{p}^{\mu}\overline{q}^{\mu}$$

$$\{\mu\}$$

```
FCI[SUNT[a, b]]

FCGetDummyIndices[%, {SUNIndex}]
```

$$T^a.T^b$$

$$\{\}$$

```
FCI[SUNT[a, a]]

FCGetDummyIndices[%, {SUNIndex}]
```

$$T^a.T^a$$

$$\{a\}$$

4.18 FCGetFreeIndices

`FCGetFreeIndices[exp, {head1, head2, ...}]` returns the list of free (uncontracted) indices from heads **`head1`**, **`head2`**, ...

As always in FeynCalc, Einstein summation convention is implicitly assumed. The function is optimized for large expressions, i.e. it is not so good as a criterion in e.g. **`Select`**.

If it is understood that each term in the expression contains the same number of free indices, setting the option **`First`** to **`True`** can considerably speed up the evaluation.

4.18.1 See also

Overview, FCRenameDummyIndices, Contract, FCGetDummyIndices, DummyIndexFreeQ, FreeIndexFreeQ.

4.18.2 Examples

```
FCI[FV[p, \[Mu]] FV[q, \[Nu]]]
FCGetFreeIndices[%, {LorentzIndex}]
```

$$\overline{p}^{\mu}\overline{q}^{\nu}$$

$$\{\mu, \nu\}$$

```
FCI[FV[p, \[Mu]] FV[q, \[Mu]]]
FCGetFreeIndices[%, {LorentzIndex}]
```

$$\overline{p}^{\mu}\overline{q}^{\mu}$$

$$\{\}$$

```
FCI[SUNT[a, b]]
FCGetFreeIndices[%, {SUNIndex}]
```

$$T^{a}.T^{b}$$

$$\{a, b\}$$

```
FCI[SUNT[a, a]]
FCGetFreeIndices[%, {SUNIndex}]
```

$$T^{a}.T^{a}$$

$$\{\}$$

4.19 FCGetScalarProducts

FCGetScalarProducts[{p1, p2, ...}] returns all scalar products involving external momenta **p1, p2, ...** that were set using down values.

Using the option **SetDimensions** one can specify the dimensions of scalar products one is interested in.

4.19.1 See also

Overview, ScalarProduct, Pair, SP, SPD.

4.19.2 Examples

```
FCClearScalarProducts[];
SP[p1] = m1^2;
SP[p1] = m2^2;
SP[p1, p2] = s;
SPD[q1] = M1^2;
SPD[q2] = M2^2;
SPD[q1, q2] = t;
```

```
FCGetScalarProducts[{p1, p2, q1, q2}]
```

$$\{\text{Hold[Pair]}\,[\overline{\text{p1}},\overline{\text{p1}}] \to \text{m2}^2, \text{Hold[Pair]}\,[\overline{\text{p1}},\overline{\text{p2}}] \to s, \text{Hold[Pair][q1, q1]} \to \text{M1}^2, \text{Hold[Pair][q2, q2]} \to \text{M2}^2, \text{Hold[Pair][q1, q2]} \to t\}$$

```
FCGetScalarProducts[{p1, p2, q1, q2}, SetDimensions -> {4}]
```

$$\{\text{Hold[Pair]}\,[\overline{\text{p1}},\overline{\text{p1}}] \to \text{m2}^2, \text{Hold[Pair]}\,[\overline{\text{p1}},\overline{\text{p2}}] \to s\}$$

```
FCGetScalarProducts[{p1, p2, q1, q2}, SetDimensions -> {D}]
```

$$\{\text{Hold[Pair][q1, q1]} \to \text{M1}^2, \text{Hold[Pair][q2, q2]} \to \text{M2}^2, \text{Hold[Pair][q1, q2]} \to t\}$$

4.20 FCPermuteMomentaRules

FCPermuteMomentaRules[{p1, p2, ...}] returns a set of rules that contain all possible permutations of the momenta **p1**, **p2**, ... This can be useful when working with amplitudes that exhibit a symmetry in some or all of the final state momenta or when trying to find mappings between loop integrals from different topologies.

4.20.1 See also

Overview, FCReplaceMomenta.

4.20.2 Examples

```
FCPermuteMomentaRules[{p1, p2}]

f[p1, p2] /. %
```

$$\{\{\},\{\text{p1} \to \text{p2}, \text{p2} \to \text{p1}\}\}$$

$$\{f(\text{p1},\text{p2}), f(\text{p2},\text{p1})\}$$

```
FCPermuteMomentaRules[{p1, p2, p3}]

f[p1, p2, p3] /. %
```

$$\{\{\},\{\text{p1} \to \text{p2}, \text{p2} \to \text{p1}\},\{\text{p1} \to \text{p3}, \text{p3} \to \text{p1}\},\{\text{p2} \to \text{p3}, \text{p3} \to \text{p2}\},\\ \{\text{p1} \to \text{p2}, \text{p2} \to \text{p3}, \text{p3} \to \text{p1}\},\{\text{p1} \to \text{p3}, \text{p2} \to \text{p1}, \text{p3} \to \text{p2}\}\}$$

$$\{f(\text{p1},\text{p2},\text{p3}), f(\text{p2},\text{p1},\text{p3}), f(\text{p3},\text{p2},\text{p1}), f(\text{p1},\text{p3},\text{p2}), f(\text{p2},\text{p3},\text{p1}), f(\text{p3},\text{p1},\text{p2})\}$$

4.21 FCRenameDummyIndices

`FCRenameDummyIndices[expr]` identifies dummy indices and changes their names pairwise to random symbols. This can be useful if you have an expression that contains dummy indices and want to compute the square of it. For example, the square of **`GA[a, l, a]`** equals 16. However, if you forget to rename the dummy indices and compute **`GA[a, l, a, a, l, a]`** instead of **`GA[a, l, a, b, l, b]`**, you will get 64.

Notice that this routine does not perform any canonicalization. Use **`FCCanonicalizeDummyIndices`** for that.

4.21.1 See also

Overview, ComplexConjugate, FCCanonicalizeDummyIndices.

4.21.2 Examples

```
FVD[q, mu] FVD[p, mu] + FVD[q, nu] FVD[p, nu] + FVD[q, si] FVD[r, si]

FCRenameDummyIndices[%] // Factor2
```

$$p^{\text{mu}} q^{\text{mu}} + p^{\text{nu}} q^{\text{nu}} + q^{\text{si}} r^{\text{si}}$$

$$p^{\$\text{AL}(\$19)} q^{\$\text{AL}(\$19)} + p^{\$\text{AL}(\$20)} q^{\$\text{AL}(\$20)} + q^{\$\text{AL}(\$21)} r^{\$\text{AL}(\$21)}$$

```
Uncontract[SPD[q, p]^2, q, p, Pair -> All]

FCRenameDummyIndices[%]
```

$$p^{\$\text{AL}(\$22)} p^{\$\text{AL}(\$23)} q^{\$\text{AL}(\$22)} q^{\$\text{AL}(\$23)}$$

$$p^{\$\text{AL}(\$24)} p^{\$\text{AL}(\$25)} q^{\$\text{AL}(\$24)} q^{\$\text{AL}(\$25)}$$

```
amp = -(Spinor[Momentum[k1], SMP["m_mu"], 1] . GA[Lor1] .
Spinor[-Momentum[k2],
         SMP["m_mu"], 1]*Spinor[-Momentum[p2], SMP["m_e"], 1] . GA[Lor1] .
Spinor[Momentum[p1],
         SMP["m_e"], 1]*FAD[k1 + k2, Dimension -> 4]*SMP["e"]^2);

amp // FCRenameDummyIndices
```

$$-\frac{\text{e}^2 \left(\varphi(-\overline{\text{p2}}, m_e)\right) .\bar{\gamma}^{\$\text{AL}(\$26)} . \left(\varphi(\overline{\text{p1}}, m_e)\right) \left(\varphi(\overline{\text{k1}}, m_\mu)\right) .\bar{\gamma}^{\$\text{AL}(\$26)} . \left(\varphi(-\overline{\text{k2}}, m_\mu)\right)}{(\overline{\text{k1}} + \overline{\text{k2}})^2}$$

```
CVD[p, i] CVD[q, i] + CVD[p, j] CVD[r, j]

% // FCRenameDummyIndices
```

$$p^i q^i + p^j r^j$$

$$p^{\$\text{AL}(\$27)} q^{\$\text{AL}(\$27)} + p^{\$\text{AL}(\$28)} r^{\$\text{AL}(\$28)}$$

```
SUNT[a, b, a] + SUNT[c, b, c]

% // FCRenameDummyIndices
```

$$T^a.T^b.T^a + T^c.T^b.T^c$$

$$T^{\$AL(\$29)}.T^b.T^{\$AL(\$29)} + T^{\$AL(\$30)}.T^b.T^{\$AL(\$30)}$$

```
DCHN[GA[mu], i, j] DCHN[GA[nu], j, k]

% // FCRenameDummyIndices
```

$$(\bar{\gamma}^{\text{mu}})_{ij} (\bar{\gamma}^{\text{nu}})_{jk}$$

$$(\bar{\gamma}^{\text{mu}})_{i\$AL(\$31)} (\bar{\gamma}^{\text{nu}})_{\$AL(\$31)k}$$

```
PCHN[CSI[a], i, j] PCHN[CSI[b], j, k]

% // FCRenameDummyIndices
```

$$(\overline{\sigma}^a)_{ij} (\overline{\sigma}^b)_{jk}$$

$$(\overline{\sigma}^a)_{i\$AL(\$32)} (\overline{\sigma}^b)_{\$AL(\$32)k}$$

4.22 FCReplaceMomenta

`FCReplaceMomenta[exp, rule]` replaces the given momentum according to the specified replacement rules. Various options can be used to customize the replacement procedure.

4.22.1 See also

Overview, FCPermuteMomentaRules.

4.22.2 Examples

```
amp = (-I)*Spinor[-Momentum[l2], ME, 1] . GA[\[Mu]] . Spinor[Momentum[l1],
ME, 1]*
   Spinor[Momentum[p1], SMP["m_Q"], 1] . GS[Polarization[kp, -I,
      Transversality -> True]] . (GS[kp + p1] + SMP["m_Q"]) . GA[\[Mu]] .
Spinor[-Momentum[p2],
     SMP["m_Q"], 1]*FAD[kp + p1 + p2, Dimension -> 4]*FAD[{-l1 - l2 - p2,
SMP["m_Q"]},
    Dimension -> 4]*SDF[cq, cqbar]*SMP["e"]^3*SMP["Q_u"]^2
```

$$-\frac{i\,\mathrm{e}^3 Q_u^2 \delta_{\mathrm{cq\ cqbar}} \left(\varphi(-\overline{\mathrm{l2}}, \mathrm{ME})\right).\bar{\gamma}^{\mu}.\left(\varphi(\overline{\mathrm{l1}}, \mathrm{ME})\right)\left(\varphi(\overline{\mathrm{p1}}, m_Q)\right).(\bar{\gamma}\cdot\bar{\varepsilon}^*(\mathrm{kp})).\left(\bar{\gamma}\cdot\left(\overline{\mathrm{kp}}+\overline{\mathrm{p1}}\right)+m_Q\right).\bar{\gamma}^{\mu}.\left(\varphi(-\overline{\mathrm{p2}}, m_Q)\right)}{(\overline{\mathrm{kp}}+\overline{\mathrm{p1}}+\overline{\mathrm{p2}})^2\left((-\overline{\mathrm{l1}}-\overline{\mathrm{l2}}-\overline{\mathrm{p2}})^2-m_Q^2\right)}$$

```
FCReplaceMomenta[amp, {p1 -> P + 1/2 q, p2 -> P - 1/2 q}]

ClearAll[amp]
```

$$-\frac{i\,\mathrm{e}^3 Q_u^2 \delta_{\mathrm{cq\ cqbar}} \left(\varphi(-\overline{\mathrm{l2}}, \mathrm{ME})\right).\bar{\gamma}^{\mu}.\left(\varphi(\overline{\mathrm{l1}}, \mathrm{ME})\right)\left(\varphi(\overline{\mathrm{p1}}, m_Q)\right).(\bar{\gamma}\cdot\bar{\varepsilon}^*(\mathrm{kp})).\left(\bar{\gamma}\cdot\overline{\mathrm{kp}}+\bar{\gamma}\cdot\left(\overline{P}+\frac{\overline{q}}{2}\right)+m_Q\right).\bar{\gamma}^{\mu}.\left(\varphi(-\overline{\mathrm{p2}},\right.}{(\overline{\mathrm{kp}}+2\overline{P})^2\left((-\overline{\mathrm{l1}}-\overline{\mathrm{l2}}-\overline{P}+\frac{\overline{q}}{2})^2-m_Q^2\right)}$$

Notice that **FCReplaceMomenta** is not suitable for expanding in 4-momenta (soft limits etc.) as it does not check for cases where a particular substitution yields a singularity. For example, the following code obviously returns a nonsensical result

```
FCClearScalarProducts[];

SPD[q] = 0;

FCReplaceMomenta[FAD[q + p], {p -> 0}]

FCClearScalarProducts[];
```

$$\frac{1}{0}$$

FCReplaceMomenta equally works with **FCTopology** objects. There it is actually the preferred way to perform momentum shifts. Consider e.g.

```
ex = FCTopology[topo, {SFAD[{{p1, 0}, {0, 1}, 1}], SFAD[{{p2 + p3, 0}, {0,
1}, 1}],
    SFAD[{{p2 - Q, 0}, {0, 1}, 1}], SFAD[{{p1 + p3 - Q, 0}, {0, 1}, 1}]},
{p1, p2, p3}, {Q}, {}, {}]
```

$$\text{FCTopology}\left(\text{topo}, \left\{\frac{1}{(\text{p1}^2+i\eta)}, \frac{1}{((\text{p2}+\text{p3})^2+i\eta)}, \frac{1}{((\text{p2}-Q)^2+i\eta)}, \frac{1}{((\text{p1}+\text{p3}-Q)^2+i\eta)}\right\}, \{\text{p1}, \text{p2}, \text{p3}\}, \{Q\}, \{\}, \{\}\right)$$

where we want to shift **p_2** to **p_2 + Q**. Doing so naively messes us the topology by invalidating the list of loop momenta

```
ex /. p2 -> p2 + Q

FCLoopValidTopologyQ[%]
```

$$\text{FCTopology}\left(\text{topo}, \left\{\frac{1}{(\text{p1}^2+i\eta)}, \frac{1}{((\text{p2}+\text{p3}+Q)^2+i\eta)}, \frac{1}{(\text{p2}^2+i\eta)}, \frac{1}{((\text{p1}+\text{p3}-Q)^2+i\eta)}\right\}, \{\text{p1}, \text{p2}+Q, \text{p3}\}, \{Q\}, \{\}, \{\}\right)$$

FCLoopValidTopologyQ: Topology validation failed: FCTopology[topo, {FeynAmpDenominator[StandardPropagatorDenominator[Momentum[p1, D], 0, 0, {1, 1}]], FeynAmpDenominator[Standar ... r[Momentum[p1, D] + Momentum[p3, D] - Momentum[Q, D], 0, 0, {1, 1}]]}, {p1, p2 + Q, p3}, {Q}, {}, {}] is not a proper topology.

False

Using **FCReplaceMomenta** we immediately get we want

```
FCReplaceMomenta[ex, {p2 -> p2 + Q}]
```

$$\text{FCTopology}\left(\text{topo}, \left\{\frac{1}{(\text{p1}^2+i\eta)}, \frac{1}{((\text{p2}+\text{p3}+Q)^2+i\eta)}, \frac{1}{(\text{p2}^2+i\eta)}, \frac{1}{((\text{p1}+\text{p3}-Q)^2+i\eta)}\right\}, \{\text{p1}, \text{p2}, \text{p3}\}, \{Q\}, \{\}, \{\}\right)$$

4.23 FCReplaceD

FCReplaceD[expr, rule] replaces **D** in expr according to the supplied replacement rule (e.g. **D -> 4 - 2*Epsilon**) but doesn't touch **D** inside **Pair**s and **DiracGamma**s, i.e the dimension of scalar products, metric tensors and Dirac matrices is unchanged. The latter can and should be done via **ChangeDimension**.

4.23.1 See also

Overview.

4.23.2 Examples

Applying the replacement rule directly to the expression doesn't give the desired result

```
FCI[D MTD[\[Mu], \[Nu]]]

% /. D -> 4 - 2 Epsilon
```

$$D g^{\mu\nu}$$

$$(4-2\varepsilon) g_{\{4-2\varepsilon,4-2\varepsilon\}}{}^{\mu\nu}$$

With **FCReplaceD** we get what we want

```
FCReplaceD[D MTD[\[Mu], \[Nu]], D -> 4 - 2 Epsilon]

ChangeDimension[%, 4]
```

$$(4-2\varepsilon) g^{\mu\nu}$$

$$(4-2\varepsilon) \bar{g}^{\mu\nu}$$

4.24 FCRerouteMomenta

FCRerouteMomenta[exp, {p1, p2, ...}, {k1, k2, ...}] changes the routing of the momenta by exploiting the 4-momentum conservation law $p_1 + p_2 + \ldots = k_1 + k_2 + \ldots$.

The main aim of this function is to simplify the input expression by replacing simple linear combinations of the external momenta with shorter expressions.

For example, in a process $a(p_1) + b(p_2) -> c(k_1) + d(k_2) + e(k_3)$, the combination $k_1 + k_2 - p_2$ can be replaced with the shorter expression $p_1 - k_3$.

The replacements are applied using the **FeynCalcExternal** form of the expression. Ideally, this function should be used directly on the output of a diagram generator such as FeynArts or QGRAF.

4.24.1 See also

Overview.

4.24.2 Examples

Reroute momenta according to the momentum conservation relation $l_1 + l_2 = p_1 + p_2 + k_p$.

```
exp = (-I)*Spinor[-Momentum[l2], ME, 1] . GA[\[Mu]] . Spinor[Momentum[l1],
ME,
     1]*Spinor[Momentum[p1], SMP["m_Q"], 1] . GS[Polarization[kp, -I,
      Transversality -> True]] . (GS[kp + p1] + SMP["m_Q"]) . GA[\[Mu]] .
Spinor[-Momentum[p2],
     SMP["m_Q"], 1]*FAD[kp + p1 + p2, Dimension -> 4]*FAD[{-l1 - l2 - p2,
SMP["m_Q"]},
    Dimension -> 4]*SDF[cq, cqbar]*SMP["e"]^3*SMP["Q_u"]^2
```

$$-\frac{i\,\mathrm{e}^3 Q_u^2 \delta_{\mathrm{cq\ cqbar}} \left(\varphi(-\overline{\mathrm{l2}}, \mathrm{ME})\right).\bar{\gamma}^{\mu}.\left(\varphi(\overline{\mathrm{l1}}, \mathrm{ME})\right)\left(\varphi(\overline{\mathrm{p1}}, m_Q)\right).(\bar{\gamma}\cdot\bar{\varepsilon}^*(\mathrm{kp})).\left(\bar{\gamma}\cdot\left(\overline{\mathrm{kp}}+\overline{\mathrm{p1}}\right)+m_Q\right).\bar{\gamma}^{\mu}.\left(\varphi(-\overline{\mathrm{p2}}, m_Q)\right)}{(\overline{\mathrm{kp}}+\overline{\mathrm{p1}}+\overline{\mathrm{p2}})^2\left((-\overline{\mathrm{l1}}-\overline{\mathrm{l2}}-\overline{\mathrm{p2}})^2-m_Q^2\right)}$$

```
FCRerouteMomenta[exp, {l1, l2}, {p1, p2, kp}]
```

$$-\frac{i\,\mathrm{e}^3 Q_u^2 \delta_{\mathrm{cq\ cqbar}} \left(\varphi(-\overline{\mathrm{l2}}, \mathrm{ME})\right).\bar{\gamma}^{\mu}.\left(\varphi(\overline{\mathrm{l1}}, \mathrm{ME})\right)\left(\varphi(\overline{\mathrm{p1}}, m_Q)\right).(\bar{\gamma}\cdot\bar{\varepsilon}^*(\mathrm{kp})).\left(\bar{\gamma}\cdot\left(\overline{\mathrm{kp}}+\overline{\mathrm{p1}}\right)+m_Q\right).\bar{\gamma}^{\mu}.\left(\varphi(-\overline{\mathrm{p2}}, m_Q)\right)}{(\overline{\mathrm{l1}}+\overline{\mathrm{l2}})^2\left((-\overline{\mathrm{l1}}-\overline{\mathrm{l2}}-\overline{\mathrm{p2}})^2-m_Q^2\right)}$$

4.25 FCSchoutenBruteForce

`FCSchoutenBruteForce[exp, {}, {}]` can be used to show that certain terms are zero by repeatedly applying, Schouten's identity in a brute force way.

The algorithm tries to find replacements which follow from the Schouten's identity and make the length of the given expression shorter.

It is not guaranteed to terminate and in general can often get stuck. Still, with some luck it is often possible to show that certain terms vanish by a sequence of transformations that would be otherwise very difficult to find.

4.25.1 See also

Overview, Schouten.

4.25.2 Examples

One may not recognize it easily, but the following expression is zero by Schouten's identity

```
FCClearScalarProducts[]

exp = LC[][p1, p2, p3, p4] SP[p5, p6] + LC[][p2, p3, p4, p5] SP[p1, p6] +
   LC[][p3, p4, p5, p1] SP[p2, p6] + LC[][p4, p5, p1, p2] SP[p3, p6] -
   LC[][p1, p2, p3, p5] SP[p4, p6]
```

$$(\overline{p5}\cdot\overline{p6})\,\bar{\epsilon}^{\overline{p1}\,\overline{p2}\,\overline{p3}\,\overline{p4}} - (\overline{p4}\cdot\overline{p6})\,\bar{\epsilon}^{\overline{p1}\,\overline{p2}\,\overline{p3}\,\overline{p5}} + (\overline{p1}\cdot\overline{p6})\,\bar{\epsilon}^{\overline{p2}\,\overline{p3}\,\overline{p4}\,\overline{p5}} + (\overline{p2}\cdot\overline{p6})\,\bar{\epsilon}^{\overline{p3}\,\overline{p4}\,\overline{p5}\,\overline{p1}} + (\overline{p3}\cdot\overline{p6})\,\bar{\epsilon}^{\overline{p4}\,\overline{p5}\,\overline{p1}\,\overline{p2}}$$

```
FCSchoutenBruteForce[exp, {}, {}]
```

FCSchoutenBruteForce: The following rule was applied: $\bar{\epsilon}^{\overline{p2}\,\overline{p3}\,\overline{p4}\,\overline{p5}}\,(\overline{p1}\cdot\overline{p6})$:
$\to \bar{\epsilon}^{\overline{p1}\,\overline{p3}\,\overline{p4}\,\overline{p5}}\,(\overline{p2}\cdot\overline{p6}) - \bar{\epsilon}^{\overline{p1}\,\overline{p2}\,\overline{p4}\,\overline{p5}}\,(\overline{p3}\cdot\overline{p6}) + \bar{\epsilon}^{\overline{p1}\,\overline{p2}\,\overline{p3}\,\overline{p5}}\,(\overline{p4}\cdot\overline{p6}) - \bar{\epsilon}^{\overline{p1}\,\overline{p2}\,\overline{p3}\,\overline{p4}}\,(\overline{p5}\cdot\overline{p6})$

FCSchoutenBruteForce: The numbers of terms in the expression decreased by: 5

FCSchoutenBruteForce: Current length of the expression: 0

0

4.26 FCSetScalarProducts

`FCSetScalarProducts[]` assigns values in the second list to scalar products (or other kinematic-related symbols such as **`Momentum`**, **`CartesianMomentum`**, **`TC`** etc.) in the first list.

The values can be also modified if the quantities in the first list are entered by hand. To modify the definitions programmatically without resorting to **`With`** and similar delayed evaluation tricks one can use placeholders in conjunction with the **`InitialSubstitutions`** option.

4.26.1 See also

Overview, ScalarProduct.

4.26.2 Examples

```
FCClearScalarProducts[];

FCSetScalarProducts[{SPD[p1], SPD[p2], SPD[p3, p4]}, {0, xx1, xx2}];
```

```
{SPD[p1], SPD[p2], SPD[p3, p4]}
```

$$\{0, xx1, xx2\}$$

```
FCSetScalarProducts[{spd[p1]}, {val}, InitialSubstitutions -> {spd -> SPD}]
```

{val}

4.27 ScalarProduct

ScalarProduct[p, q] is the input for the scalar product of two Lorentz vectors p and q.

ScalarProduct[p] is equivalent to ScalarProduct[p, p].

Expansion of sums of momenta in **ScalarProduct** is done with **ExpandScalarProduct**.

Scalar products may be set, e.g. via **ScalarProduct[a, b] = m^2**; but **a** and **b** may not contain sums.

ScalarProduct[a] corresponds to **ScalarProduct[a, a]**

Note that **ScalarProduct[a, b] = m^2** actually sets Lorentzian scalar products in different dimensions specified by the value of the **SetDimensions** option.

It is highly recommended to set **ScalarProduct**s before any calculation. This improves the performance of FeynCalc.

4.27.1 See also

Overview, Calc, FCClearScalarProducts, ExpandScalarProduct, ScalarProductCancel, Pair, SP, SPD.

4.27.2 Examples

```
ScalarProduct[p, q]
```

$$\overline{p} \cdot \overline{q}$$

```
ScalarProduct[p + q, -q]
```

$$-(\overline{q} \cdot (\overline{p} + \overline{q}))$$

```
ScalarProduct[p, p]
```

$$\overline{p}^2$$

```
ScalarProduct[q]
```

$$\overline{q}^2$$

```
ScalarProduct[p, q] // StandardForm

(*Pair[Momentum[p], Momentum[q]]*)
```

```
ScalarProduct[p, q, Dimension -> D] // StandardForm

(*Pair[Momentum[p, D], Momentum[q, D]]*)
```

```
ScalarProduct[Subscript[p, 1], Subscript[p, 2]] = s/2
```

$$\frac{s}{2}$$

```
ExpandScalarProduct[ ScalarProduct[Subscript[p, 1] - q, Subscript[p, 2] -
k]]
```

$$-\overline{k}\cdot\overline{p}_1+\overline{k}\cdot\overline{q}-\overline{q}\cdot\overline{p}_2+\frac{s}{2}$$

```
Calc[ ScalarProduct[Subscript[p, 1] - q, Subscript[p, 2] - k]]
```

$$-\overline{k}\cdot\overline{p}_1+\overline{k}\cdot\overline{q}-\overline{q}\cdot\overline{p}_2+\frac{s}{2}$$

```
ScalarProduct[q1] = qq;
```

```
SP[q1]
```

qq

```
FCClearScalarProducts[]
```

4.28 FCSetMetricSignature

FCSetMetricSignature sets the signature of the Minkowski metric used when working with Cartesian objects, like **CartesianPair**, **CartesianIndex**, **CartesianMomentum** etc.

The default choice is $(1, -1, -1, -1)$ which corresponds to **FCSetMetricSignature[{1, -1}]**.

4.28.1 See also

Overview, FCGetMetricSignature.

4.28.2 Examples

```
FCSetMetricSignature[{-1, 1}]

SPD[p, q] // LorentzToCartesian
```

$$p \cdot q - p^0 q^0$$

```
FCSetMetricSignature[{1, -1}]

SPD[p, q] // LorentzToCartesian
```

$$p^0 q^0 - p \cdot q$$

4.29 FCGetMetricSignature

FCGetMetricSignature[] returns the signature of the Minkowski metric used when working with Cartesian objects, such as **CartesianPair**, **CartesianIndex**, **CartesianMomentum** etc.

{1, -1} corresponds to $(1, -1, -1, -1)$ and **{-1, 1}** means $(-1, 1, 1, 1)$.

4.29.1 See also

Overview, FCSetMetricSignature.

4.29.2 Examples

```
FCGetMetricSignature[]
```

$$\{1, -1\}$$

4.30 FourDivergence

FourDivergence[exp, FV[p, mu]] calculates the partial derivative of exp w.r.t p^μ. **FourDivergence[exp, FV[p, mu], FV[p, nu], ...]** gives the multiple derivative.

4.30.1 See also

Overview, ThreeDivergence.

4.30.2 Examples

```
SP[p, q]

FourDivergence[%, FV[q, \[Mu]]]
```

$$\overline{p} \cdot \overline{q}$$

$$\overline{p}^\mu$$

```
SP[p - k, q]

FourDivergence[%, FV[k, \[Mu]]]
```

$$(\overline{p} - \overline{k}) \cdot \overline{q}$$

$$-\overline{q}^\mu$$

```
SFAD[{p, m^2}]

FourDivergence[%, FVD[p, \[Nu]]]
```

$$\frac{1}{(p^2 - m^2 + i\eta)}$$

$$-\frac{2p^\nu}{(p^2 - m^2 + i\eta)^2}$$

```
FVD[l, \[Mu]] FAD[{l, 0}, {l - p, 0}]

FourDivergence[%, FVD[l, \[Mu]]]
```

$$\frac{l^{\mu}}{l^2.(l-p)^2}$$

$$\frac{D}{l^2.(l-p)^2}-\frac{2l^2}{(l^2)^2.(l-p)^2}+\frac{2(l\cdot p)-2l^2}{l^2.(l-p)^4}$$

```
SP[p, w]*SpinorUBar[p2, m] . GS[w] . SpinorU[p1, m]

FourDivergence[%, FV[w, a]]
```

$$(\overline{p}\cdot\overline{w})\,\bar{u}(\mathrm{p2},m).(\bar{\gamma}\cdot\overline{w})\,.u(\mathrm{p1},m)$$

$$(\overline{p}\cdot\overline{w})\left(\varphi(\overline{\mathrm{p2}},m)\right).\bar{\gamma}^{a}.\left(\varphi(\overline{\mathrm{p1}},m)\right)+\overline{p}^{a}\left(\varphi(\overline{\mathrm{p2}},m)\right).(\bar{\gamma}\cdot\overline{w}).\left(\varphi(\overline{\mathrm{p1}},m)\right)$$

Differentiation of 4-vectors living in different dimensions (4, D, $D-4$) works only in the t'Hooft-Veltman scheme

```
FourDivergence[FVD[p, mu], FV[p, nu]]
```

```
FourDivergence: Warning! The input expression also depends on p in dimensions other than 4. The derivative of a
    vector in one dimension (D, 4 or D- 4) w.r.t. the same vector in a different dimension (D, 4 or D- 4) is
    meaningnful only when using the t'Hooft- Veltman scheme. For every other scheme please recheck your
    input expressions and ensure that all matrices, spinors and tensors are purely D- dimensional or
    4- dimensional. You might want to use FCGetDimensions[exp] to find the offending terms. If you explicitly
    intend to use the t'Hooft- Veltman scheme, please activate it via FCSetDiracGammaScheme["BMHV"].
```

$Aborted

```
FCSetDiracGammaScheme["BMHV"];

FourDivergence[FVD[p, mu], FV[p, nu]]
```

$$\bar{g}^{\mathrm{mu\ nu}}$$

4.31 FourLaplacian

FourLaplacian[exp, p, q] is $\frac{\partial}{\partial p_\mu}\frac{\partial}{\partial q_\mu}$ applied to **exp**.

4.31.1 See also

Overview, FourDivergence, ThreeDivergence.

4.31.2 Examples

```
SP[q, q]

FourLaplacian[%, q, q]
```

$$\overline{q}^2$$

$$2D$$

```
SOD[q]^OPEmFAD[q, q - p]

FourLaplacian[%, q, q]
```

$$(\Delta \cdot q)^{\text{OPEmFAD}(q,q-p)}$$

$$0$$

4.32 FreeIndexFreeQ

FreeIndexFreeQ[exp, {head1, head2, ...}] returns **True** if the expression contains uncontracted indices with heads **head1**, **head2**, ... and **False** otherwise.

As always in FeynCalc, Einstein summation convention is implicitly assumed. The function is optimized for large expressions, i.e. it is not so good as a criterion in e.g. **Select**.

4.32.1 See also

Overview, FCRenameDummyIndices, Contract, DummyIndexFreeQ.

4.32.2 Examples

```
FCI[FV[p, \[Mu]] FV[q, \[Nu]]]

FreeIndexFreeQ[%, {LorentzIndex}]
```

$$\overline{p}^{\mu}\overline{q}^{\nu}$$

False

```
FCI[FV[p, \[Mu]] FV[q, \[Mu]]]

FreeIndexFreeQ[%, {LorentzIndex}]
```

$$\overline{p}^{\mu}\overline{q}^{\mu}$$

True

```
FCI[SUNT[a, b]]

FreeIndexFreeQ[%, {SUNIndex}]
```

$$T^{a}.T^{b}$$

False

```
FCI[SUNT[a, a]]

FreeIndexFreeQ[%, {SUNIndex}]
```

$$T^{a}.T^{a}$$

True

4.33 LorentzToCartesian

`LorentzToCartesian[exp]` rewrites Lorentz tensors in form of Cartesian tensors (when possible). Using options one can specify which types of tensors should be converted.

4.33.1 See also

Overview, CartesianToLorentz.

4.33.2 Examples

```
SPD[p, q]

% // LorentzToCartesian
```

$$p \cdot q$$

$$p^0 q^0 - p \cdot q$$

```
LC[\[Mu], \[Nu]][p, q]

% // LorentzToCartesian
```

$$\bar{\epsilon}^{\mu\nu\overline{pq}}$$

$$\bar{g}^{0\mu}\bar{g}^{\$MU(\$20)\nu}\left(-\bar{\epsilon}^{\$MU(\$20)\overline{pq}}\right) - \bar{g}^{\$MU(\$20)\mu}\left(\bar{g}^{0\nu}\left(-\bar{\epsilon}^{\$MU(\$20)\overline{pq}}\right) - \bar{g}^{\$MU(\$21)\nu}\left(q^0\bar{\epsilon}^{\$MU(\$20)\$MU(\$21)\overline{p}} - p^0\bar{\epsilon}^{\$MU(\$20)\$MU(\$21)\overline{q}}\right)\right)$$

```
GAD[\[Mu]]

% // LorentzToCartesian
```

$$\gamma^\mu$$

$$\bar{\gamma}^0\bar{g}^{0\mu} - \gamma^{\$MU(\$22)}g^{\$MU(\$22)\mu}$$

4.34 MomentumCombine

`MomentumCombine[expr]` is the inverse operation to **`MomentumExpand`** and **`ExpandScalarProduct`**. **`MomentumCombine`** combines also **`Pair`**s.

4.34.1 See also

Overview, ExpandScalarProduct, Momentum, MomentumExpand.

4.34.2 Examples

```
Momentum[p] - 2 Momentum[q] // MomentumCombine // StandardForm

(*Momentum[p - 2 q]*)
```

```
FV[p, \[Mu]] + 2 FV[q, \[Mu]]

ex = MomentumCombine[%]
```

$$\overline{p}^{\mu} + 2\overline{q}^{\mu}$$

$$(\overline{p} + 2\overline{q})^{\mu}$$

```
ex // StandardForm

(*Pair[LorentzIndex[\[Mu]], Momentum[p + 2 q]]*)
```

```
ex // ExpandScalarProduct
```

$$\overline{p}^{\mu} + 2\overline{q}^{\mu}$$

```
3 Pair[LorentzIndex[\[Mu]], Momentum[p]] + 2 Pair[LorentzIndex[\[Mu]],
Momentum[q]]

ex = MomentumCombine[%]
```

$$3\overline{p}^{\mu} + 2\overline{q}^{\mu}$$

$$(3\overline{p} + 2\overline{q})^{\mu}$$

```
ex // StandardForm

(*Pair[LorentzIndex[\[Mu]], Momentum[3 p + 2 q]]*)
```

4.35 MomentumExpand

MomentumExpand[expr] expands **Momentum[a+b+ ...]** in **expr** into **Momentum[a] + Momentum[b] + ...**.

4.35.1 See also

Overview, ExpandScalarProduct, MomentumCombine.

4.35.2 Examples

```
MomentumExpand[Momentum[p + q]] // StandardForm

(*Momentum[p] + Momentum[q]*)
```

```
ScalarProduct[p + q, r]
```

$$(\overline{p} + \overline{q}) \cdot \overline{r}$$

```
ScalarProduct[p + q, r] // StandardForm

(*Pair[Momentum[p + q], Momentum[r]]*)
```

```
ex = MomentumExpand[ScalarProduct[p + q, r]]
```

$$(\overline{p} + \overline{q}) \cdot \overline{r}$$

```
ex // StandardForm

(*Pair[Momentum[p] + Momentum[q], Momentum[r]]*)
```

```
ex = MomentumExpand[ScalarProduct[p + q, r - p]]
```

$$(\overline{p} + \overline{q}) \cdot (\overline{r} - \overline{p})$$

```
ex // StandardForm

(*Pair[Momentum[p] + Momentum[q], -Momentum[p] + Momentum[r]]*)
```

4.36 PairContract

PairContract is like **Pair**, but with (local) contraction properties. The function fully supports the BMHV algebra and will expand momenta inside scalar products when it leads to simpler expressions.

PairContract is an auxiliary function used in higher level FeynCalc functions that require fast contractions between multiple expressions, where **Contract** would be too slow.

4.36.1 See also

Overview, Contract, PairContract2, PairContract3.

4.36.2 Examples

```
Pair[LorentzIndex[\[Mu]], Momentum[p]] Pair[LorentzIndex[\[Mu]],
Momentum[q]]

% /. Pair -> PairContract
```

$$\overline{p}^{\mu}\overline{q}^{\mu}$$

$$\overline{p}\cdot\overline{q}$$

```
Pair[LorentzIndex[\[Mu]], Momentum[p]] Pair[LorentzIndex[\[Nu]],
Momentum[q]] Pair[LorentzIndex[\[Mu]], LorentzIndex[\[Nu]]]

% /. Pair -> PairContract
```

$$\overline{p}^{\mu}\overline{q}^{\nu}\bar{g}^{\mu\nu}$$

$$\overline{p}\cdot\overline{q}$$

```
Pair[LorentzIndex[\[Mu]], Momentum[p + q]] Pair[LorentzIndex[\[Mu]],
Momentum[r + s]]

% /. Pair -> PairContract
```

$$(\overline{p}+\overline{q})^{\mu}\,(\overline{r}+\overline{s})^{\mu}$$

$$(\overline{p}+\overline{q})\cdot(\overline{r}+\overline{s})$$

```
FCClearScalarProducts[];
SP[p1, p2] = s2;
SP[p1, p3] = s3;
FCI[SP[p1, p2 + p3]] /. Pair -> PairContract
```

$$\text{s2} + \text{s3}$$

4.37 PairContract2

PairContract2 is like **Pair**, but with local contraction properties. It works best with products of **Pair**s that are expected to evaluate to a product of scalar products.

- Suitable contractions between products of **PairContract2** symbols are evaluated immediately.
- Momenta are never expanded and every **PairContract2** symbol containing **Momentum** in both slots is immediately converted to a **Pair**.
- BMHV algebra is not supported, every tensor must be purely **4** or **D**-dimensional

PairContract2 is an auxiliary function used in higher level FeynCalc functions that require fast contractions between multiple expressions, where **Contract** would be too slow.

4.37.1 See also

Overview, Contract, PairContract, PairContract3.

4.37.2 Examples

```
Pair[LorentzIndex[\[Mu]], Momentum[p]] Pair[LorentzIndex[\[Mu]],
Momentum[q]]

% /. Pair -> PairContract2
```

$$\overline{p}^{\mu}\overline{q}^{\mu}$$

$$\overline{p}\cdot\overline{q}$$

```
Pair[LorentzIndex[\[Mu]], Momentum[p + q]] Pair[LorentzIndex[\[Mu]],
Momentum[r + s]]

% /. Pair -> PairContract2
```

$$(\overline{p}+\overline{q})^{\mu}\,(\overline{r}+\overline{s})^{\mu}$$

$$(\overline{p}+\overline{q})\cdot(\overline{r}+\overline{s})$$

4.38 PairContract3

PairContract3 is like **Pair**, but with local contraction properties among **PairContract3**s. The function fully supports the BMHV algebra and, unlike **PairContract** or **PairContract2** will always expand momenta inside scalar products.

PairContract3 is an auxiliary function used in higher level FeynCalc functions that require fast contractions between multiple expressions, where **Contract** would be too slow.

4.38.1 See also

Overview, Contract, PairContract, PairContract2.

4.38.2 Examples

```
Pair[LorentzIndex[\[Mu]], Momentum[p]] Pair[LorentzIndex[\[Mu]],
Momentum[q]]

% /. Pair -> PairContract3
```

$$\overline{p}^{\mu}\overline{q}^{\mu}$$

$$\overline{p}\cdot\overline{q}$$

```
Pair[LorentzIndex[\[Mu]], Momentum[p]] Pair[LorentzIndex[\[Nu]],
Momentum[q]] Pair[LorentzIndex[\[Mu]], LorentzIndex[\[Nu]]]

% /. Pair -> PairContract3
```

$$\overline{p}^{\mu}\overline{q}^{\nu}\bar{g}^{\mu\nu}$$

$$\overline{p}\cdot\overline{q}$$

```
Pair[LorentzIndex[\[Mu]], Momentum[p + q]] Pair[LorentzIndex[\[Mu]],
Momentum[r + s]]

% /. Pair -> PairContract3
```

$$(\overline{p}+\overline{q})^{\mu}\,(\overline{r}+\overline{s})^{\mu}$$

$$\overline{p}\cdot\overline{r}+\overline{p}\cdot\overline{s}+\overline{q}\cdot\overline{r}+\overline{q}\cdot\overline{s}$$

4.39 Schouten

Schouten[exp] attempts to automatically remove spurious terms in **exp** by applying the Schouten's identity.

Schouten applies the identity for 4-vectors on at most 42 terms in a sum. If it should operate on a larger expression you can give a second argument, e.g. **Schouten[expr, 4711]** which will work on sums with less than 4711 terms.

Schouten is also an option of **Contract** and **DiracTrace**. It may be set to an integer indicating the maximum number of terms onto which the function **Schouten** will be applied.

4.39.1 See also

Overview, Contract, DiracTrace, FCSchoutenBruteForce.

4.39.2 Examples

```
((LC[\[Mu], \[Nu], \[Rho], \[Sigma]] FV[p, \[Tau]] + LC[\[Nu], \[Rho],
\[Sigma], \[Tau]] FV[p, \[Mu]] + LC[\[Rho], \[Sigma], \[Tau], \[Mu]] FV[p,
\[Nu]] +
     LC[\[Sigma], \[Tau], \[Mu], \[Nu]] FV[p, \[Rho]] + LC[\[Tau], \[Mu],
\[Nu], \[Rho]] FV[p, \[Sigma]]))

Schouten[%]
```

$$\overline{p}^{\tau}\bar{\epsilon}^{\mu\nu\rho\sigma} + \overline{p}^{\mu}\bar{\epsilon}^{\nu\rho\sigma\tau} + \overline{p}^{\nu}\bar{\epsilon}^{\rho\sigma\tau\mu} + \overline{p}^{\rho}\bar{\epsilon}^{\sigma\tau\mu\nu} + \overline{p}^{\sigma}\bar{\epsilon}^{\tau\mu\nu\rho}$$

$$0$$

4.40 SetMandelstam

SetMandelstam[s, t, u, p1 , p2 , p3 , p4 , m1 , m2 , m3 , m4] defines the Mandelstam variables $s = (p_1 + p_2)^2$, $t = (p_1 + p_3)^2$, $u = (p_1 + p_4)^2$ and sets the momenta on-shell: $p_1^2 = m_1^2$, $p_2^2 = m_2^2$, $p_3^2 = m_3^2$, $p_4^2 = m_4^2$. Notice that $p_1 + p_2 + p_3 + p_4 = 0$ is assumed.

SetMandelstam[x, {p1, p2, p3, p4, p5}, {m1, m2, m3, m4, m5}] defines $x[i, j] = (p_i + p_j)^2$ and sets the p_i on-shell. The p_i satisfy: $p_1 + p_2 + p_3 + p_4 + p_5 = 0$.

4.40.1 See also

Overview, Mandelstam.

4.40.2 Examples

SetMandelstam assumes all momenta to be ingoing. For scattering processes with $p_1 + p_2 = p_3 + p_4$, the outgoing momenta should be written with a minus sign.

```
FCClearScalarProducts[]

SetMandelstam[s, t, u, p1, p2, -p3, -p4, m1, m2, m3, m4];

SP[p1, p2]

SP[p1, p3]

SP[p1, p4]
```

$$-\frac{\text{m1}^2}{2} - \frac{\text{m2}^2}{2} + \frac{s}{2}$$

$$\frac{\text{m1}^2}{2} + \frac{\text{m3}^2}{2} - \frac{t}{2}$$

$$\frac{\text{m1}^2}{2} + \frac{\text{m4}^2}{2} - \frac{u}{2}$$

SetMandelstam simultaneously sets scalar products in 4 and $D dimensions. This is controlled by the option **Dimension**.

```
SPD[p1, p2]

SPD[p1, p3]
```

$$-\frac{\text{m1}^2}{2} - \frac{\text{m2}^2}{2} + \frac{s}{2}$$

$$\frac{\text{m1}^2}{2} + \frac{\text{m3}^2}{2} - \frac{t}{2}$$

It is also possible to have more than just 4 momenta. For example, for $p1 + p2 = p3 + p4 + p5$ we can obtain **x[i, j]** given by $(p_i + p_j)^2$

```
FCClearScalarProducts[];

SetMandelstam[x, {p1, p2, -p3, -p4, -p5}, {m1, m2, m3, m4, m5}];

SPD[p4, p5]
```

$$\frac{1}{2}x(4,5) - \frac{\text{m4}^2}{2} - \frac{\text{m5}^2}{2}$$

4.41 SetTemporalComponent

SetTemporalComponent[p, val] sets the value of the temporal component of a 4-vector p, **TemporalPair[ExplicitLorentzIndex[0], TemporalMomentum[p]]** to **val**.

4.41.1 See also

Overview, TC, TemporalPair, TemporalMomentum.

4.41.2 Examples

```
FCClearScalarProducts[]

ClearAll[t]

SetTemporalComponent[p, t]

TC[p]
```

$$t$$

```
TC[p + q] // ExpandScalarProduct
```

$$q^0 + t$$

```
SP[p, q] // LorentzToCartesian
```

$$q^0 t - \overline{p} \cdot \overline{q}$$

4.42 TensorFunction

`TensorFunction[t, mu, nu, ...]` transform into **`t[LorentzIndex[mu], LorentzIndex[nu], ...]`**, i.e., it can be used as an unspecified tensorial function **`t`**.

A symmetric tensor can be obtained by **`TensorFunction[{t, "S"}, mu, nu, ...]`**, and an antisymmetric one by **`TensorFunction[{t, "A"}, mu, nu, ...]`**.

4.42.1 See also

Overview, FCSymmetrize, FCAntiSymmetrize.

4.42.2 Examples

```
TensorFunction[t, \[Mu], \[Nu], \[Tau]]
```

$$t(\mu, \nu, \tau)$$

```
TensorFunction[t, \[Mu], \[Nu], \[Tau]] // StandardForm

(*t[LorentzIndex[\[Mu]], LorentzIndex[\[Nu]], LorentzIndex[\[Tau]]]*)
```

```
Contract[FV[p, \[Mu]] TensorFunction[t, \[Mu], \[Nu], \[Tau]]]
```

$$t\left(\overline{p}, \nu, \tau\right)$$

```
Contract[FV[p, \[Mu]] TensorFunction[t, \[Mu], \[Nu], \[Tau]]] //
StandardForm

(*t[Momentum[p], LorentzIndex[\[Nu]], LorentzIndex[\[Tau]]]*)
```

```
TensorFunction[{f, "S"}, \[Alpha], \[Beta]]
```

$$f(\alpha, \beta)$$

```
TensorFunction[{f, "S"}, \[Beta], \[Alpha]] // StandardForm

(*f[LorentzIndex[\[Alpha]], LorentzIndex[\[Beta]]]*)
```

```
Attributes[f]
ClearAttributes[f, Orderless]
```

{Orderless}

4.43 ThreeDivergence

ThreeDivergence[exp, CV[p, i]] calculates the partial derivative of **exp** w.r.t. p^i.

ThreeDivergence[exp, CV[p, i], CV[p, i], ...] gives the multiple derivative.

Owing to the fact that in FeynCalc dummy Cartesian index are always understood to be upper indices, applying **ThreeDivergence** to an expression is equivalent to the action of $\nabla^i = \frac{\partial}{\partial p^i}$.

4.43.1 See also

Overview, FourDivergence.

4.43.2 Examples

```
CSP[p, q]

ThreeDivergence[%, CV[q, i]]
```

$$\overline{p} \cdot \overline{q}$$

$$\overline{p}^i$$

```
CSP[p - k, q]

ThreeDivergence[%, CV[k, i]]
```

$$(\overline{p} - \overline{k}) \cdot \overline{q}$$

$$-\overline{q}^i$$

```
CFAD[{p, m^2}, p - q]

ThreeDivergence[%, CVD[p, i]]
```

$$\frac{1}{(p^2+m^2-i\eta).((p-q)^2-i\eta)}$$

$$\frac{2q^i-2p^i}{(p^2+m^2-i\eta).((p-q)^2-i\eta)^2}-\frac{2p^i}{(p^2+m^2-i\eta)^2.((p-q)^2-i\eta)}$$

Differentiation of 3-vectors living in different dimensions $(3, D-1, D-4)$ works only in the t'Hooft-Veltman scheme

```
ThreeDivergence[CVD[p, i], CV[p, j]]
```

ThreeDivergence: Warning! The input expression also depends on p in dimensions other than 3. The derivative of a vector in one dimension (D- 1, 3 or D- 4) w.r.t. the same vector in a different dimension (D- 1, 3 or D- 4) is meaningful only when using the t'Hooft- Veltman scheme. For every other scheme please recheck your input expressions and ensure that all matrices, spinors and tensors are purely D- 1- dimensional or 3- dimensional. You might want to use FCGetDimensions[exp] to find the offending terms. If you explicitly intend to use the t'Hooft- Veltman scheme, please activate it via FCSetDiracGammaScheme["BMHV"].

$Aborted

```
FCSetDiracGammaScheme["BMHV"];
```

```
ThreeDivergence[CVD[p, i], CV[p, j]]
```

$$\bar{\delta}^{ij}$$

4.44 TrickMandelstam

`TrickMandelstam[expr, {s, t, u, m1^2 + m2^2 + m3^2 + m4^2}]` simplifies all sums in **`expr`** so that one of the Mandelstam variables s, t or u is eliminated by the relation $s+t+u = m_1^2+m_2^2+m_3^2+m_4^2$. The trick is that the resulting sum has the most short number of terms.

4.44.1 See also

Overview, SetMandelstam.

4.44.2 Examples

```
ClearAll[s, t, u]

(s + t - u) (2 SMP["m_W"]^2 - t - u)

TrickMandelstam[%, {s, t, u, 2 SMP["m_W"]^2}] // Factor2
```

$$(s+t-u)\left(2m_W^2-t-u\right)$$

$$-2s\left(u-m_W^2\right)$$

```
M^2 s - s^2 + M^2 t - s t + M^2 u - s u

TrickMandelstam[%, {s, t, u, 2 M^2}]
```

$$M^2s+M^2t+M^2u-s^2-st-su$$

$$2M^2\left(M^2-s\right)$$

4.45 Uncontract

`Uncontract[exp, q1, q2, ...]` uncontracts **`Eps`** and **`DiracGamma`**.

`Uncontract[exp, q1, q2, Pair -> {p}]` uncontracts also $p \cdot q_1$ and $p \cdot q_2$;

The option **`Pair -> All`** uncontracts all momenta except **`OPEDelta`**.

4.45.1 See also

Overview, Contract.

4.45.2 Examples

```
LC[\[Mu], \[Nu]][p, q]
```

```
Uncontract[%, p]
```

$$\bar{\epsilon}^{\mu\nu\overline{pq}}$$

$$\overline{p}^{\$AL(\$19)}\,\bar{\epsilon}^{\mu\nu\ \$AL(\$19)\overline{q}}$$

```
GS[p]
```

```
Uncontract[%, p]
```

$$\bar{\gamma}\cdot\overline{p}$$

$$\bar{\gamma}^{\$AL(\$20)}\overline{p}^{\$AL(\$20)}$$

```
Uncontract[LC[\[Mu], \[Nu]][p, q], p, q]
```

$$\overline{p}^{\$AL(\$22)}\overline{q}^{\$AL(\$21)}\left(-\bar{\epsilon}^{\mu\nu\ \$AL(\$21)\$AL(\$22)}\right)$$

By default scalar products are not uncontracted.

```
Uncontract[SP[p, q], q]
```

$$\overline{p}\cdot\overline{q}$$

Use the option **`Pair->All`** to make the function take care of the scalar products as well

```
Uncontract[SP[p, q], q, Pair -> All]
```

$$\overline{p}^{\$AL(\$23)}\overline{q}^{\$AL(\$23)}$$

```
Uncontract[SP[p, q]^2, q, Pair -> All]
```

$$\overline{p}^{\$AL(\$24)}\overline{p}^{\$AL(\$25)}\overline{q}^{\$AL(\$24)}\overline{q}^{\$AL(\$25)}$$

For Cartesian scalar products you need to use the option **CartesianPair->All**

```
Uncontract[CSP[p, q], q, Pair -> All]
```

$$\overline{p} \cdot \overline{q}$$

```
Uncontract[CSP[p, q], q, CartesianPair -> All]
```

$$\overline{p}^{\$AL(\$26)}\overline{q}^{\$AL(\$26)}$$

4.46 UnDeclareFCTensor

UnDeclareFCTensor[a, b, ...] undeclares **a, b, ...** to be tensor heads, i.e., **DataType[a, b, ..., FCTensor]** is set to **False**.

4.46.1 See also

Overview, DeclareFCTensor, FCTensor.

4.46.2 Examples

```
ClearAll[myTens]

DeclareFCTensor[myTens]

ExpandScalarProduct[myTens[z, Momentum[a + b], Momentum[c + d]]]
```

$$\text{myTens}\left(z,\overline{a},\overline{c}\right)+\text{myTens}\left(z,\overline{a},\overline{d}\right)+\text{myTens}\left(z,\overline{b},\overline{c}\right)+\text{myTens}\left(z,\overline{b},\overline{d}\right)$$

```
UnDeclareFCTensor[myTens]
```

5 Dirac algebra

5.1 Anti5

`Anti5[exp]` anticommutes all γ^5 in exp to the right. **`Anti5[exp, n]`** anticommutes all γ^5 n-times to the right. **`Anti5[exp, -n]`** anticommutes all γ^5 n-times to the left.

5.1.1 See also

Overview, DiracOrder, DiracSimplify, DiracTrick.

5.1.2 Examples

```
GA[5, \[Mu]]

Anti5[%]

Anti5[%, -1]
```

$$\bar{\gamma}^5.\bar{\gamma}^\mu$$

$$-\bar{\gamma}^\mu.\bar{\gamma}^5$$

$$\bar{\gamma}^5.\bar{\gamma}^\mu$$

```
GA[5, \[Alpha], \[Beta], \[Gamma], \[Delta]]

Anti5[%, 2]

Anti5[%%, Infinity]

Anti5[%, -Infinity]
```

$$\bar{\gamma}^5.\bar{\gamma}^\alpha.\bar{\gamma}^\beta.\bar{\gamma}^\gamma.\bar{\gamma}^\delta$$

$$\bar{\gamma}^\alpha.\bar{\gamma}^\beta.\bar{\gamma}^5.\bar{\gamma}^\gamma.\bar{\gamma}^\delta$$

$$\bar{\gamma}^\alpha.\bar{\gamma}^\beta.\bar{\gamma}^\gamma.\bar{\gamma}^\delta.\bar{\gamma}^5$$

$$\bar{\gamma}^5.\bar{\gamma}^\alpha.\bar{\gamma}^\beta.\bar{\gamma}^\gamma.\bar{\gamma}^\delta$$

In the naive γ^5-scheme D-dimensional γ-matrices anticommute with γ^5.

```
GA5 . GAD[\[Mu]]

Anti5[%]
```

$$\bar{\gamma}^5.\gamma^\mu$$

$$-\gamma^\mu.\bar{\gamma}^5$$

`Anti5` also works in the t'Hooft-Veltman-Breitenlohner-Maison scheme

```
FCSetDiracGammaScheme["BMHV"];

Anti5[GA5 . GAD[\[Mu]]]
```

$$2\hat{\gamma}^\mu.\bar{\gamma}^5 - \gamma^\mu.\bar{\gamma}^5$$

```
FCSetDiracGammaScheme["NDR"];
```

5.2 Chisholm

`Chisholm[exp]` substitutes products of three Dirac matrices or slashes by the Chisholm identity.

5.2.1 See also

Overview

5.2.2 Examples

```
GA[\[Mu], \[Nu], \[Rho]]

EpsChisholm[%]
```

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}$$

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}$$

Notice that the output contains dummy indices.

```
GA[\[Alpha], \[Beta], \[Mu], \[Nu]]

Chisholm[%]
```

$$\bar{\gamma}^{\alpha}.\bar{\gamma}^{\beta}.\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}$$

$$i\bar{\gamma}^{\alpha}.\bar{\gamma}^{\text{\$MU(\$22)}}.\bar{\gamma}^{5}\bar{\epsilon}^{\beta\mu\nu\ \text{\$MU(\$22)}} + \bar{\gamma}^{\alpha}.\bar{\gamma}^{\nu}\bar{g}^{\beta\mu} - \bar{\gamma}^{\alpha}.\bar{\gamma}^{\mu}\bar{g}^{\beta\nu} + \bar{\gamma}^{\alpha}.\bar{\gamma}^{\beta}\bar{g}^{\mu\nu}$$

Dummy Lorentz indices may also appear as FCGV.

```
SpinorVBar[p1, m1] . GA[\[Alpha], \[Beta], \[Mu], \[Nu]] . SpinorU[p2, m2]

Chisholm[%]
```

$$\bar{v}(\text{p1},\text{m1}).\bar{\gamma}^{\alpha}.\bar{\gamma}^{\beta}.\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.u(\text{p2},\text{m2})$$

$$i\bar{\epsilon}^{\beta\mu\nu\ \text{\$MU(\$31)}}\left(\varphi(-\overline{\text{p1}},\text{m1})\right).\bar{\gamma}^{\alpha}.\bar{\gamma}^{\text{\$MU(\$31)}}.\bar{\gamma}^{5}.\left(\varphi(\overline{\text{p2}},\text{m2})\right) + \bar{g}^{\beta\mu}\left(\varphi(-\overline{\text{p1}},\text{m1})\right).\bar{\gamma}^{\alpha}.\bar{\gamma}^{\nu}.\left(\varphi(\overline{\text{p2}},\text{m2})\right) - \bar{g}^{\beta\nu}\left(\varphi(-\overline{\text{p1}},\text{m1})\right).\bar{\gamma}^{\alpha}.\bar{\gamma}^{\mu}.\left(\varphi(\overline{\text{p2}},\text{m2})\right) + \bar{g}^{\mu\nu}\left(\varphi(-\overline{\text{p1}},\text{m1})\right).\bar{\gamma}^{\alpha}.\bar{\gamma}^{\beta}.\left(\varphi(\overline{\text{p2}},\text{m2})\right)$$

Chisholm only works with Dirac matrices in 4 dimensions, D-dimensional objects are ignored.

```
Chisholm[GAD[\[Mu], \[Nu], \[Rho]]]
```

$$\gamma^{\mu}.\gamma^{\nu}.\gamma^{\rho}$$

```
Chisholm[GA[\[Alpha], \[Beta], \[Mu]]] . Chisholm[GA[\[Alpha], \[Beta],
\[Mu]]]

DiracSimplify[%]
```

$$\left(i\bar{\gamma}^{\$\mathrm{MU}(\$58)}.\bar{\gamma}^5\bar{\epsilon}^{\alpha\beta\mu\ \$\mathrm{MU}(\$58)} + \bar{\gamma}^{\mu}\bar{g}^{\alpha\beta} - \bar{\gamma}^{\beta}\bar{g}^{\alpha\mu} + \bar{\gamma}^{\alpha}\bar{g}^{\beta\mu}\right).\left(i\bar{\gamma}^{\$\mathrm{MU}(\$67)}.\bar{\gamma}^5\bar{\epsilon}^{\alpha\beta\mu\ \$\mathrm{MU}(\$67)} + \bar{\gamma}^{\mu}\bar{g}^{\alpha\beta} - \bar{\gamma}^{\beta}\bar{g}^{\alpha\mu} + \bar{\gamma}^{\alpha}\bar{g}^{\beta\mu}\right)$$

16

```
Chisholm[GA[\[Alpha], \[Beta], \[Mu], \[Nu]]] . Chisholm[GA[\[Alpha],
\[Beta], \[Mu], \[Nu]]]

DiracSimplify[%]
```

$$\left(i\bar{\gamma}^{\alpha}.\bar{\gamma}^{\$\mathrm{MU}(\$81)}.\bar{\gamma}^5\bar{\epsilon}^{\beta\mu\nu\ \$\mathrm{MU}(\$81)} + \bar{\gamma}^{\alpha}.\bar{\gamma}^{\nu}\bar{g}^{\beta\mu} - \bar{\gamma}^{\alpha}.\bar{\gamma}^{\mu}\bar{g}^{\beta\nu} + \bar{\gamma}^{\alpha}.\bar{\gamma}^{\beta}\bar{g}^{\mu\nu}\right).\left(i\bar{\gamma}^{\alpha}.\bar{\gamma}^{\$\mathrm{MU}(\$90)}.\bar{\gamma}^5\bar{\epsilon}^{\beta\mu\nu\ \$\mathrm{MU}(\$90)} + \bar{\gamma}^{\alpha}.\bar{\gamma}^{\nu}\bar{g}^{\beta\mu} - \bar{\gamma}^{\alpha}.\bar{\gamma}^{\mu}\bar{g}^{\beta\nu} + \bar{\gamma}^{\alpha}.\bar{\gamma}^{\beta}\bar{g}^{\mu\nu}\right)$$

$$-128$$

```
GS[p, q, r]

Chisholm[%]
```

$$(\bar{\gamma}\cdot\overline{p}).(\bar{\gamma}\cdot\overline{q}).(\bar{\gamma}\cdot\bar{r})$$

$$-i\bar{\gamma}^{\$\mathrm{MU}(\$116)}.\bar{\gamma}^5\bar{\epsilon}^{\$\mathrm{MU}(\$116)\overline{pqr}} + (\overline{p}\cdot\overline{q})\,\bar{\gamma}\cdot\bar{r} - (\overline{p}\cdot\bar{r})\,\bar{\gamma}\cdot\overline{q} + \bar{\gamma}\cdot\overline{p}\,(\overline{q}\cdot\bar{r})$$

```
GA[\[Mu], \[Nu], \[Rho], \[Sigma], \[Tau], \[Kappa]]

Chisholm[%]
```

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\sigma}.\bar{\gamma}^{\tau}.\bar{\gamma}^{\kappa}$$

$$\begin{aligned}
&i\bar{g}^{\nu\rho}\bar{\gamma}^{\mu}.\bar{\gamma}^{\$MU(\$125)}.\bar{\gamma}^{5}\bar{\epsilon}^{\kappa\sigma\tau\ \$MU(\$125)} - i\bar{g}^{\kappa\sigma}\bar{\gamma}^{\mu}.\bar{\gamma}^{\$MU(\$127)}.\bar{\gamma}^{5}\bar{\epsilon}^{\nu\rho\tau\ \$MU(\$127)} \\
&+ i\bar{g}^{\kappa\tau}\bar{\gamma}^{\mu}.\bar{\gamma}^{\$MU(\$128)}.\bar{\gamma}^{5}\bar{\epsilon}^{\nu\rho\sigma\ \$MU(\$128)} + i\bar{g}^{\sigma\tau}\bar{\gamma}^{\mu}.\bar{\gamma}^{\$MU(\$129)}.\bar{\gamma}^{5}\bar{\epsilon}^{\kappa\nu\rho\ \$MU(\$129)} - i\bar{\gamma}^{\mu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{5}\bar{\epsilon}^{\kappa\nu\sigma\tau} \\
&+ i\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{5}\bar{\epsilon}^{\kappa\rho\sigma\tau} - \bar{\gamma}^{\mu}.\bar{\gamma}^{\tau}\bar{g}^{\kappa\sigma}\bar{g}^{\nu\rho} + \bar{\gamma}^{\mu}.\bar{\gamma}^{\sigma}\bar{g}^{\kappa\tau}\bar{g}^{\nu\rho} + \bar{\gamma}^{\mu}.\bar{\gamma}^{\tau}\bar{g}^{\kappa\rho}\bar{g}^{\nu\sigma} - \bar{\gamma}^{\mu}.\bar{\gamma}^{\rho}\bar{g}^{\kappa\tau}\bar{g}^{\nu\sigma} \\
&- \bar{\gamma}^{\mu}.\bar{\gamma}^{\sigma}\bar{g}^{\kappa\rho}\bar{g}^{\nu\tau} + \bar{\gamma}^{\mu}.\bar{\gamma}^{\rho}\bar{g}^{\kappa\sigma}\bar{g}^{\nu\tau} - \bar{\gamma}^{\mu}.\bar{\gamma}^{\tau}\bar{g}^{\kappa\nu}\bar{g}^{\rho\sigma} + \bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}\bar{g}^{\kappa\tau}\bar{g}^{\rho\sigma} + \bar{\gamma}^{\mu}.\bar{\gamma}^{\kappa}\bar{g}^{\nu\tau}\bar{g}^{\rho\sigma} + \bar{\gamma}^{\mu}.\bar{\gamma}^{\sigma}\bar{g}^{\kappa\nu}\bar{g}^{\rho\tau} \\
&- \bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}\bar{g}^{\kappa\sigma}\bar{g}^{\rho\tau} - \bar{\gamma}^{\mu}.\bar{\gamma}^{\kappa}\bar{g}^{\nu\sigma}\bar{g}^{\rho\tau} - \bar{\gamma}^{\mu}.\bar{\gamma}^{\rho}\bar{g}^{\kappa\nu}\bar{g}^{\sigma\tau} + \bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}\bar{g}^{\kappa\rho}\bar{g}^{\sigma\tau} + \bar{\gamma}^{\mu}.\bar{\gamma}^{\kappa}\bar{g}^{\nu\rho}\bar{g}^{\sigma\tau}
\end{aligned}$$

Check the equality of the expressions before and after applying **Chisholm**.

```
DiracSimplify[GA[\[Mu], \[Nu], \[Rho], \[Sigma], \[Tau], \[Kappa]] .
GA[\[Mu], \[Nu], \[Rho], \[Sigma], \[Tau], \[Kappa]]]
```

$$-2048$$

```
DiracSimplify[Chisholm[GA[\[Mu], \[Nu], \[Rho], \[Sigma], \[Tau],
\[Kappa]]] . Chisholm[GA[\[Mu], \[Nu], \[Rho], \[Sigma], \[Tau],
\[Kappa]]]]
```

$$-2048$$

```
DiracReduce[GA[\[Mu], \[Nu], \[Rho], \[Sigma], \[Tau], \[Kappa]] .
Chisholm[GA[\[Mu], \[Nu], \[Rho], \[Sigma], \[Tau], \[Kappa]]]]
```

$$-2048$$

Older FeynCalc versions had a function called **Chisholm2** that acted on expressions like $\gamma^{\mu}\gamma^{\nu}\gamma^{5}$. This functionality is now part of **Chisholm** and can be activated by setting the option **Mode** to **2**.

```
GA[\[Mu], \[Nu], 5]

Chisholm[%, Mode -> 2]
```

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{5}$$

$$\frac{1}{2}\sigma^{\$MU(\$1022)\$MU(\$1023)}\bar{\epsilon}^{\mu\nu\ \$MU(\$1022)\$MU(\$1023)} + \bar{\gamma}^{5}\bar{g}^{\mu\nu}$$

5.3 DiracChainJoin

DiracChainJoin[exp] joins chains of Dirac matrices with explicit Dirac indices wrapped with a head **DiracChain**. Notice that **DiracChainJoin** is not suitable for creating closed Dirac chains out of the FeynArts output with explicit Dirac indices, e.g. when the model contains 4-fermion operators. Use **FCFADiracChainJoin** for that.

5.3.1 See also

Overview, DiracChain, DCHN, DiracIndex, DiracIndexDelta, DIDelta, DiracChainCombine, DiracChainExpand, DiracChainFactor, FCFADiracChainJoin.

5.3.2 Examples

```
DCHN[SpinorUBar[p1, m1], i] DCHN[GAD[\[Mu]] . GAD[\[Nu]], i, j] DCHN[j,
SpinorV[p2, m2]]

DiracChainJoin[%]
```

$$(v(\mathrm{p2},\mathrm{m2}))_j\,(\gamma^\mu.\gamma^\nu)_{ij}\,(\bar{u}(\mathrm{p1},\mathrm{m1}))_i$$

$$\left(\varphi(\overline{\mathrm{p1}},\mathrm{m1})\right).\gamma^\mu.\gamma^\nu.\left(\varphi(-\overline{\mathrm{p2}},\mathrm{m2})\right)$$

5.4 FCFADiracChainJoin

FCFADiracChainJoin[exp] processes the output of FeynArts (after **FCFAConvert**) with explicit Dirac indices and joins matrices and spinors into closed chains. This is necessary e. g. for models with 4-fermion operators, where FeynArts cannot determine the correct relative signs. When two matrices have a common index but the positions do not match, as in $A_{ij}B_{ik}$, it is assumed that we can take the charge conjugate transposed of either matrix to obtain, e.g. $\left(CA^TC^{-1}\right)_{ji}B_{ik}$ or $\left(CB^TC^{-1}\right)_{ki}A_{ij}$.

5.4.1 See also

Overview, DiracChain, DCHN, DiracIndex, DiracIndexDelta, DIDelta, DiracChainCombine, DiracChainExpand, DiracChainFactor, DiracChainJoin, FCCCT.

5.4.2 Examples

Create a closed chain for the 1-loop electron self-energy

```
-(1/(16 \[Pi]^4)) I el^2 DCHN[Spinor[-Momentum[p, D], me, 1], Dir1]*
   DCHN[Spinor[Momentum[q, D], me, 1], Dir2] DCHN[GAD[Lor1], Dir1, Dir3]*
   DCHN[GAD[Lor2], Dir2, Dir4] DCHN[me - GSD[k], Dir3, Dir4] FAD[{k, me}, k
 - q] MTD[Lor1, Lor2]

res = FCFADiracChainJoin[%]
```

$$-\frac{i\ \mathrm{el}^2 g^{\mathrm{Lor1}\ \mathrm{Lor2}} \left(\gamma^{\mathrm{Lor1}}\right)_{\mathrm{Dir1}\ \mathrm{Dir3}} \left(\gamma^{\mathrm{Lor2}}\right)_{\mathrm{Dir2}\ \mathrm{Dir4}} (\mathrm{me} - \gamma \cdot k)_{\mathrm{Dir3}\ \mathrm{Dir4}} (\varphi(-p, \mathrm{me}))_{\mathrm{Dir1}} (\varphi(q, \mathrm{me}))_{\mathrm{Dir2}}}{16\pi^4 \left(k^2 - \mathrm{me}^2\right).(k-q)^2}$$

$$-\frac{i\ \mathrm{el}^2 g^{\mathrm{Lor1}\ \mathrm{Lor2}} (\varphi(q, \mathrm{me})).\gamma^{\mathrm{Lor2}}.\left(-(\gamma \cdot k + \mathrm{me}).\gamma^{\mathrm{Lor1}}\right).(\varphi(p, \mathrm{me}))}{16\pi^4 \left(k^2 - \mathrm{me}^2\right).(k-q)^2}$$

Sometimes the ordering of the spinors is not the one wants to have. However, we can always transpose the chains to reorder the spinors as we like, which doesn't change the final result

```
SpinorChainTranspose[res, Select -> {{Spinor[__], Spinor[__]}}]
```

$$-\frac{i\ \mathrm{el}^2 g^{\mathrm{Lor1}\ \mathrm{Lor2}} (\varphi(-p, \mathrm{me})).\gamma^{\mathrm{Lor1}}.(\mathrm{me} - \gamma \cdot k).\gamma^{\mathrm{Lor2}}.(\varphi(-q, \mathrm{me}))}{16\pi^4 \left(k^2 - \mathrm{me}^2\right).(k-q)^2}$$

Using patterns in the **`Select`** option one can create very fine-grained criteria for transposing the chains.

5.5 DiracChainCombine

`DiracChainCombine[exp]` is (nearly) the inverse operation to **`DiracChainExpand`**.

5.5.1 See also

Overview, DiracChain, DCHN, DiracIndex, DiracIndexDelta, DIDelta, DiracChainJoin, DiracChainExpand, DiracChainFactor.

5.5.2 Examples

```
(DCHN[GSD[q], Dir3, Dir4] FAD[{k, me}])/(2 SPD[q, q]) + 1/(2 SPD[q, q])
FAD[k,
     {k - q, me}] (-2 DCHN[GSD[q], Dir3, Dir4] SPD[q, q] + 2 DCHN[1, Dir3,
     Dir4] me SPD[q, q] +
      DCHN[GSD[q], Dir3, Dir4] (-me^2 + SPD[q, q]))

DiracChainCombine[%]
```

$$\frac{\left(q^2-\text{me}^2\right)(\gamma\cdot q)_{\text{Dir3 Dir4}}+2\,\text{me}q^2(1)_{\text{Dir3 Dir4}}-2q^2(\gamma\cdot q)_{\text{Dir3 Dir4}}}{2q^2k^2.\left((k-q)^2-\text{me}^2\right)}+\frac{(\gamma\cdot q)_{\text{Dir3 Dir4}}}{2q^2\left(k^2-\text{me}^2\right)}$$

$$\frac{\left(\left(q^2-\text{me}^2\right)\gamma\cdot q+2\,\text{me}q^2-2q^2\gamma\cdot q\right)_{\text{Dir3 Dir4}}}{2q^2k^2.\left((k-q)^2-\text{me}^2\right)}+\frac{(\gamma\cdot q)_{\text{Dir3 Dir4}}}{2q^2\left(k^2-\text{me}^2\right)}$$

5.6 DiracChainExpand

`DiracChainExpand[exp]` expands all Dirac chains with explicit indices using linearity, e.g. **`DCHN[GA[p1]+GA[p2]+m, i, j]`** becomes **`DCHN[GA[p1], i, j]+DCHN[GA[p2], i, j]+m*DCHN[1, i, j]`**.

5.6.1 See also

Overview, DiracChain, DCHN, DiracIndex, DiracIndexDelta, DIDelta, DiracChainJoin, DiracChainCombine, DiracChainFactor.

5.6.2 Examples

```
DCHN[(GS[p] + m) . GA[mu], i, j]

DiracChainExpand[%]
```

$$\left((\bar{\gamma}\cdot\overline{p}+m).\bar{\gamma}^{\text{mu}}\right)_{ij}$$

$$m\left(\bar{\gamma}^{\text{mu}}\right)_{ij}+\left((\bar{\gamma}\cdot\overline{p}).\bar{\gamma}^{\text{mu}}\right)_{ij}$$

5.7 DiracChainFactor

DiracChainFactor[exp] factors out all expressions inside a **DiracChain** to which the chain doesn't apply. For example, all objects that are not Dirac matrices can be safely factrored out from every Dirac chain.

5.7.1 See also

Overview, DiracChain, DCHN, DiracIndex, DiracIndexDelta, DIDelta, DiracChainJoin, DiracChainCombine, DiracChainExpand.

5.7.2 Examples

```
DCHN[FV[p, \[Nu]] GA[\[Mu]] . GA[\[Nu]] . GA[\[Mu]], i, j]

DiracChainFactor[%]
```

$$\left(\overline{p}^{\nu}\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\mu}\right)_{ij}$$

$$\overline{p}^{\nu}\left(\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\mu}\right)_{ij}$$

5.8 DiracEquation

DiracEquation[exp] applies the Dirac equation without expanding exp. If expansions are necessary, use **DiracSimplify**.

5.8.1 See also

Overview

5.8.2 Examples

```
GS[p] . SpinorU[p, m]

DiracSimplify[%]
```

$$\left(\bar{\gamma}\cdot\overline{p}\right).u(p,m)$$

$$m\left(\varphi(\overline{p},m)\right)$$

```
GS[p] . SpinorU[p, m]
DiracEquation[%]
```

$$(\bar{\gamma} \cdot \overline{p}) .u(p, m)$$

$$m \left(\varphi(\overline{p}, m)\right)$$

```
GS[p] . SpinorV[p, m]
DiracEquation[%]
```

$$(\bar{\gamma} \cdot \overline{p}) .v(p, m)$$

$$-m \left(\varphi(-\overline{p}, m)\right)$$

```
SpinorUBar[p, 0] . GS[p]
DiracEquation[%]
```

$$\bar{u}(p). (\bar{\gamma} \cdot \overline{p})$$

$$0$$

`DiracEquation` also works in D-dimensions

```
SpinorVBarD[p, m] . GSD[p]
DiracEquation[%]
```

$$\bar{v}(p, m).(\gamma \cdot p)$$

$$-m(\varphi(-p, m))$$

5.9 DiracGammaCombine

DiracGammaCombine[exp] is (nearly) the inverse operation to **DiracGammaExpand**.

5.9.1 See also

Overview, DiracGamma, DiracGammaExpand, DiracSimplify, DiracTrick.

5.9.2 Examples

```
GS[p] + GS[q]

ex = DiracGammaCombine[%]
```

$$\bar{\gamma} \cdot \overline{p} + \bar{\gamma} \cdot \overline{q}$$

$$\bar{\gamma} \cdot (\overline{p} + \overline{q})$$

```
ex // StandardForm

(*DiracGamma[Momentum[p + q]]*)
```

```
2 GSD[p] - 3 GSD[q]

ex = DiracGammaCombine[%]
```

$$2\gamma \cdot p - 3\gamma \cdot q$$

$$\gamma \cdot (2p - 3q)$$

```
ex // StandardForm

(*DiracGamma[Momentum[2 p - 3 q, D], D]*)
```

```
DiracGammaCombine[2 GSD[p] - 3 GSD[q]]

DiracGammaExpand[%]
```

$$\gamma \cdot (2p - 3q)$$

$$2\gamma \cdot p - 3\gamma \cdot q$$

5.10 DiracGammaExpand

DiracGammaExpand[exp] expands Dirac matrices contracted to linear combinations of 4-vectors. All **DiracGamma[Momentum[a+b+ ...]]** will be expanded to **DiracGamma[Momentum[a]] + DiracGamma[Momentum[b]] + DiracGamma[Momentum[...]]** .

5.10.1 See also

Overview, DiracGamma, DiracGammaCombine, DiracSimplify, DiracTrick.

5.10.2 Examples

```
GS[q] . GS[p - q]

ex = DiracGammaExpand[%]
```

$$(\bar{\gamma}\cdot\bar{q}).(\bar{\gamma}\cdot(\bar{p}-\bar{q}))$$

$$(\bar{\gamma}\cdot\bar{q}).(\bar{\gamma}\cdot\bar{p}-\bar{\gamma}\cdot\bar{q})$$

```
ex // StandardForm

(*DiracGamma[Momentum[q]] . (DiracGamma[Momentum[p]] -
DiracGamma[Momentum[q]])*)
```

DiracGammaExpand rewrites $\gamma^{\mu}(p-q)_{\mu}$ as $\gamma^{mu}p_{mu}-\gamma^{\mu}q_{\mu}$.

The inverse operation is **DiracGammaCombine**.

```
GS[q] . (GS[p] - GS[q])

ex = DiracGammaCombine[%]
```

$$(\bar{\gamma}\cdot\bar{q}).(\bar{\gamma}\cdot\bar{p}-\bar{\gamma}\cdot\bar{q})$$

$$(\bar{\gamma}\cdot\bar{q}).(\bar{\gamma}\cdot(\bar{p}-\bar{q}))$$

```
ex // StandardForm

(*DiracGamma[Momentum[q]] . DiracGamma[Momentum[p - q]]*)
```

It is possible to perform the expansions only on Dirac matrices contracted with particular momenta.

```
c1 GAD[\[Mu]] . (GSD[p1 + p2] + m) . GAD[\[Nu]] + c2 GAD[\[Mu]] . (GSD[q1 +
q2] + m) . GAD[\[Nu]]

DiracGammaExpand[%, Momentum -> {q1}]
```

$$\text{c1}\gamma^{\mu}.(m+\gamma\cdot(\text{p1}+\text{p2})).\gamma^{\nu}+\text{c2}\gamma^{\mu}.(m+\gamma\cdot(\text{q1}+\text{q2})).\gamma^{\nu}$$

$$\text{c1}\gamma^{\mu}.(m+\gamma\cdot(\text{p1}+\text{p2})).\gamma^{\nu}+\text{c2}\gamma^{\mu}.(m+\gamma\cdot\ \text{q1}+\gamma\cdot\ \text{q2}).\gamma^{\nu}$$

If the input expression contains **DiracSigma**, **DiracGammaExpand** will expand Feynman slashes inside **DiracSigma** and call **DiracSigmaExpand**.

```
DiracSigma[GSD[p + q], GSD[r]]

DiracGammaExpand[%]
```

$$\sigma^{p+qr}$$

$$\sigma^{pr}+\sigma^{qr}$$

The call to **DiracSigmaExpand** can be inhibited by disabling the corresponding option.

```
DiracGammaExpand[DiracSigma[GSD[p + q], GSD[r]], DiracSigmaExpand -> False]
```

$$\text{DiracSigma}(\gamma\cdot p+\gamma\cdot q,\gamma\cdot r)$$

Use **DiracSimplify** for noncommutative expansions with the corresponding simplifications.

```
DiracSimplify[GS[q] . (GS[p - q])]
```

$$(\bar{\gamma}\cdot\bar{q}).(\bar{\gamma}\cdot\bar{p})-\bar{q}^2$$

If simplifications are not required, you may also combine **DiracGammaExpand** with **DotSimplify**.

```
DotSimplify[DiracGammaExpand[GS[q] . (GS[p - q])]]
```

$$(\bar{\gamma}\cdot\bar{q}).(\bar{\gamma}\cdot\bar{p})-(\bar{\gamma}\cdot\bar{q}).(\bar{\gamma}\cdot\bar{q})$$

5.11 DiracOrder

DiracOrder[exp] orders the Dirac matrices in **exp** lexicographically. **DiracOrder[exp, orderlist]** orders the Dirac matrices in **exp** according to **orderlist**. **DiracOrder** is also an option of **DiracSimplify** and some other functions dealing with Dirac algebra. If set to **True**, the function **DiracOrder** will be applied to the intermediate result to reorder the Dirac matrices lexicographically.

5.11.1 See also

Overview, DiracSimplify, DiracTrick.

5.11.2 Examples

```
GA[\[Beta], \[Alpha]]

DiracOrder[%]
```

$$\bar{\gamma}^{\beta}.\bar{\gamma}^{\alpha}$$

$$2\bar{g}^{\alpha\beta} - \bar{\gamma}^{\alpha}.\bar{\gamma}^{\beta}$$

DiracOrder also works with Dirac matrices in D-dimensions.

```
GAD[\[Rho], \[Nu], \[Mu], \[Nu]]

DiracOrder[%]
```

$$\gamma^{\rho}.\gamma^{\nu}.\gamma^{\mu}.\gamma^{\nu}$$

$$(D-2)\gamma^{\mu}.\gamma^{\rho} + 2(2-D)g^{\mu\rho}$$

By default γ^5 is moved to the right.

```
GA[5, \[Mu], \[Nu]]

DiracOrder[%]
```

$$\bar{\gamma}^{5}.\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}$$

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{5}$$

```
GA[6, \[Mu], 7]

DiracOrder[%]
```

$$\bar{\gamma}^6.\bar{\gamma}^\mu.\bar{\gamma}^7$$

$$\bar{\gamma}^\mu.\bar{\gamma}^7$$

orderlist comes into play when we need an ordering that is not lexicographic

```
GA[\[Alpha], \[Beta], \[Delta]]

DiracOrder[%]
```

$$\bar{\gamma}^\alpha.\bar{\gamma}^\beta.\bar{\gamma}^\delta$$

$$\bar{\gamma}^\alpha.\bar{\gamma}^\beta.\bar{\gamma}^\delta$$

```
DiracOrder[GA[\[Alpha], \[Beta], \[Delta]], {\[Delta], \[Beta], \[Alpha]}]
```

$$-\bar{\gamma}^\delta.\bar{\gamma}^\beta.\bar{\gamma}^\alpha + 2\bar{\gamma}^\delta \bar{g}^{\alpha\beta} - 2\bar{\gamma}^\beta \bar{g}^{\alpha\delta} + 2\bar{\gamma}^\alpha \bar{g}^{\beta\delta}$$

Reordering of Dirac matrices in long chains is expensive, so that **DiracSimplify** does not do it by default.

```
DiracSimplify[GAD[\[Mu], \[Nu]] + GAD[\[Nu], \[Mu]]]
```

$$\gamma^\mu.\gamma^\nu + \gamma^\nu.\gamma^\mu$$

However, if you know that it can lead to simpler expressions, you can activate the reordering via the option **DiracOrder**.

```
DiracSimplify[GAD[\[Mu], \[Nu]] + GAD[\[Nu], \[Mu]], DiracOrder -> True]
```

$$2g^{\mu\nu}$$

Reproduce Eq. 18.128 from An Introduction to Quantum Field Theory by M. Peskin and D. Schroeder.

```
DiracSimplify[1/2 (GAD[\[Mu], \[Alpha], \[Nu]] + GAD[\[Nu], \[Alpha],
\[Mu]]), DiracOrder -> True]
```

$$\gamma^{\nu} g^{\alpha\mu} + \gamma^{\mu} g^{\alpha\nu} - \gamma^{\alpha} g^{\mu\nu}$$

5.12 DiracReduce

DiracReduce[exp] reduces all 4-dimensional Dirac matrices in exp to the standard basis (S, P, V, A, T) using the Chisholm identity.

In the result the basic Dirac structures can be wrapped with a head **DiracBasis**, that is

- S: **DiracBasis[1]**
- P: **DiracBasis[GA[5]]**
- V: **DiracBasis[GA[mu]]**
- A: **DiracBasis[GA[mu, 5]]**
- T: **DiracBasis[DiracSigma[GA[mu, nu]]]**

By default **DiracBasis** is substituted to **Identity**.

5.12.1 See also

Overview, Chisholm, DiracSimplify, EpsChisholm.

5.12.2 Examples

```
GA[\[Mu], \[Nu]]

DiracReduce[%]
```

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}$$

$$\bar{g}^{\mu\nu} - i\sigma^{\mu\nu}$$

DiracReduce only works with Dirac matrices in 4 dimensions, D-dimensional matrices are ignored.

```
GAD[\[Mu], \[Nu]]

DiracReduce[%]
```

$$\gamma^{\mu}.\gamma^{\nu}$$

$$\gamma^{\mu}.\gamma^{\nu}$$

```
SpinorUBar[Subscript[p, 1], Subscript[m, 1]] . GA[\[Mu], \[Nu], \[Rho]] .
SpinorV[Subscript[p, 2], Subscript[m, 2]]

DiracReduce[%]
```

$$\bar{u}\left(p_1, m_1\right).\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.v\left(p_2, m_2\right)$$

$$i\bar{\epsilon}^{\mu\nu\rho\,\$\mathrm{MU}(\$31)}\left(\varphi(\overline{p}_1, m_1)\right).\bar{\gamma}^{\$\mathrm{MU}(\$31)}.\bar{\gamma}^{5}.\left(\varphi(-\overline{p}_2, m_2)\right) + \bar{g}^{\mu\nu}\left(\varphi(\overline{p}_1, m_1)\right).\bar{\gamma}^{\rho}.\left(\varphi(-\overline{p}_2, m_2)\right) - \bar{g}^{\mu\rho}\left(\varphi(\overline{p}_1, m_1)\right).\bar{\gamma}^{\nu}.\left(\varphi(-\overline{p}_2, m_2)\right) + \bar{g}^{\nu\rho}\left(\varphi(\overline{p}_1, m_1)\right).\bar{\gamma}^{\mu}.\left(\varphi(-\overline{p}_2, m_2)\right)$$

```
GA[\[Mu], \[Nu], \[Rho], \[Sigma]]

DiracReduce[%]
```

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\sigma}$$

$$-i\bar{\gamma}^{5}\bar{\epsilon}^{\mu\nu\rho\sigma} - i\sigma^{\rho\sigma}\bar{g}^{\mu\nu} + i\sigma^{\nu\sigma}\bar{g}^{\mu\rho} - i\sigma^{\nu\rho}\bar{g}^{\mu\sigma} - i\sigma^{\mu\sigma}\bar{g}^{\nu\rho} + i\sigma^{\mu\rho}\bar{g}^{\nu\sigma} - i\sigma^{\mu\nu}\bar{g}^{\rho\sigma} + \bar{g}^{\mu\sigma}\bar{g}^{\nu\rho} - \bar{g}^{\mu\rho}\bar{g}^{\nu\sigma} + \bar{g}^{\mu\nu}\bar{g}^{\rho\sigma}$$

Do some checks of the results

```
DiracSimplify[GA[\[Mu], \[Nu], \[Rho], \[Sigma]] . GA[\[Mu], \[Nu], \[Rho],
\[Sigma]]]
```

$$-128$$

```
DiracSimplify[DiracReduce[GA[\[Mu], \[Nu], \[Rho], \[Sigma]]] .
DiracReduce[GA[\[Mu], \[Nu], \[Rho], \[Sigma]]]]
```

$$-128$$

We may also keep the head **DiracBasis** in the final result

```
DiracReduce[GA[\[Mu], \[Nu], \[Rho], \[Sigma]], FinalSubstitutions -> {}]
```

$$
\begin{aligned}
&-i\bar{g}^{\mu\nu}\ \text{DiracBasis}\left(\text{DiracSigma}\left(\text{DiracBasis}\left(\bar{\gamma}^{\rho}\right),\right.\right.\\
&\left.\left.\text{DiracBasis}\left(\bar{\gamma}^{\sigma}\right)\right)\right)+i\bar{g}^{\mu\rho}\ \text{DiracBasis}\left(\text{DiracSigma}\left(\text{DiracBasis}\left(\bar{\gamma}^{\nu}\right),\right.\right.\\
&\left.\left.\text{DiracBasis}\left(\bar{\gamma}^{\sigma}\right)\right)\right)-i\bar{g}^{\mu\sigma}\ \text{DiracBasis}\left(\text{DiracSigma}\left(\text{DiracBasis}\left(\bar{\gamma}^{\nu}\right),\right.\right.\\
&\left.\left.\text{DiracBasis}\left(\bar{\gamma}^{\rho}\right)\right)\right)-i\bar{g}^{\nu\rho}\ \text{DiracBasis}\left(\text{DiracSigma}\left(\text{DiracBasis}\left(\bar{\gamma}^{\mu}\right),\right.\right.\\
&\left.\left.\text{DiracBasis}\left(\bar{\gamma}^{\sigma}\right)\right)\right)+i\bar{g}^{\nu\sigma}\ \text{DiracBasis}\left(\text{DiracSigma}\left(\text{DiracBasis}\left(\bar{\gamma}^{\mu}\right),\right.\right.\\
&\left.\left.\text{DiracBasis}\left(\bar{\gamma}^{\rho}\right)\right)\right)-i\bar{g}^{\rho\sigma}\ \text{DiracBasis}\left(\text{DiracSigma}\left(\text{DiracBasis}\left(\bar{\gamma}^{\mu}\right),\text{DiracBasis}\left(\bar{\gamma}^{\nu}\right)\right)\right)\\
&-i\ \text{DiracBasis}\left(\bar{\gamma}^{5}\right)\bar{\epsilon}^{\mu\nu\rho\sigma}+\text{DiracBasis}(1)\bar{g}^{\mu\sigma}\bar{g}^{\nu\rho}-\text{DiracBasis}(1)\bar{g}^{\mu\rho}\bar{g}^{\nu\sigma}+\text{DiracBasis}(1)\bar{g}^{\mu\nu}\bar{g}^{\rho\sigma}
\end{aligned}
$$

5.13 DiracSigmaExpand

`DiracSigmaExpand[exp]` applies linearity to the arguments of **`DiracSigma`**.

5.13.1 See also

Overview, DiracGamma, DiracSigma.

5.13.2 Examples

```
DiracSigma[GSD[p] + GSD[q], GSD[r]]

ex = % // DiracSigmaExpand
```

$$\text{DiracSigma}(\gamma\cdot p+\gamma\cdot q,\gamma\cdot r)$$

$$\sigma^{pr}+\sigma^{qr}$$

```
ex // FCE // StandardForm

(*DiracSigma[GSD[p], GSD[r]] + DiracSigma[GSD[q], GSD[r]]*)
```

Notice that DiracSigmaExpand does not expand Dirac matrices contracted to linear combinations of 4-vectors by default.

```
DiracSigma[GSD[p + q], GSD[r]]

DiracSigmaExpand[%]
```

$$\sigma^{p+qr}$$

$$\sigma^{p+qr}$$

If such expansions are required, use the option **DiracGammaExpand**.

```
DiracSigmaExpand[DiracSigma[GSD[p + q], GSD[r]], DiracGammaExpand -> True]
```

$$\sigma^{pr} + \sigma^{qr}$$

The option Momentum allows us to perform more fine-grained expansions of **DiracSigma**.

```
DiracSigma[GSD[p], GSD[r] + GSD[t]] + DiracSigma[GSD[l] + GSD[n], GSD[p]]

DiracSigmaExpand[%, Momentum -> {r}]
```

$$\text{DiracSigma}(\gamma \cdot l + \gamma \cdot n, \gamma \cdot p) + \text{DiracSigma}(\gamma \cdot p, \gamma \cdot r + \gamma \cdot t)$$

$$\text{DiracSigma}(\gamma \cdot l + \gamma \cdot n, \gamma \cdot p) + \sigma^{pr} + \sigma^{pt}$$

5.14 DiracSigmaExplicit

DiracSigmaExplicit[exp] inserts in exp for all **DiracSigma** its definition. **DiracSigmaExplict** is also an option of **DiracSimplify**. **DiracSigmaExplict** is also an option of various FeynCalc functions that handle the Dirac algebra.

5.14.1 See also

Overview, DiracGamma, DiracSigma.

5.14.2 Examples

```
DiracSigma[GA[\[Alpha]], GA[\[Beta]]]

DiracSigmaExplicit[%]
```

$$\sigma^{\alpha\beta}$$

$$\frac{1}{2} i \left(\bar{\gamma}^{\alpha}.\bar{\gamma}^{\beta} - \bar{\gamma}^{\beta}.\bar{\gamma}^{\alpha}\right)$$

```
GSD[p] . DiracSigma[GAD[\[Mu]], GAD[\[Nu]]] . GSD[q]

DiracSigmaExplicit[%]
```

$$(\gamma \cdot p).\sigma^{\mu\nu}.(\gamma \cdot q)$$

$$\frac{1}{2} i \left((\gamma \cdot p).\gamma^{\mu}.\gamma^{\nu}.(\gamma \cdot q) - (\gamma \cdot p).\gamma^{\nu}.\gamma^{\mu}.(\gamma \cdot q)\right)$$

5.15 DiracSimplify

`DiracSimplify[exp]` simplifies products of Dirac matrices in **`exp`** and expands noncommutative products. The simplifications are done by applying **`Contract`**, **`DiracEquation`**, **`DiracTrick`**, **`SpinorChainTrick`** and **`SirlinSimplify`**. All γ^5, γ^6 and γ^7 are moved to the right. The order of the other Dirac matrices is not changed, unless the option DiracOrder is set to **`True`**.

5.15.1 See also

Overview, Contract, DiracEquation, DiracSigmaExplicit, DiracSubstitute5, DiracSubstitute67, DiracGamma, DiracGammaExpand, DiracOrder, DiracTrace, DiracTraceEvaluate, DiracTrick, FCDiracIsolate, SirlinSimplify, SpinorChainTrick, SpinorChainEvaluate, ToDiracGamma67.

5.15.2 Examples

Simplify a 4-dimensional Dirac matrix chain with a dummy Lorentz index

```
GA[\[Mu], \[Nu], \[Mu]]

DiracSimplify[%]
```

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\mu}$$

$$-2\bar{\gamma}^{\nu}$$

Another common simplification concerns Dirac matrices contracted to the same 4-vector

```
GS[p] . GS[p]

DiracSimplify[%]
```

$$(\bar{\gamma}\cdot\overline{p})\,.\,(\bar{\gamma}\cdot\overline{p})$$

$$\overline{p}^2$$

Unlike **DiracTrick**, **DiracSimplify** also carries out noncommutative expansions

```
GS[a + b] . GS[p] . GS[c + d] . GS[p]

DiracSimplify[%]
```

$$\left(\bar{\gamma}\cdot\left(\overline{a}+\overline{b}\right)\right).(\bar{\gamma}\cdot\overline{p}).\left(\bar{\gamma}\cdot\left(\overline{c}+\overline{d}\right)\right).(\bar{\gamma}\cdot\overline{p})$$

$$\begin{aligned}&2\,(\overline{c}\cdot\overline{p})\,(\bar{\gamma}\cdot\overline{a})\,.\,(\bar{\gamma}\cdot\overline{p})-\overline{p}^2\,(\bar{\gamma}\cdot\overline{a})\,.\,(\bar{\gamma}\cdot\overline{c})+2\,(\overline{d}\cdot\overline{p})\,(\bar{\gamma}\cdot\overline{a})\,.\,(\bar{\gamma}\cdot\overline{p})-\overline{p}^2\,(\bar{\gamma}\cdot\overline{a})\,.\,(\bar{\gamma}\cdot\overline{d})\\&+2\,(\overline{c}\cdot\overline{p})\,(\bar{\gamma}\cdot\overline{b})\,.\,(\bar{\gamma}\cdot\overline{p})-\overline{p}^2\,(\bar{\gamma}\cdot\overline{b})\,.\,(\bar{\gamma}\cdot\overline{c})+2\,(\overline{d}\cdot\overline{p})\,(\bar{\gamma}\cdot\overline{b})\,.\,(\bar{\gamma}\cdot\overline{p})-\overline{p}^2\,(\bar{\gamma}\cdot\overline{b})\,.\,(\bar{\gamma}\cdot\overline{d})\end{aligned}$$

```
DiracTrick[GS[a + b] . GS[p] . GS[c + d] . GS[p]]
```

$$2\left(\bar{\gamma}\cdot\left(\overline{a}+\overline{b}\right)\right).(\bar{\gamma}\cdot\overline{p})\left((\overline{c}+\overline{d})\cdot\overline{p}\right)-\overline{p}^2\left(\bar{\gamma}\cdot\left(\overline{a}+\overline{b}\right)\right).\left(\bar{\gamma}\cdot\left(\overline{c}+\overline{d}\right)\right)$$

Some of those expansions can be inhibited via the option **Expanding**.

```
DiracSimplify[GS[a + b] . GS[p] . GS[c + d] . GS[p], Expanding -> False]
```

$$-\overline{p}^2\left(\bar{\gamma}\cdot\overline{a}+\bar{\gamma}\cdot\overline{b}\right).\left(\bar{\gamma}\cdot\overline{c}+\bar{\gamma}\cdot\overline{d}\right)+2\,(\overline{c}\cdot\overline{p})\left(\bar{\gamma}\cdot\overline{a}+\bar{\gamma}\cdot\overline{b}\right).(\bar{\gamma}\cdot\overline{p})+2\left(\overline{d}\cdot\overline{p}\right)\left(\bar{\gamma}\cdot\overline{a}+\bar{\gamma}\cdot\overline{b}\right).(\bar{\gamma}\cdot\overline{p})$$

The matrix chain may also live in D dimensions

```
GAD[\[Mu], \[Nu], \[Mu]]

DiracSimplify[%]
```

$$\gamma^{\mu}.\gamma^{\nu}.\gamma^{\mu}$$

$$2\gamma^{\nu}-D\gamma^{\nu}$$

```
GSD[p] . GAD[\[Alpha], \[Beta]] . GSD[p]

DiracSimplify[%]
```

$$(\gamma \cdot p).\gamma^{\alpha}.\gamma^{\beta}.(\gamma \cdot p)$$

$$p^2\gamma^{\alpha}.\gamma^{\beta} + 2p^{\alpha}\gamma^{\beta}.(\gamma \cdot p) - 2p^{\beta}\gamma^{\alpha}.(\gamma \cdot p)$$

```
GAD @@ Join[{\[Mu]}, Table[Subscript[\[Nu], i], {i, 6}], {\[Mu]}]

DiracSimplify[%]
```

$$\gamma^{\mu}.\gamma^{\nu_1}.\gamma^{\nu_2}.\gamma^{\nu_3}.\gamma^{\nu_4}.\gamma^{\nu_5}.\gamma^{\nu_6}.\gamma^{\mu}$$

$$\begin{aligned}
&-12\gamma^{\nu_1}.\gamma^{\nu_2}.\gamma^{\nu_3}.\gamma^{\nu_4}.\gamma^{\nu_5}.\gamma^{\nu_6} + D\gamma^{\nu_1}.\gamma^{\nu_2}.\gamma^{\nu_3}.\gamma^{\nu_4}.\gamma^{\nu_5}.\gamma^{\nu_6} + 4\gamma^{\nu_3}.\gamma^{\nu_4}.\gamma^{\nu_5}.\gamma^{\nu_6}g^{\nu_1\nu_2} \\
&- 4\gamma^{\nu_2}.\gamma^{\nu_4}.\gamma^{\nu_5}.\gamma^{\nu_6}g^{\nu_1\nu_3} + 4\gamma^{\nu_2}.\gamma^{\nu_3}.\gamma^{\nu_5}.\gamma^{\nu_6}g^{\nu_1\nu_4} - 4\gamma^{\nu_2}.\gamma^{\nu_3}.\gamma^{\nu_4}.\gamma^{\nu_6}g^{\nu_1\nu_5} \\
&+ 4\gamma^{\nu_2}.\gamma^{\nu_3}.\gamma^{\nu_4}.\gamma^{\nu_5}g^{\nu_1\nu_6} + 4\gamma^{\nu_1}.\gamma^{\nu_4}.\gamma^{\nu_5}.\gamma^{\nu_6}g^{\nu_2\nu_3} - 4\gamma^{\nu_1}.\gamma^{\nu_3}.\gamma^{\nu_5}.\gamma^{\nu_6}g^{\nu_2\nu_4} + 4\gamma^{\nu_1}.\gamma^{\nu_3}.\gamma^{\nu_4}.\gamma^{\nu_6}g^{\nu_2\nu_5} \\
&- 4\gamma^{\nu_1}.\gamma^{\nu_3}.\gamma^{\nu_4}.\gamma^{\nu_5}g^{\nu_2\nu_6} + 4\gamma^{\nu_1}.\gamma^{\nu_2}.\gamma^{\nu_5}.\gamma^{\nu_6}g^{\nu_3\nu_4} - 4\gamma^{\nu_1}.\gamma^{\nu_2}.\gamma^{\nu_4}.\gamma^{\nu_6}g^{\nu_3\nu_5} + 4\gamma^{\nu_1}.\gamma^{\nu_2}.\gamma^{\nu_4}.\gamma^{\nu_5}g^{\nu_3\nu_6} \\
&+ 4\gamma^{\nu_1}.\gamma^{\nu_2}.\gamma^{\nu_3}.\gamma^{\nu_6}g^{\nu_4\nu_5} - 4\gamma^{\nu_1}.\gamma^{\nu_2}.\gamma^{\nu_3}.\gamma^{\nu_5}g^{\nu_4\nu_6} + 4\gamma^{\nu_1}.\gamma^{\nu_2}.\gamma^{\nu_3}.\gamma^{\nu_4}g^{\nu_5\nu_6}
\end{aligned}$$

```
-1/2 GA[5] . (GAD[\[Mu]] . GSD[v] - FVD[v, \[Mu]]) FVD[v, \[Mu]]

DiracSimplify[%]
```

$$-\frac{1}{2}v^{\mu}\bar{\gamma}^5.(\gamma^{\mu}.(\gamma \cdot v) - v^{\mu})$$

$$0$$

γ^5 and the chirality projectors are always moved to the right

```
GA[5, \[Mu], \[Nu]]

DiracSimplify[%]
```

$$\bar{\gamma}^5.\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}$$

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^5$$

```
GA[6] . GS[p + q]

DiracSimplify[%]
```

$$\bar{\gamma}^6.(\bar{\gamma}\cdot(\overline{p}+\overline{q}))$$

$$(\bar{\gamma}\cdot\overline{p}).\bar{\gamma}^7+(\bar{\gamma}\cdot\overline{q}).\bar{\gamma}^7$$

The properties of the chirality projectors are taken into account without substituting explicit expressions for γ^6 and γ^7.

```
GA[\[Mu]] . (c1 GA[6] + c2 GA[7]) . (GA[p] + m) . (c3 GA[6] + c4 GA[7]) .
GA[\[Mu]]

DiracSimplify[%]
```

$$\bar{\gamma}^{\mu}.\left(\text{c1}\bar{\gamma}^6+\text{c2}\bar{\gamma}^7\right).(\bar{\gamma}^p+m).\left(\text{c3}\bar{\gamma}^6+\text{c4}\bar{\gamma}^7\right).\bar{\gamma}^{\mu}$$

$$4\ \text{c1}\ \text{c3}m\bar{\gamma}^7-2\ \text{c1}\ \text{c4}\bar{\gamma}^p.\bar{\gamma}^6-2\ \text{c2}\ \text{c3}\bar{\gamma}^p.\bar{\gamma}^7+4\ \text{c2}\ \text{c4}m\bar{\gamma}^6$$

Moreover, $\frac{1}{2}\left(1\pm\gamma^5\right)$ is automatically replaced by $\gamma^{6/7}$.

```
(1/2 - GA[5]/2) . (-((a + GS[p + q])/b)) . (1/2 + GA[5]/2)

DiracSimplify[%]
```

$$\left(\frac{1}{2}-\frac{\bar{\gamma}^5}{2}\right).\left(-\frac{\bar{\gamma}\cdot(\overline{p}+\overline{q})+a}{b}\right).\left(\frac{\bar{\gamma}^5}{2}+\frac{1}{2}\right)$$

$$-\frac{(\bar{\gamma}\cdot\overline{p}).\bar{\gamma}^6}{b}-\frac{(\bar{\gamma}\cdot\overline{q}).\bar{\gamma}^6}{b}$$

Suitable combinations of γ^5 will not be rewritten in terms of chirality projectors, if the option **`ToDiracGamma67`** is set to **`False`**.

```
DiracSimplify[(1/2 - GA[5]/2) . (-((a + GS[p + q])/b)) . (1/2 + GA[5]/2),
  ToDiracGamma67 -> False]
```

$$-\frac{\bar{\gamma}\cdot\overline{p}}{2b}-\frac{(\bar{\gamma}\cdot\overline{p}).\bar{\gamma}^5}{2b}-\frac{\bar{\gamma}\cdot\overline{q}}{2b}-\frac{(\bar{\gamma}\cdot\overline{q}).\bar{\gamma}^5}{2b}$$

However, it the final result must contain only γ^5 but not γ^6 or γ^7, it is better to invoke the option **`DiracSubstitute67`**. This way DiracSimplify can perform more intermediate simplifications before presenting the final result.

```
DiracSimplify[(1/2 - GA[5]/2) . (-((a + GS[p + q])/b)) . (1/2 + GA[5]/2),
  DiracSubstitute67 -> True]
```

$$-\frac{\bar{\gamma}\cdot\overline{p}}{2b}-\frac{(\bar{\gamma}\cdot\overline{p}).\bar{\gamma}^5}{2b}-\frac{\bar{\gamma}\cdot\overline{q}}{2b}-\frac{(\bar{\gamma}\cdot\overline{q}).\bar{\gamma}^5}{2b}$$

It is also possible to eliminate γ^5 by rewriting it in terms of the chirality projectors

```
DiracSimplify[GA[5, \[Mu], \[Nu]], DiracSubstitute5 -> True]
```

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^6-\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^7$$

The Dirac equation is routinely used to simplify closed spinor chains.

```
(SpinorVBar[Subscript[p, 2], Subscript[m, 2]] . (GS[Subscript[p, 1]] +
      Subscript[m, 1]) . SpinorU[Subscript[p, 1], Subscript[m, 1]])

DiracSimplify[%]
```

$$\bar{v}\left(p_2,m_2\right).\left(\bar{\gamma}\cdot\overline{p}_1+m_1\right).u\left(p_1,m_1\right)$$

$$2m_1\left(\varphi(-\overline{p}_2,m_2)\right).\left(\varphi(\overline{p}_1,m_1)\right)$$

```
SpinorVBar[p] . GS[p] . SpinorU[q]

DiracSimplify[%]
```

$$\bar{v}(p).\left(\bar{\gamma}\cdot\overline{p}\right).u(q)$$

$$0$$

Use the option **`DiracEquation`** to deactivate this type of simplifications.

```
DiracSimplify[SpinorVBar[p] . GS[p] . SpinorU[q], DiracEquation -> False]
```

$$(\varphi(-\overline{p})) . (\bar{\gamma} \cdot \overline{p}) . (\varphi(\overline{q}))$$

Suitable products of 4-dimensional spinor chains are simplified using Sirlin's identities

```
(SpinorUBar[Subscript[p, 3], Subscript[m, 3]] . GA[\[Mu], \[Rho], \[Nu], 6]
. SpinorU[Subscript[p, 1],
      Subscript[m, 1]] SpinorUBar[Subscript[p, 4],
      Subscript[m, 4]] . GA[\[Mu], \[Tau], \[Nu], 6] . SpinorU[Subscript[p,
      2], Subscript[m, 2]])

DiracSimplify[%]
```

$$\bar{u}\left(p_3, m_3\right) . \bar{\gamma}^{\mu} . \bar{\gamma}^{\rho} . \bar{\gamma}^{\nu} . \bar{\gamma}^{6} . u\left(p_1, m_1\right) \bar{u}\left(p_4, m_4\right) . \bar{\gamma}^{\mu} . \bar{\gamma}^{\tau} . \bar{\gamma}^{\nu} . \bar{\gamma}^{6} . u\left(p_2, m_2\right)$$

$$(\varphi(\overline{p}_3, m_3)) . \bar{\gamma}^{\mu} . \bar{\gamma}^{\rho} . \bar{\gamma}^{\nu} . \bar{\gamma}^{6} . (\varphi(\overline{p}_1, m_1)) (\varphi(\overline{p}_4, m_4)) . \bar{\gamma}^{\mu} . \bar{\gamma}^{\tau} . \bar{\gamma}^{\nu} . \bar{\gamma}^{6} . (\varphi(\overline{p}_2, m_2))$$

The applications of Sirlin's identities can be disabled by setting the option **`SirlinSimplify`** to **`False`**.

```
DiracSimplify[SpinorUBar[Subscript[p, 3], Subscript[m, 3]] . GA[\[Mu],
\[Rho], \[Nu],
     6] . SpinorU[Subscript[p, 1], Subscript[m, 1]]*
   SpinorUBar[Subscript[p, 4], Subscript[m, 4]] . GA[\[Mu], \[Tau], \[Nu],
     6] . SpinorU[Subscript[p, 2], Subscript[m, 2]], SirlinSimplify ->
     False]
```

$$(\varphi(\overline{p}_3, m_3)) . \bar{\gamma}^{\mu} . \bar{\gamma}^{\rho} . \bar{\gamma}^{\nu} . \bar{\gamma}^{6} . (\varphi(\overline{p}_1, m_1)) (\varphi(\overline{p}_4, m_4)) . \bar{\gamma}^{\mu} . \bar{\gamma}^{\tau} . \bar{\gamma}^{\nu} . \bar{\gamma}^{6} . (\varphi(\overline{p}_2, m_2))$$

Even when the usage of Sirlin's identities is disabled, DiracSimplify will still try to perform some simplifications on the spinor chains, e.g. by canonicalizing the dummy indices.

```
(c1 SpinorUBar[Subscript[p, 3], Subscript[m, 3]] . GA[\[Mu], \[Rho], \[Nu],
6] . SpinorU[Subscript[p,
        1], Subscript[m, 1]] SpinorUBar[Subscript[p, 4], Subscript[m,
        4]] . GA[\[Mu], \[Tau], \[Nu], 6] . SpinorU[Subscript[p, 2],
        Subscript[m, 2]] +
    c2 SpinorUBar[Subscript[p, 3], Subscript[m, 3]] . GA[\[Alpha], \[Rho],
       \[Nu], 6] . SpinorU[Subscript[p, 1], Subscript[m, 1]]
       SpinorUBar[Subscript[p,
        4], Subscript[m, 4]] . GA[\[Alpha], \[Tau], \[Nu], 6] .
        SpinorU[Subscript[p, 2], Subscript[m, 2]])

DiracSimplify[%, SirlinSimplify -> False] // Factor
```

$$\begin{aligned}&\text{c1}\bar{u}\left(p_3,m_3\right).\bar{\gamma}^{\mu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\nu}.\bar{\gamma}^{6}.u\left(p_1,m_1\right)\bar{u}\left(p_4,m_4\right).\bar{\gamma}^{\mu}.\bar{\gamma}^{\tau}.\bar{\gamma}^{\nu}.\bar{\gamma}^{6}.u\left(p_2,\right.\\&\left.m_2\right)+\text{c2}\bar{u}\left(p_3,m_3\right).\bar{\gamma}^{\alpha}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\nu}.\bar{\gamma}^{6}.u\left(p_1,m_1\right)\bar{u}\left(p_4,m_4\right).\bar{\gamma}^{\alpha}.\bar{\gamma}^{\tau}.\bar{\gamma}^{\nu}.\bar{\gamma}^{6}.u\left(p_2,m_2\right)\end{aligned}$$

$$\begin{aligned}&\text{c1}\left(\varphi(\overline{p}_3,m_3)\right).\bar{\gamma}^{\mu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\nu}.\bar{\gamma}^{6}.\left(\varphi(\overline{p}_1,m_1)\right)\left(\varphi(\overline{p}_4,m_4)\right).\bar{\gamma}^{\mu}.\bar{\gamma}^{\tau}.\bar{\gamma}^{\nu}.\bar{\gamma}^{6}.\left(\varphi(\overline{p}_2,\right.\\&\left.m_2\right))+\text{c2}\left(\varphi(\overline{p}_3,m_3)\right).\bar{\gamma}^{\alpha}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\nu}.\bar{\gamma}^{6}.\left(\varphi(\overline{p}_1,m_1)\right)\left(\varphi(\overline{p}_4,m_4)\right).\bar{\gamma}^{\alpha}.\bar{\gamma}^{\tau}.\bar{\gamma}^{\nu}.\bar{\gamma}^{6}.\left(\varphi(\overline{p}_2,m_2)\right)\end{aligned}$$

Setting **SpinorChainTrick** to 'False disables this behavior.

```
DiracSimplify[c1 SpinorUBar[Subscript[p, 3], Subscript[m, 3]] . GA[\[Mu],
\[Rho],
      \[Nu], 6] . SpinorU[Subscript[p, 1], Subscript[m, 1]]
      SpinorUBar[Subscript[p,
       4], Subscript[m, 4]] . GA[\[Mu], \[Tau], \[Nu], 6] .
       SpinorU[Subscript[p, 2],
      Subscript[m, 2]] + c2 SpinorUBar[Subscript[p, 3], Subscript[m,
       3]] . GA[\[Alpha], \[Rho], \[Nu], 6] . SpinorU[Subscript[p, 1],
       Subscript[m,
       1]] SpinorUBar[Subscript[p, 4], Subscript[m, 4]] . GA[\[Alpha],
       \[Tau],
      \[Nu], 6] . SpinorU[Subscript[p, 2], Subscript[m, 2]],
  SirlinSimplify -> False, SpinorChainTrick -> False]
```

$$\begin{aligned}&\text{c1}\left(\varphi(\overline{p}_3,m_3)\right).\bar{\gamma}^{\mu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\nu}.\bar{\gamma}^{6}.\left(\varphi(\overline{p}_1,m_1)\right)\left(\varphi(\overline{p}_4,m_4)\right).\bar{\gamma}^{\mu}.\bar{\gamma}^{\tau}.\bar{\gamma}^{\nu}.\bar{\gamma}^{6}.\left(\varphi(\overline{p}_2,\right.\\&\left.m_2\right))+\text{c2}\left(\varphi(\overline{p}_3,m_3)\right).\bar{\gamma}^{\alpha}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\nu}.\bar{\gamma}^{6}.\left(\varphi(\overline{p}_1,m_1)\right)\left(\varphi(\overline{p}_4,m_4)\right).\bar{\gamma}^{\alpha}.\bar{\gamma}^{\tau}.\bar{\gamma}^{\nu}.\bar{\gamma}^{6}.\left(\varphi(\overline{p}_2,m_2)\right)\end{aligned}$$

DiracSimplify will not reorder Dirac matrices lexicographically, but can be forced to do so via the option **DiracOrder**.

```
DiracSimplify[GA[\[Nu], \[Mu]]]

DiracSimplify[GA[\[Nu], \[Mu]], DiracOrder -> True]
```

$$\bar{\gamma}^{\nu}.\bar{\gamma}^{\mu}$$

$$2\bar{g}^{\mu\nu}-\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}$$

Setting **InsideDiracTrace** to **True**$ makes the function assume that it is acting inside a Dirac trace. For instance, chains with an odd number of Dirac matrices will be set to zero.

```
GA[\[Mu], \[Nu], \[Rho]]

DiracSimplify[%, InsideDiracTrace -> True]
```

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}$$

$$0$$

Yet, it will not explicitly calculate the trace

```
GA[\[Mu], \[Nu], \[Rho], \[Sigma]]

DiracSimplify[%, InsideDiracTrace -> True]
```

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\sigma}$$

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\sigma}$$

Since FeynCalc 9.3, **DiracSimplify** will automatically evaluate Dirac traces in the input expression

```
DiracTrace[GA[\[Mu], \[Nu], \[Rho], \[Sigma]]]

DiracSimplify[%]
```

$$\text{tr}\left(\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\sigma}\right)$$

$$4\bar{g}^{\mu\sigma}\bar{g}^{\nu\rho} - 4\bar{g}^{\mu\rho}\bar{g}^{\nu\sigma} + 4\bar{g}^{\mu\nu}\bar{g}^{\rho\sigma}$$

```
DiracTrace[(-GSD[q] + SMP["m_e"]) . GAD[\[Nu]] . (GSD[p - q] + SMP["m_e"])
. GAD[\[Mu]]]

DiracSimplify[%]
```

$$\text{tr}\left(\left(m_e - \gamma \cdot q\right).\gamma^{\nu}.\left(m_e + \gamma \cdot (p-q)\right).\gamma^{\mu}\right)$$

$$4m_e^2 g^{\mu\nu} + 4g^{\mu\nu}(p\cdot q) - 4q^2 g^{\mu\nu} - 4p^{\nu}q^{\mu} - 4p^{\mu}q^{\nu} + 8q^{\mu}q^{\nu}$$

This will not happen if the option **DiracTraceEvaluate** is set to **False**. However, **DiracSimplify** will still perform some simplifications inside the trace, without evaluating it explicitly.

```
DiracSimplify[DiracTrace[(-GSD[q] + SMP["m_e"]) . GAD[\[Nu]] . (GSD[p - q]
+
      SMP["m_e"]) . GAD[\[Mu]]] , DiracTraceEvaluate -> False]
```

$$\text{tr}\left(m_e^2\gamma^\nu.\gamma^\mu + m_e\gamma^\nu.(\gamma\cdot p).\gamma^\mu - m_e\gamma^\nu.(\gamma\cdot q).\gamma^\mu - m_e(\gamma\cdot q).\gamma^\nu.\gamma^\mu \\ - (\gamma\cdot q).\gamma^\nu.(\gamma\cdot p).\gamma^\mu - q^2\gamma^\nu.\gamma^\mu + 2q^\nu(\gamma\cdot q).\gamma^\mu\right)$$

Set **DiracTrace** to **False** if you want **DiracSimplify** not to touch the Dirac traces.

```
DiracSimplify[DiracTrace[(-GSD[q] + SMP["m_e"]) . GAD[\[Nu]] . (GSD[p - q]
+
      SMP["m_e"]) . GAD[\[Mu]]] , DiracTraceEvaluate -> False, DiracTrace
-> False]
```

$$\text{tr}\left(\left(m_e - \gamma\cdot q\right).\gamma^\nu.\left(m_e + \gamma\cdot(p-q)\right).\gamma^\mu\right)$$

When doing calculations at one loop and above, you may encounter expressions that contain D- and 4-dimensional objects.

Although **DiracSimplify** can handle such terms effortlessly, it will not do so unless you certify that you fully understand what you are doing: being sloppy with the dimensions easily leads to inconsistencies and wrong results!

```
GAD[\[Mu]] . (GA[p] + m) . GAD[\[Mu]]

DiracSimplify[%]
```

$$\gamma^\mu.\left(\bar{\gamma}^p + m\right).\gamma^\mu$$

DiracTrace: Expressions that mix D- , 4- and D- 4- dimensional quantities are forbidden in Dirac matrix chains unless you are using the t'Hooft- Veltman scheme. For every other scheme, please recheck your input expressions and ensure that all matrices, spinors and tensors are purely D- dimensional. You might want to use FCGetDimensions[exp] to find the offending terms and fix them by hand or employ ChangeDimension[exp,D] to convert the whole expression to D- dimensions. If you explicitly intend to use the t'Hooft- Veltman scheme, please activate it via FCSetDiracGammaScheme["BMHV"].

$Aborted

By default, FeynCalc uses the naive dimensional regularization (NDR) scheme, where all Dirac matrices are taken to be D-dimensional. Therefore, in NDR you may not have mixtures of Dirac matrices living in D and 4 dimensions. However, such expressions are possible in the t'Hooft-Veltman-Breitenlohner-Maison (BMHV) scheme.

```
FCSetDiracGammaScheme["BMHV"];

DiracSimplify[GAD[\[Mu]] . (GA[p] + m) . GAD[\[Mu]]]
```

$$-D\bar{\gamma}^p + 2\bar{\gamma}^p + Dm$$

```
FCSetDiracGammaScheme["NDR"];
```

The BMHV scheme is a special prescription for handling γ^5 in dimensional regularization. Do not activate this scheme mindlessly just to get rid of errors from DiracSimplify! If you are doing a calculation in NDR or a calculation that does not involve γ^5, better make sure that your input expressions are correctly written to be D-dimensional objects.

Traces that contain an odd number of γ^5 or chirality projectors cannot be calculated unambiguously in NDR. To avoid inconsistencies, DiracTrace will refuse to evaluate such traces in NDR.

```
DiracTrace[GAD[\[Mu], \[Nu], \[Rho], \[Sigma], \[Alpha], \[Beta]] . GA[5]]

DiracSimplify[%]
```

$$\text{tr}\left(\gamma^\mu.\gamma^\nu.\gamma^\rho.\gamma^\sigma.\gamma^\alpha.\gamma^\beta.\bar{\gamma}^5\right)$$

$$\text{tr}\left(\gamma^\mu.\gamma^\nu.\gamma^\rho.\gamma^\sigma.\gamma^\alpha.\gamma^\beta.\bar{\gamma}^5\right)$$

Such traces can be consistently calculated in the BMHV scheme. Our scheme choice as of course also possible, but those are not implemented in FeynCalc.

```
FCSetDiracGammaScheme["BMHV"];

DiracSimplify[DiracTrace[GAD[\[Mu], \[Nu], \[Rho], \[Sigma], \[Alpha],
\[Beta]] . GA[5]]]
```

$$\begin{aligned}&-4ig^{\alpha\beta}\bar{\epsilon}^{\mu\nu\rho\sigma} - 4ig^{\alpha\mu}\bar{\epsilon}^{\beta\nu\rho\sigma} + 4ig^{\alpha\nu}\bar{\epsilon}^{\beta\mu\rho\sigma} - 4ig^{\alpha\rho}\bar{\epsilon}^{\beta\mu\nu\sigma} + 4ig^{\alpha\sigma}\bar{\epsilon}^{\beta\mu\nu\rho}\\&+ 4ig^{\beta\mu}\bar{\epsilon}^{\alpha\nu\rho\sigma} - 4ig^{\beta\nu}\bar{\epsilon}^{\alpha\mu\rho\sigma} + 4ig^{\beta\rho}\bar{\epsilon}^{\alpha\mu\nu\sigma} - 4ig^{\beta\sigma}\bar{\epsilon}^{\alpha\mu\nu\rho} - 4ig^{\mu\nu}\bar{\epsilon}^{\alpha\beta\rho\sigma}\\&+ 4ig^{\mu\rho}\bar{\epsilon}^{\alpha\beta\nu\sigma} - 4ig^{\mu\sigma}\bar{\epsilon}^{\alpha\beta\nu\rho} - 4ig^{\nu\rho}\bar{\epsilon}^{\alpha\beta\mu\sigma} + 4ig^{\nu\sigma}\bar{\epsilon}^{\alpha\beta\mu\rho} - 4ig^{\rho\sigma}\bar{\epsilon}^{\alpha\beta\mu\nu}\end{aligned}$$

```
FCSetDiracGammaScheme["NDR"];
```

Keep in mind that the BMHV scheme violates axial Ward identities and requires special model-dependent counter-terms to restore those. Therefore, just setting **FCSetDiracGammaScheme["BMHV"]** does not magically resolve all your troubles with γ^5 in D-dimensions. The proper treatment of γ^5 in dimensional regularization is an intricate issue that cannot be boiled down to a simple and universal recipe. FeynCalc merely carries out the algebraic operations that you request, but it is still your task to ensure that what you do makes sense.

Since FeynCalc 9.3 it is also possible to simplify Dirac matrices with Cartesian or temporal indices. However, the support of nonrelativistic calculations is a very new feature, so that things may not work as smooth as they do for manifestly Lorentz covariant expressions.

```
CGA[i] . CGA[i]

DiracSimplify[%]
```

$$\overline{\gamma}^i.\overline{\gamma}^i$$

$$-3$$

```
CGA[i] . CGS[p] . CGA[j] . CGS[p + q]

DiracSimplify[%]
```

$$\overline{\gamma}^i.(\overline{\gamma}\cdot\overline{p}).\overline{\gamma}^j.(\overline{\gamma}\cdot(\overline{p}+\overline{q}))$$

$$\overline{p}^2\overline{\gamma}^i.\overline{\gamma}^j-2\overline{p}^j\overline{\gamma}^i.(\overline{\gamma}\cdot\overline{p})+\overline{\gamma}^i.(\overline{\gamma}\cdot\overline{p}).\overline{\gamma}^j.(\overline{\gamma}\cdot\overline{q})$$

```
CGA[i] . CGS[p] . CGA[j] . CGS[p + q] KD[i, j]

DiracSimplify[%]
```

$$\overline{\delta}^{ij}\overline{\gamma}^i.(\overline{\gamma}\cdot\overline{p}).\overline{\gamma}^j.(\overline{\gamma}\cdot(\overline{p}+\overline{q}))$$

$$(\overline{\gamma}\cdot\overline{p}).(\overline{\gamma}\cdot\overline{q})-\overline{p}^2$$

```
TGA[] . CGA[i] . TGA[]

DiracSimplify[%]
```

$$\bar{\gamma}^0.\bar{\gamma}^i.\bar{\gamma}^0$$

$$-\bar{\gamma}^i$$

```
DiracTrace[CGAD[i, j, k, l]]

DiracSimplify[%]
```

$$\text{tr}\left(\gamma^i.\gamma^j.\gamma^k.\gamma^l\right)$$

$$4\delta^{il}\delta^{jk} - 4\delta^{ik}\delta^{jl} + 4\delta^{ij}\delta^{kl}$$

For performance reasons **DiracSimplify** will not canonically order Dirac matrices and canonicalize Lorentz/Cartesian indices by default. However, amplitudes involving 4-fermion operators may require such additional simplifications. In this case they should explicitly activated by the user. Of course, this will somewhat slow down the evaluation.

```
ex = (Spinor[-Momentum[p1, D], mb, 1] . GAD[\[Mu]] . GA[7] . GAD[\[Nu]] .
GAD[\[Alpha]] .
      GAD[\[Beta]] . GAD[\[Delta]] . GA[7] . Spinor[-Momentum[p4, D], 0, 1]
Spinor[Momentum[p3, D], 0,
       1] . GAD[\[Alpha]] . GAD[\[Beta]] . GAD[\[Delta]] . GA[7] .
       GAD[\[Nu]] . GAD[\[Mu]] .
      GA[7] . Spinor[Momentum[p2, D], 0, 1])

DiracSimplify[ex]
```

$$(\varphi(\text{p3})).\gamma^\alpha.\gamma^\beta.\gamma^\delta.\bar{\gamma}^7.\gamma^\nu.\gamma^\mu.\bar{\gamma}^7.(\varphi(\text{p2}))(\varphi(-\text{p1},\text{mb})).\gamma^\mu.\bar{\gamma}^7.\gamma^\nu.\gamma^\alpha.\gamma^\beta.\gamma^\delta.\bar{\gamma}^7.(\varphi(-\text{p4}))$$

$$(\varphi(\text{p3})).\gamma^\alpha.\gamma^\beta.\gamma^\delta.\gamma^\nu.\gamma^\mu.\bar{\gamma}^7.(\varphi(\text{p2}))(\varphi(-\text{p1},\text{mb})).\gamma^\mu.\gamma^\nu.\gamma^\alpha.\gamma^\beta.\gamma^\delta.\bar{\gamma}^7.(\varphi(-\text{p4}))$$

```
DiracSimplify[ex, DiracOrder -> True, LorentzIndexNames -> {i1, i2, i3, i4,
i5}]
```

$$\begin{aligned}&-24D^2(\varphi(\text{p3})).\gamma^{\text{i1}}.\bar{\gamma}^7.(\varphi(\text{p2}))(\varphi(-\text{p1},\\&\text{mb})).\gamma^{\text{i1}}.\bar{\gamma}^7.(\varphi(-\text{p4})) + 14D(\varphi(\text{p3})).\gamma^{\text{i1}}.\gamma^{\text{i2}}.\gamma^{\text{i3}}.\bar{\gamma}^7.(\varphi(\text{p2}))(\varphi(-\text{p1},\\&\text{mb})).\gamma^{\text{i1}}.\gamma^{\text{i2}}.\gamma^{\text{i3}}.\bar{\gamma}^7.(\varphi(-\text{p4})) + 112D(\varphi(\text{p3})).\gamma^{\text{i1}}.\bar{\gamma}^7.(\varphi(\text{p2}))(\varphi(-\text{p1},\\&\text{mb})).\gamma^{\text{i1}}.\bar{\gamma}^7.(\varphi(-\text{p4})) - (\varphi(\text{p3})).\gamma^{\text{i1}}.\gamma^{\text{i2}}.\gamma^{\text{i3}}.\gamma^{\text{i4}}.\gamma^{\text{i5}}.\bar{\gamma}^7.(\varphi(\text{p2}))(\varphi(-\text{p1},\\&\text{mb})).\gamma^{\text{i1}}.\gamma^{\text{i2}}.\gamma^{\text{i3}}.\gamma^{\text{i4}}.\gamma^{\text{i5}}.\bar{\gamma}^7.(\varphi(-\text{p4})) - 36(\varphi(\text{p3})).\gamma^{\text{i1}}.\gamma^{\text{i2}}.\gamma^{\text{i3}}.\bar{\gamma}^7.(\varphi(\text{p2}))(\varphi(-\text{p1},\\&\text{mb})).\gamma^{\text{i1}}.\gamma^{\text{i2}}.\gamma^{\text{i3}}.\bar{\gamma}^7.(\varphi(-\text{p4})) - 64(\varphi(\text{p3})).\gamma^{\text{i1}}.\bar{\gamma}^7.(\varphi(\text{p2}))(\varphi(-\text{p1},\text{mb})).\gamma^{\text{i1}}.\bar{\gamma}^7.(\varphi(-\text{p4}))\end{aligned}$$

DiracSimplify automatically evaluates suitable spinor products with equal momenta, e.g.

```
ex = SpinorUBar[p, m] . SpinorU[p, m]

DiracSimplify[ex]
```

$$\bar{u}(p,m).u(p,m)$$

$$2m$$

This behavior can be turned off by setting the option **SpinorChainEvaluate** to **False**

```
DiracSimplify[ex, SpinorChainEvaluate -> False]
```

$$(\varphi(\overline{p},m)).(\varphi(\overline{p},m))$$

5.16 DiracSubstitute5

DiracSubstitute5[exp] rewrites γ^5 in terms of the chirality projectors γ^6 and γ^7. **DiracSubstitute5** is also an option of various FeynCalc functions that handle Dirac algebra.

5.16.1 See also

Overview, DiracSubstitute67, DiracGamma, ToDiracGamma67.

5.16.2 Examples

```
GA[5]

DiracSubstitute5[%]
```

$$\bar{\gamma}^5$$

$$\bar{\gamma}^6 - \bar{\gamma}^7$$

```
SpinorUBar[Subscript[p, 1]] . GA[\[Mu]] . GA[5] . GA[\[Nu]] .
SpinorU[Subscript[p, 2]]

DiracSubstitute5[%]
```

$$\bar{u}\left(p_1\right).\bar{\gamma}^{\mu}.\bar{\gamma}^{5}.\bar{\gamma}^{\nu}.u\left(p_2\right)$$

$$\left(\varphi(\overline{p}_1)\right).\bar{\gamma}^{\mu}.\left(\bar{\gamma}^{6}-\bar{\gamma}^{7}\right).\bar{\gamma}^{\nu}.\left(\varphi(\overline{p}_2)\right)$$

5.17 DiracSubstitute67

DiracSubstitute67[exp] inserts the explicit definitions of the chirality projectors γ^6 and γ^7. **DiracSubstitute67** is also an option of various FeynCalc functions that handle Dirac algebra.

5.17.1 See also

Overview, DiracSubstitute5, DiracGamma, ToDiracGamma67.

5.17.2 Examples

```
DiracGamma[6]

DiracSubstitute67[%]
```

$$\bar{\gamma}^{6}$$

$$\frac{\bar{\gamma}^{5}}{2}+\frac{1}{2}$$

```
DiracGamma[7]

DiracSubstitute67[%]
```

$$\bar{\gamma}^{7}$$

$$\frac{1}{2}-\frac{\bar{\gamma}^{5}}{2}$$

```
SpinorUBar[Subscript[p, 1]] . GA[6] . SpinorU[Subscript[p, 2]]

DiracSubstitute67[%]
```

$$\bar{u}\,(p_1)\,.\bar{\gamma}^6.u\,(p_2)$$

$$(\varphi(\overline{p}_1))\,.\left(\frac{\bar{\gamma}^5}{2}+\frac{1}{2}\right).(\varphi(\overline{p}_2))$$

```
SpinorUBar[Subscript[p, 1]] . GA[7] . SpinorU[Subscript[p, 2]]

DiracSubstitute67[%]
```

$$\bar{u}\,(p_1)\,.\bar{\gamma}^7.u\,(p_2)$$

$$(\varphi(\overline{p}_1))\,.\left(\frac{1}{2}-\frac{\bar{\gamma}^5}{2}\right).(\varphi(\overline{p}_2))$$

5.18 DiracTrace

DiracTrace[exp] is the head of Dirac traces. By default the trace is not evaluated. The evaluation occurs only when the option **DiracTraceEvaluate** is set to **True**. It is recommended to use **DiracSimplify**, which will automatically evaluate all Dirac traces in the input expression.

5.18.1 See also

Overview, Contract, DiracEquation, DiracGamma, DiracGammaExpand, DiracTrick, FCGetDiracGammaScheme, FCSetDiracGammaScheme.

5.18.2 Examples

There is no automatic evaluation of Dirac traces

```
DiracTrace[GA[\[Mu], \[Nu]]]
```

$$\text{tr}\,(\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu})$$

```
DiracTrace[GA[\[Mu], \[Nu], \[Rho], \[Sigma]]]
```

$$\text{tr}\,(\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\sigma})$$

You can either set the option **DiracTraceEvaluate** to **True** or use **DiracSimplify**.

```
DiracTrace[GA[\[Mu], \[Nu], \[Rho], \[Sigma]], DiracTraceEvaluate -> True]
```

$$4\left(\bar{g}^{\mu\sigma}\bar{g}^{\nu\rho}-\bar{g}^{\mu\rho}\bar{g}^{\nu\sigma}+\bar{g}^{\mu\nu}\bar{g}^{\rho\sigma}\right)$$

```
DiracSimplify[DiracTrace[GA[\[Mu], \[Nu], \[Rho], \[Sigma]]]]
```

$$4\bar{g}^{\mu\sigma}\bar{g}^{\nu\rho}-4\bar{g}^{\mu\rho}\bar{g}^{\nu\sigma}+4\bar{g}^{\mu\nu}\bar{g}^{\rho\sigma}$$

```
DiracTrace[GS[p, q, r, s]]

DiracSimplify[%]
```

$$\text{tr}\left((\bar{\gamma}\cdot\bar{p}).(\bar{\gamma}\cdot\bar{q}).(\bar{\gamma}\cdot\bar{r}).(\bar{\gamma}\cdot\bar{s})\right)$$

$$4\left(\bar{p}\cdot\bar{s}\right)\left(\bar{q}\cdot\bar{r}\right)-4\left(\bar{p}\cdot\bar{r}\right)\left(\bar{q}\cdot\bar{s}\right)+4\left(\bar{p}\cdot\bar{q}\right)\left(\bar{r}\cdot\bar{s}\right)$$

The old methods of evaluating traces by replacing **`DiracTrace`** with **`Tr`** or **`TR`** are deprecated and should not be used anymore. In particular, they are slower are less efficient than using **`DiracSimplify`**.

Traces involving γ^5 or chirality projectors in 4 dimensions are also possible

```
DiracTrace[GA[\[Mu], \[Nu], \[Rho], \[Sigma], 5]]

DiracSimplify[%]
```

$$\text{tr}\left(\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\sigma}.\bar{\gamma}^{5}\right)$$

$$-4i\bar{\epsilon}^{\mu\nu\rho\sigma}$$

```
DiracTrace[GA[\[Mu], \[Nu], \[Rho], \[Sigma], \[Delta], \[Tau], 5]]

DiracSimplify[%]
```

$$\text{tr}\left(\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\sigma}.\bar{\gamma}^{\delta}.\bar{\gamma}^{\tau}.\bar{\gamma}^{5}\right)$$

$$-4i\bar{g}^{\delta\mu}\bar{\epsilon}^{\nu\rho\sigma\tau}-4i\bar{g}^{\delta\tau}\bar{\epsilon}^{\mu\nu\rho\sigma}-4i\bar{g}^{\mu\tau}\bar{\epsilon}^{\delta\nu\rho\sigma}-4i\bar{g}^{\nu\rho}\bar{\epsilon}^{\delta\mu\sigma\tau}+4i\bar{g}^{\nu\sigma}\bar{\epsilon}^{\delta\mu\rho\tau}-4i\bar{g}^{\rho\sigma}\bar{\epsilon}^{\delta\mu\nu\tau}$$

```
DiracTrace[GA[\[Mu], \[Nu], \[Rho], \[Sigma], \[Delta], \[Tau], 6]]

DiracSimplify[%]
```

$$\text{tr}\left(\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\sigma}.\bar{\gamma}^{\delta}.\bar{\gamma}^{\tau}.\bar{\gamma}^{6}\right)$$

$$\begin{aligned}&-2\bar{g}^{\delta\mu}\bar{g}^{\nu\tau}\bar{g}^{\rho\sigma}+2\bar{g}^{\delta\mu}\bar{g}^{\nu\sigma}\bar{g}^{\rho\tau}-2\bar{g}^{\delta\mu}\bar{g}^{\nu\rho}\bar{g}^{\sigma\tau}+2\bar{g}^{\delta\tau}\bar{g}^{\mu\sigma}\bar{g}^{\nu\rho}+2\bar{g}^{\delta\sigma}\bar{g}^{\mu\tau}\bar{g}^{\nu\rho}-2\bar{g}^{\delta\tau}\bar{g}^{\mu\rho}\bar{g}^{\nu\sigma}-2\bar{g}^{\delta\rho}\bar{g}^{\mu\tau}\bar{g}^{\nu\sigma}\\&-2\bar{g}^{\delta\sigma}\bar{g}^{\mu\rho}\bar{g}^{\nu\tau}+2\bar{g}^{\delta\rho}\bar{g}^{\mu\sigma}\bar{g}^{\nu\tau}+2\bar{g}^{\delta\tau}\bar{g}^{\mu\nu}\bar{g}^{\rho\sigma}+2\bar{g}^{\delta\nu}\bar{g}^{\mu\tau}\bar{g}^{\rho\sigma}+2\bar{g}^{\delta\sigma}\bar{g}^{\mu\nu}\bar{g}^{\rho\tau}-2\bar{g}^{\delta\nu}\bar{g}^{\mu\sigma}\bar{g}^{\rho\tau}-2\bar{g}^{\delta\rho}\bar{g}^{\mu\nu}\bar{g}^{\sigma\tau}\\&+2\bar{g}^{\delta\nu}\bar{g}^{\mu\rho}\bar{g}^{\sigma\tau}-2i\bar{g}^{\delta\mu}\bar{\epsilon}^{\nu\rho\sigma\tau}-2i\bar{g}^{\delta\tau}\bar{\epsilon}^{\mu\nu\rho\sigma}-2i\bar{g}^{\mu\tau}\bar{\epsilon}^{\delta\nu\rho\sigma}-2i\bar{g}^{\nu\rho}\bar{\epsilon}^{\delta\mu\sigma\tau}+2i\bar{g}^{\nu\sigma}\bar{\epsilon}^{\delta\mu\rho\tau}-2i\bar{g}^{\rho\sigma}\bar{\epsilon}^{\delta\mu\nu\tau}\end{aligned}$$

D-dimensional traces that do not involve γ^5 are unambiguous.

```
DiracTrace[(-GSD[q] + SMP["m_e"]) . GAD[\[Nu]] . (GSD[p - q] + SMP["m_e"])
. GAD[\[Mu]]]

DiracSimplify[%]
```

$$\text{tr}\left(\left(m_e-\gamma\cdot q\right).\gamma^{\nu}.\left(m_e+\gamma\cdot(p-q)\right).\gamma^{\mu}\right)$$

$$4m_e^2g^{\mu\nu}+4g^{\mu\nu}(p\cdot q)-4q^2g^{\mu\nu}-4p^{\nu}q^{\mu}-4p^{\mu}q^{\nu}+8q^{\mu}q^{\nu}$$

Traces that contain γ^5 in D dimensions are scheme-dependent. The default scheme used in FeynCalc is the naive dimension regularization (NDR), where γ^5 is assumed to anticommute with all other Dirac matrices. However, chiral traces are ambiguous in NDR, unless the trace contains an even number of γ^5. This is why FeynCalc will leave such traces unevaluated.

```
DiracTrace[GAD[\[Mu], \[Nu], \[Rho]] . GA[5] . GAD[\[Sigma], \[Delta],
\[Tau]] . GA[5]]

DiracSimplify[%]
```

$$\text{tr}\left(\gamma^{\mu}.\gamma^{\nu}.\gamma^{\rho}.\bar{\gamma}^{5}.\gamma^{\sigma}.\gamma^{\delta}.\gamma^{\tau}.\bar{\gamma}^{5}\right)$$

$$\begin{aligned}&-4g^{\delta\tau}g^{\mu\sigma}g^{\nu\rho}-4g^{\delta\sigma}g^{\mu\tau}g^{\nu\rho}+4g^{\delta\mu}g^{\nu\rho}g^{\sigma\tau}+4g^{\delta\tau}g^{\mu\rho}g^{\nu\sigma}+4g^{\delta\rho}g^{\mu\tau}g^{\nu\sigma}\\&+4g^{\delta\sigma}g^{\mu\rho}g^{\nu\tau}-4g^{\delta\rho}g^{\mu\sigma}g^{\nu\tau}-4g^{\delta\tau}g^{\mu\nu}g^{\rho\sigma}-4g^{\delta\nu}g^{\mu\tau}g^{\rho\sigma}+4g^{\delta\mu}g^{\nu\tau}g^{\rho\sigma}\\&-4g^{\delta\sigma}g^{\mu\nu}g^{\rho\tau}+4g^{\delta\nu}g^{\mu\sigma}g^{\rho\tau}-4g^{\delta\mu}g^{\nu\sigma}g^{\rho\tau}+4g^{\delta\rho}g^{\mu\nu}g^{\sigma\tau}-4g^{\delta\nu}g^{\mu\rho}g^{\sigma\tau}\end{aligned}$$

```
DiracTrace[GAD[\[Mu], \[Nu], \[Rho]] . GA[5] . GAD[\[Sigma], \[Delta],
\[Tau]] . GA[7]]

DiracSimplify[%]
```

$$\text{tr}\left(\gamma^{\mu}.\gamma^{\nu}.\gamma^{\rho}.\bar{\gamma}^{5}.\gamma^{\sigma}.\gamma^{\delta}.\gamma^{\tau}.\bar{\gamma}^{7}\right)$$

$$\begin{aligned}
&-\frac{1}{2}\,\text{tr}\left(\gamma^{\mu}.\gamma^{\nu}.\gamma^{\rho}.\gamma^{\sigma}.\gamma^{\delta}.\gamma^{\tau}.\bar{\gamma}^{5}\right) + 2g^{\delta\tau}g^{\mu\sigma}g^{\nu\rho} + 2g^{\delta\sigma}g^{\mu\tau}g^{\nu\rho} - 2g^{\delta\tau}g^{\mu\rho}g^{\nu\sigma} - 2g^{\delta\rho}g^{\mu\tau}g^{\nu\sigma} \\
&- 2g^{\delta\sigma}g^{\mu\rho}g^{\nu\tau} + 2g^{\delta\rho}g^{\mu\sigma}g^{\nu\tau} + 2g^{\delta\tau}g^{\mu\nu}g^{\rho\sigma} + 2g^{\delta\nu}g^{\mu\tau}g^{\rho\sigma} - 2g^{\delta\mu}g^{\nu\tau}g^{\rho\sigma} + 2g^{\delta\sigma}g^{\mu\nu}g^{\rho\tau} \\
&- 2g^{\delta\nu}g^{\mu\sigma}g^{\rho\tau} + 2g^{\delta\mu}g^{\nu\sigma}g^{\rho\tau} - 2g^{\delta\rho}g^{\mu\nu}g^{\sigma\tau} + 2g^{\delta\nu}g^{\mu\rho}g^{\sigma\tau} - 2g^{\delta\mu}g^{\nu\rho}g^{\sigma\tau}
\end{aligned}$$

Over the years people invented many different schemes to deal with γ^5 in dimensional regularization. Currently, only the t'Hooft-Veltman-Breitenlohner-Maison (BMHV) prescription is fully supported in FeynCalc.

```
FCSetDiracGammaScheme["BMHV"];

DiracSimplify[DiracTrace[GAD[\[Mu], \[Nu], \[Rho]] . GA[5] . GAD[\[Sigma],
\[Delta], \[Tau]] . GA[7]]]
```

$$\begin{aligned}
&4i\bar{\epsilon}^{\nu\rho\sigma\tau}\hat{g}^{\delta\mu} + 16\hat{g}^{\nu\tau}\hat{g}^{\rho\sigma}\hat{g}^{\delta\mu} - 8g^{\nu\tau}\hat{g}^{\rho\sigma}\hat{g}^{\delta\mu} - 16\hat{g}^{\nu\sigma}\hat{g}^{\rho\tau}\hat{g}^{\delta\mu} + 8g^{\nu\sigma}\hat{g}^{\rho\tau}\hat{g}^{\delta\mu} \\
&- 8\hat{g}^{\nu\tau}g^{\rho\sigma}\hat{g}^{\delta\mu} + 4g^{\nu\tau}g^{\rho\sigma}\hat{g}^{\delta\mu} + 8\hat{g}^{\nu\sigma}g^{\rho\tau}\hat{g}^{\delta\mu} - 4g^{\nu\sigma}g^{\rho\tau}\hat{g}^{\delta\mu} + 4g^{\nu\rho}g^{\sigma\tau}\hat{g}^{\delta\mu} \\
&- 4i\bar{\epsilon}^{\mu\rho\sigma\tau}\hat{g}^{\delta\nu} + 4i\bar{\epsilon}^{\mu\nu\sigma\tau}\hat{g}^{\delta\rho} - 2i\bar{\epsilon}^{\nu\rho\sigma\tau}g^{\delta\mu} + 2i\bar{\epsilon}^{\mu\rho\sigma\tau}g^{\delta\nu} - 2i\bar{\epsilon}^{\mu\nu\sigma\tau}g^{\delta\rho} + 2i\bar{\epsilon}^{\mu\nu\rho\tau}g^{\delta\sigma} \\
&+ 2i\bar{\epsilon}^{\mu\nu\rho\sigma}g^{\delta\tau} - 4i\bar{\epsilon}^{\delta\nu\rho\tau}\hat{g}^{\mu\sigma} + 4i\bar{\epsilon}^{\delta\nu\rho\sigma}\hat{g}^{\mu\tau} + 2i\bar{\epsilon}^{\delta\rho\sigma\tau}g^{\mu\nu} - 2i\bar{\epsilon}^{\delta\nu\sigma\tau}g^{\mu\rho} + 2i\bar{\epsilon}^{\delta\nu\rho\tau}g^{\mu\sigma} \\
&- 2i\bar{\epsilon}^{\delta\nu\rho\sigma}g^{\mu\tau} + 4i\bar{\epsilon}^{\delta\mu\rho\tau}\hat{g}^{\nu\sigma} + 16\hat{g}^{\delta\rho}\hat{g}^{\mu\tau}\hat{g}^{\nu\sigma} - 8g^{\delta\rho}\hat{g}^{\mu\tau}\hat{g}^{\nu\sigma} + 4g^{\delta\tau}g^{\mu\rho}\hat{g}^{\nu\sigma} \\
&- 8\hat{g}^{\delta\rho}g^{\mu\tau}\hat{g}^{\nu\sigma} + 4g^{\delta\rho}g^{\mu\tau}\hat{g}^{\nu\sigma} - 4i\bar{\epsilon}^{\delta\mu\rho\sigma}\hat{g}^{\nu\tau} - 16\hat{g}^{\delta\rho}\hat{g}^{\mu\sigma}\hat{g}^{\nu\tau} + 8g^{\delta\rho}\hat{g}^{\mu\sigma}\hat{g}^{\nu\tau} \\
&+ 4g^{\delta\sigma}g^{\mu\rho}\hat{g}^{\nu\tau} + 8\hat{g}^{\delta\rho}g^{\mu\sigma}\hat{g}^{\nu\tau} - 4g^{\delta\rho}g^{\mu\sigma}\hat{g}^{\nu\tau} + 2i\bar{\epsilon}^{\delta\mu\sigma\tau}g^{\nu\rho} - 4g^{\delta\tau}\hat{g}^{\mu\sigma}g^{\nu\rho} \\
&- 4g^{\delta\sigma}\hat{g}^{\mu\tau}g^{\nu\rho} + 2g^{\delta\tau}g^{\mu\sigma}g^{\nu\rho} + 2g^{\delta\sigma}g^{\mu\tau}g^{\nu\rho} - 2i\bar{\epsilon}^{\delta\mu\rho\tau}g^{\nu\sigma} - 8\hat{g}^{\delta\rho}\hat{g}^{\mu\tau}g^{\nu\sigma} \\
&+ 4g^{\delta\rho}\hat{g}^{\mu\tau}g^{\nu\sigma} - 2g^{\delta\tau}g^{\mu\rho}g^{\nu\sigma} + 4\hat{g}^{\delta\rho}g^{\mu\tau}g^{\nu\sigma} - 2g^{\delta\rho}g^{\mu\tau}g^{\nu\sigma} + 2i\bar{\epsilon}^{\delta\mu\rho\sigma}g^{\nu\tau} \\
&+ 8\hat{g}^{\delta\rho}\hat{g}^{\mu\sigma}g^{\nu\tau} - 4g^{\delta\rho}\hat{g}^{\mu\sigma}g^{\nu\tau} - 2g^{\delta\sigma}g^{\mu\rho}g^{\nu\tau} - 4\hat{g}^{\delta\rho}g^{\mu\sigma}g^{\nu\tau} + 2g^{\delta\rho}g^{\mu\sigma}g^{\nu\tau} \\
&- 4i\bar{\epsilon}^{\delta\mu\nu\tau}\hat{g}^{\rho\sigma} - 16\hat{g}^{\delta\nu}\hat{g}^{\mu\tau}\hat{g}^{\rho\sigma} + 8g^{\delta\nu}\hat{g}^{\mu\tau}\hat{g}^{\rho\sigma} - 4g^{\delta\tau}g^{\mu\nu}\hat{g}^{\rho\sigma} + 8\hat{g}^{\delta\nu}g^{\mu\tau}\hat{g}^{\rho\sigma} \\
&- 4g^{\delta\nu}g^{\mu\tau}\hat{g}^{\rho\sigma} - 8g^{\delta\mu}\hat{g}^{\nu\tau}\hat{g}^{\rho\sigma} + 4g^{\delta\mu}g^{\nu\tau}\hat{g}^{\rho\sigma} + 4i\bar{\epsilon}^{\delta\mu\nu\sigma}\hat{g}^{\rho\tau} + 16\hat{g}^{\delta\nu}\hat{g}^{\mu\sigma}\hat{g}^{\rho\tau} \\
&- 8g^{\delta\nu}\hat{g}^{\mu\sigma}\hat{g}^{\rho\tau} - 4g^{\delta\sigma}g^{\mu\nu}\hat{g}^{\rho\tau} - 8\hat{g}^{\delta\nu}g^{\mu\sigma}\hat{g}^{\rho\tau} + 4g^{\delta\nu}g^{\mu\sigma}\hat{g}^{\rho\tau} + 8g^{\delta\mu}\hat{g}^{\nu\sigma}\hat{g}^{\rho\tau} \\
&- 4g^{\delta\mu}g^{\nu\sigma}\hat{g}^{\rho\tau} + 2i\bar{\epsilon}^{\delta\mu\nu\tau}g^{\rho\sigma} + 8\hat{g}^{\delta\nu}\hat{g}^{\mu\tau}g^{\rho\sigma} - 4g^{\delta\nu}\hat{g}^{\mu\tau}g^{\rho\sigma} + 2g^{\delta\tau}g^{\mu\nu}g^{\rho\sigma} - 4\hat{g}^{\delta\nu}g^{\mu\tau}g^{\rho\sigma} \\
&+ 2g^{\delta\nu}g^{\mu\tau}g^{\rho\sigma} + 4g^{\delta\mu}\hat{g}^{\nu\tau}g^{\rho\sigma} - 2g^{\delta\mu}g^{\nu\tau}g^{\rho\sigma} - 2i\bar{\epsilon}^{\delta\mu\nu\sigma}g^{\rho\tau} - 8\hat{g}^{\delta\nu}\hat{g}^{\mu\sigma}g^{\rho\tau} + 4g^{\delta\nu}\hat{g}^{\mu\sigma}g^{\rho\tau} \\
&+ 2g^{\delta\sigma}g^{\mu\nu}g^{\rho\tau} + 4\hat{g}^{\delta\nu}g^{\mu\sigma}g^{\rho\tau} - 2g^{\delta\nu}g^{\mu\sigma}g^{\rho\tau} - 4g^{\delta\mu}\hat{g}^{\nu\sigma}g^{\rho\tau} + 2g^{\delta\mu}g^{\nu\sigma}g^{\rho\tau} + 2i\bar{\epsilon}^{\delta\mu\nu\rho}g^{\sigma\tau} \\
&+ 4\hat{g}^{\delta\rho}g^{\mu\nu}g^{\sigma\tau} - 2g^{\delta\rho}g^{\mu\nu}g^{\sigma\tau} - 4\hat{g}^{\delta\nu}g^{\mu\rho}g^{\sigma\tau} + 2g^{\delta\nu}g^{\mu\rho}g^{\sigma\tau} - 2g^{\delta\mu}g^{\nu\rho}g^{\sigma\tau}
\end{aligned}$$

Keep in mind that the BMHV scheme violates axial Ward identities and requires special model-dependent counter-terms to restore those. Therefore, just setting FCSetDiracGammaScheme["BMHV"] does not automatically resolve all your troubles with γ^5 in D-dimensions. The proper treatment of γ^5 in dimensional

regularization is an intricate issue that cannot be boiled down to a simple and universal recipe. FeynCalc merely carries out the algebraic operations that you request, but it is still your task to ensure that what you do makes sense.

Traces that are free of γ^5 but contain both 4- and D-dimensional Dirac matrices may appear in calculations that use the BMHV prescription, but they do not make sense in NDR. Therefore, their evaluation will be successful only if the correct scheme is used.

```
FCSetDiracGammaScheme["NDR"];
```

```
DiracTrace[(-GSD[q] + SMP["m_e"]) . GA[\[Nu]] . (GS[p] - GSD[q] +
SMP["m_e"]) . GA[\[Mu]]]

DiracSimplify[%]
```

$$\text{tr}\left((m_e - \gamma\cdot q)\,.\bar{\gamma}^{\nu}.\,(\bar{\gamma}\cdot\overline{p} + m_e - \gamma\cdot q)\,.\bar{\gamma}^{\mu}\right)$$

DiracTrace: Expressions that mix D-, 4- and D-4-dimensional quantities are forbidden in Dirac matrix chains unless you are using the t'Hooft-Veltman scheme. For every other scheme, please recheck your input expressions and ensure that all matrices, spinors and tensors are purely D-dimensional. You might want to use FCGetDimensions[exp] to find the offending terms and fix them by hand or employ ChangeDimension[exp,D] to convert the whole expression to D-dimensions. If you explicitly intend to use the t'Hooft-Veltman scheme, please activate it via FCSetDiracGammaScheme["BMHV"].

$Aborted

```
FCSetDiracGammaScheme["BMHV"];
```

```
ex = DiracSimplify[DiracTrace[(-GSD[q] + SMP["m_e"]) . GA[\[Nu]] . (GS[p] -
GSD[q] + SMP["m_e"]) . GA[\[Mu]]] ]
```

$$4m_e^2\bar{g}^{\mu\nu} + 4\bar{g}^{\mu\nu}\,(\overline{p}\cdot\overline{q}) - 4q^2\bar{g}^{\mu\nu} - 4\overline{p}^{\nu}\overline{q}^{\mu} - 4\overline{p}^{\mu}\overline{q}^{\nu} + 8\overline{q}^{\mu}\overline{q}^{\nu}$$

```
ex // FCE // StandardForm

(*-4 FV[p, \[Nu]] FV[q, \[Mu]] - 4 FV[p, \[Mu]] FV[q, \[Nu]] + 8 FV[q,
\[Mu]] FV[q, \[Nu]] + 4 MT[\[Mu], \[Nu]] SMP["m_e"]^2 + 4 MT[\[Mu], \[Nu]]
SP[p, q] - 4 MT[\[Mu], \[Nu]] SPD[q, q]*)
```

```
FCSetDiracGammaScheme["NDR"];
```

Notice that in this case the result contains 4- and D-dimensional tensors.

Traces involving γ^5 in the BMHV scheme are evaluated using West's formula. It is possible to turn it off by setting the option **`West`** to **`False`**, but then the evaluation will require much more time.

```
FCSetDiracGammaScheme["BMHV"];

AbsoluteTiming[r1 = DiracSimplify[DiracTrace[GAD[\[Mu], \[Nu], \[Rho]] .
GA[5] . GAD[\[Sigma], \[Delta], \[Tau]] . GA[7]]];]
```

$$\{0.225294, \text{Null}\}$$

```
AbsoluteTiming[r2 = DiracSimplify[DiracTrace[GAD[\[Mu], \[Nu], \[Rho]] .
GA[5] . GAD[\[Sigma], \[Delta], \[Tau]] . GA[7],
      West -> False]];]
```

$$\{1.82085, \text{Null}\}$$

```
r1 === r2
```

True

```
FCSetDiracGammaScheme["NDR"];

ClearAll[r1, r2]
```

If you know that traces with one γ^5 do not contribute to your final result, use the new NDR-Discard scheme to put them to zero

```
FCSetDiracGammaScheme["NDR-Discard"];

DiracSimplify[DiracTrace[GAD[\[Mu], \[Nu], \[Rho]] . GA[5] . GAD[\[Sigma],
\[Delta], \[Tau]] . GA[7]]]
```

$$\begin{aligned}&2g^{\delta\tau}g^{\mu\sigma}g^{\nu\rho} + 2g^{\delta\sigma}g^{\mu\tau}g^{\nu\rho} - 2g^{\delta\mu}g^{\nu\rho}g^{\sigma\tau} - 2g^{\delta\tau}g^{\mu\rho}g^{\nu\sigma} - 2g^{\delta\rho}g^{\mu\tau}g^{\nu\sigma}\\ &- 2g^{\delta\sigma}g^{\mu\rho}g^{\nu\tau} + 2g^{\delta\rho}g^{\mu\sigma}g^{\nu\tau} + 2g^{\delta\tau}g^{\mu\nu}g^{\rho\sigma} + 2g^{\delta\nu}g^{\mu\tau}g^{\rho\sigma} - 2g^{\delta\mu}g^{\nu\tau}g^{\rho\sigma}\\ &+ 2g^{\delta\sigma}g^{\mu\nu}g^{\rho\tau} - 2g^{\delta\nu}g^{\mu\sigma}g^{\rho\tau} + 2g^{\delta\mu}g^{\nu\sigma}g^{\rho\tau} - 2g^{\delta\rho}g^{\mu\nu}g^{\sigma\tau} + 2g^{\delta\nu}g^{\mu\rho}g^{\sigma\tau}\end{aligned}$$

```
FCSetDiracGammaScheme["NDR"];
```

Sorting of the matrices inside 4-dimensional traces helps to avoid some spurious terms.

```
DiracTrace[GA[\[Mu], \[Nu], 5, \[Rho], \[Sigma], \[Tau], \[Kappa]],
DiracTraceEvaluate -> True] -
   DiracTrace[GA[\[Mu], \[Nu], \[Rho], \[Sigma], \[Tau], \[Kappa],
5],DiracTraceEvaluate -> True] // Expand
```

$$0$$

When the sorting is turned off via **Sort** to **True**, one may obtain some spurious terms that vanish by Schouten's identity.

```
DiracTrace[GA[\[Mu], \[Nu], 5, \[Rho], \[Sigma], \[Tau], \[Kappa]],
DiracTraceEvaluate -> True, Sort -> False] -
   DiracTrace[GA[\[Mu], \[Nu], \[Rho], \[Sigma], \[Tau], \[Kappa],
5],DiracTraceEvaluate -> True, Sort -> False] // Expand
```

$$4i\bar{g}^{\kappa\mu}\bar{\epsilon}^{\nu\rho\sigma\tau} - 4i\bar{g}^{\kappa\nu}\bar{\epsilon}^{\mu\rho\sigma\tau} - 4i\bar{g}^{\kappa\sigma}\bar{\epsilon}^{\mu\nu\rho\tau} + 4i\bar{g}^{\kappa\tau}\bar{\epsilon}^{\mu\nu\rho\sigma} + 4i\bar{g}^{\mu\rho}\bar{\epsilon}^{\kappa\nu\sigma\tau} - 4i\bar{g}^{\nu\rho}\bar{\epsilon}^{\kappa\mu\sigma\tau} + 4i\bar{g}^{\rho\sigma}\bar{\epsilon}^{\kappa\mu\nu\tau} - 4i\bar{g}^{\rho\tau}\bar{\epsilon}^{\kappa\mu\nu\sigma}$$

The trace of the unit matrix in the Dirac space is fixed to 4, which is the standard choice in dimensional regularization.

```
DiracTrace[1]

DiracSimplify[%]
```

$$\text{tr}(1)$$

$$4$$

If, for some reason, this value must be modified, one can do so using the option **TraceOfOne**.

```
DiracTrace[1, TraceOfOne -> D, DiracTraceEvaluate -> True]
```

$$D$$

```
DiracSimplify[DiracTrace[GAD[\[Mu], \[Nu]], TraceOfOne -> D]]
```

$$Dg^{\mu\nu}$$

Since FeynCalc 9.3 it is also possible to compute traces of Dirac matrices with Cartesian or temporal indices. However, the support of nonrelativistic calculations is a very new feature, so that things may not work as smooth as they do for manifestly Lorentz covariant expressions.

```
DiracTrace[CGAD[i, j, k, l]]

DiracSimplify[%]
```

$$\text{tr}\left(\gamma^i.\gamma^j.\gamma^k.\gamma^l\right)$$

$$4\delta^{il}\delta^{jk} - 4\delta^{ik}\delta^{jl} + 4\delta^{ij}\delta^{kl}$$

```
DiracTrace[CGA[i, j, k, l] . GA[6] . CGA[m, n]]

DiracSimplify[%]
```

$$\text{tr}\left(\bar{\gamma}^i.\bar{\gamma}^j.\bar{\gamma}^k.\bar{\gamma}^l.\bar{\gamma}^6.\bar{\gamma}^m.\bar{\gamma}^n\right)$$

$$-2\bar{\delta}^{in}\bar{\delta}^{jm}\bar{\delta}^{kl} + 2\bar{\delta}^{im}\bar{\delta}^{jn}\bar{\delta}^{kl} - 2\bar{\delta}^{ij}\bar{\delta}^{kl}\bar{\delta}^{mn} + 2\bar{\delta}^{in}\bar{\delta}^{jl}\bar{\delta}^{km} - 2\bar{\delta}^{il}\bar{\delta}^{jn}\bar{\delta}^{km} - 2\bar{\delta}^{im}\bar{\delta}^{jl}\bar{\delta}^{kn} + 2\bar{\delta}^{il}\bar{\delta}^{jm}\bar{\delta}^{kn} - 2\bar{\delta}^{in}\bar{\delta}^{jk}\bar{\delta}^{lm} + 2\bar{\delta}^{ik}\bar{\delta}^{jn}\bar{\delta}^{lm} - 2\bar{\delta}^{ij}\bar{\delta}^{kn}\bar{\delta}^{lm} + 2\bar{\delta}^{im}\bar{\delta}^{jk}\bar{\delta}^{ln} - 2\bar{\delta}^{ik}\bar{\delta}^{jm}\bar{\delta}^{ln} + 2\bar{\delta}^{ij}\bar{\delta}^{km}\bar{\delta}^{ln} - 2\bar{\delta}^{il}\bar{\delta}^{jk}\bar{\delta}^{mn} + 2\bar{\delta}^{ik}\bar{\delta}^{jl}\bar{\delta}^{mn}$$

5.19 DiracTrick

`DiracTrick[exp]` contracts Dirac matrices with each other and performs several simplifications but no expansions.There are not many cases when a user will need to call this function directly. Use **`DiracSimplify`** to achieve maximal simplification of Dirac matrix chains. Regarding the treatment of γ^5 in D-dimensional expressions or the evaluation of expressions with tensors living in different dimensions, see the explanations on the help pages for **`DiracSimplify`** and **`DiracTrace`**.

5.19.1 See also

Overview, Contract, DiracEquation, DiracGamma, DiracGammaExpand, DiracTrick, SirlinSimplify, SpinorChainTrick.

5.19.2 Examples

When applied to chains of Dirac matrices that do not require noncommutative expansions, contractions with other tensors, simplifications of spinor chains or evaluations of Dirac traces, **DiracTrick** will produce results similar to those of **DiracSimplify**.

```
GA[\[Mu], \[Nu], \[Mu]]

DiracTrick[%]
```

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\mu}$$

$$-2\bar{\gamma}^{\nu}$$

```
GS[p] . GS[p]

DiracTrick[%]
```

$$(\bar{\gamma}\cdot\overline{p}).(\bar{\gamma}\cdot\overline{p})$$

$$\overline{p}^2$$

```
GA[5, \[Mu], \[Nu]]

DiracTrick[%]
```

$$\bar{\gamma}^{5}.\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}$$

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{5}$$

```
(1/2 - GA[5]/2) . (-((a + GS[p + q])/b)) . (1/2 + GA[5]/2)

DiracTrick[%]
```

$$\left(\frac{1}{2}-\frac{\bar{\gamma}^{5}}{2}\right).\left(-\frac{\bar{\gamma}\cdot(\overline{p}+\overline{q})+a}{b}\right).\left(\frac{\bar{\gamma}^{5}}{2}+\frac{1}{2}\right)$$

$$-\frac{(\bar{\gamma}\cdot(\overline{p}+\overline{q})).\bar{\gamma}^{6}}{b}$$

Dirac traces are not evaluated by **DiracTrick**

```
DiracTrace[GAD[\[Mu], \[Nu]]]

DiracTrick[%]
```

$$\text{tr}\left(\gamma^{\mu}.\gamma^{\nu}\right)$$

$$\text{tr}\left(\gamma^{\mu}.\gamma^{\nu}\right)$$

5.20 EpsChisholm

`EpsChisholm[exp]` applies the Chisholm identity to a Dirac matrix contracted with a Levi-Civita tensor.

5.20.1 See also

Overview, Chisholm, Eps, DiracGamma.

5.20.2 Examples

```
LC[\[Mu], \[Nu], \[Rho], \[Sigma]] GA[\[Sigma], 5]

EpsChisholm[%]
```

$$\bar{\gamma}^{\sigma}.\bar{\gamma}^{5}\bar{\epsilon}^{\mu\nu\rho\sigma}$$

$$-i\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}+i\bar{\gamma}^{\rho}\bar{g}^{\mu\nu}-i\bar{\gamma}^{\nu}\bar{g}^{\mu\rho}+i\bar{\gamma}^{\mu}\bar{g}^{\nu\rho}$$

This reproduces the identities given in the Appendix A of arXiv:2111.05153

```
LC[\[Alpha], \[Nu], \[Beta], \[Rho]] FV[Subscript[p, 1], \[Beta]]
SpinorUBar[Subscript[p, 2], SMP["m_s"]] . GA[\[Alpha], 7] .
SpinorV[Subscript[p, 1], SMP["m_d"]]

% // EpsChisholm // DiracSimplify // Contract
```

$$\overline{p}_1{}^{\beta}\bar{\epsilon}^{\alpha\nu\beta\rho}\bar{u}\left(p_2,m_s\right).\bar{\gamma}^{\alpha}.\bar{\gamma}^{7}.v\left(p_1,m_d\right)$$

$$im_d\bar{g}^{\nu\rho}\left(\varphi(\overline{p}_2,m_s)\right).\bar{\gamma}^{6}.\left(\varphi(-\overline{p}_1,m_d)\right)+i\overline{p}_1{}^{\nu}\left(\varphi(\overline{p}_2,m_s)\right).\bar{\gamma}^{\rho}.\bar{\gamma}^{7}.\left(\varphi(-\overline{p}_1,m_d)\right)-i\overline{p}_1{}^{\rho}\left(\varphi(\overline{p}_2,m_s)\right).\bar{\gamma}^{\nu}.\bar{\gamma}^{7}.\left(\varphi(-\overline{p}_1,m_d)\right)-im_d\left(\varphi(\overline{p}_2,m_s)\right).\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{6}.\left(\varphi(-\overline{p}_1,m_d)\right)$$

```
LC[\[Alpha], \[Nu], \[Beta], \[Rho]] FV[Subscript[p, 2], \[Beta]]
SpinorUBar[Subscript[p, 2], SMP["m_s"]] . GA[\[Alpha], 7] .
SpinorV[Subscript[p, 1], SMP["m_d"]]

% // EpsChisholm // DiracSimplify // Contract
```

$$\overline{p}_2{}^{\beta}\bar{\epsilon}^{\alpha\nu\beta\rho}\bar{u}\left(p_2,m_s\right).\bar{\gamma}^{\alpha}.\bar{\gamma}^{7}.v\left(p_1,m_d\right)$$

$$-im_s\bar{g}^{\nu\rho}\left(\varphi(\overline{p}_2,m_s)\right).\bar{\gamma}^{7}.\left(\varphi(-\overline{p}_1,m_d)\right)-i\overline{p}_2{}^{\nu}\left(\varphi(\overline{p}_2,m_s)\right).\bar{\gamma}^{\rho}.\bar{\gamma}^{7}.\left(\varphi(-\overline{p}_1,\\ m_d)\right)+i\overline{p}_2{}^{\rho}\left(\varphi(\overline{p}_2,m_s)\right).\bar{\gamma}^{\nu}.\bar{\gamma}^{7}.\left(\varphi(-\overline{p}_1,m_d)\right)+im_s\left(\varphi(\overline{p}_2,m_s)\right).\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{7}.\left(\varphi(-\overline{p}_1,m_d)\right)$$

```
LC[\[Alpha], \[Nu], \[Gamma], \[Rho]] FV[Subscript[p, 3], \[Gamma]]
SpinorUBar[Subscript[p, 3], SMP["m_s"]] . GA[\[Nu], 7] .
SpinorV[Subscript[p, 4], SMP["m_d"]]

% // EpsChisholm // DiracSimplify // Contract
```

$$\overline{p}_3{}^{\gamma}\bar{\epsilon}^{\alpha\nu\gamma\rho}\bar{u}\left(p_3,m_s\right).\bar{\gamma}^{\nu}.\bar{\gamma}^{7}.v\left(p_4,m_d\right)$$

$$im_s\bar{g}^{\alpha\rho}\left(\varphi(\overline{p}_3,m_s)\right).\bar{\gamma}^{7}.\left(\varphi(-\overline{p}_4,m_d)\right)+i\overline{p}_3{}^{\alpha}\left(\varphi(\overline{p}_3,m_s)\right).\bar{\gamma}^{\rho}.\bar{\gamma}^{7}.\left(\varphi(-\overline{p}_4,\\ m_d)\right)-i\overline{p}_3{}^{\rho}\left(\varphi(\overline{p}_3,m_s)\right).\bar{\gamma}^{\alpha}.\bar{\gamma}^{7}.\left(\varphi(-\overline{p}_4,m_d)\right)-im_s\left(\varphi(\overline{p}_3,m_s)\right).\bar{\gamma}^{\alpha}.\bar{\gamma}^{\rho}.\bar{\gamma}^{7}.\left(\varphi(-\overline{p}_4,m_d)\right)$$

```
LC[\[Beta], \[Gamma], \[Mu], \[Nu]] FV[Subscript[p, 2], \[Gamma]]
SpinorUBar[Subscript[p, 3], SMP["m_s"]] . GA[\[Beta], 7] .
SpinorV[Subscript[p, 4], SMP["m_d"]]

% // EpsChisholm // DiracSimplify // Contract
```

$$\overline{p}_2{}^{\gamma}\bar{\epsilon}^{\beta\gamma\mu\nu}\bar{u}\left(p_3,m_s\right).\bar{\gamma}^{\beta}.\bar{\gamma}^{7}.v\left(p_4,m_d\right)$$

$$i\bar{g}^{\mu\nu}\left(\varphi(\overline{p}_3,m_s)\right).\left(\bar{\gamma}\cdot\overline{p}_2\right).\bar{\gamma}^{7}.\left(\varphi(-\overline{p}_4,m_d)\right)+i\overline{p}_2{}^{\mu}\left(\varphi(\overline{p}_3,m_s)\right).\bar{\gamma}^{\nu}.\bar{\gamma}^{7}.\left(\varphi(-\overline{p}_4,\\ m_d)\right)-i\overline{p}_2{}^{\nu}\left(\varphi(\overline{p}_3,m_s)\right).\bar{\gamma}^{\mu}.\bar{\gamma}^{7}.\left(\varphi(-\overline{p}_4,m_d)\right)-i\left(\varphi(\overline{p}_3,m_s)\right).\left(\bar{\gamma}\cdot\overline{p}_2\right).\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{7}.\left(\varphi(-\overline{p}_4,m_d)\right)$$

5.21 FCCCT

FCCCT is an alias for **FCChargeConjugateTransposed**.

5.21.1 See also

Overview, FCChargeConjugateTransposed.

5.21.2 Examples

5.22 FCChargeConjugateTransposed

FCChargeConjugateTransposed[exp] represents the application of the charge conjugation operator to the transposed of **exp**, i.e. $C^{-1}\ \exp^T C$. Here **exp** is understood to be a single Dirac matrix or a chain thereof. The option setting **Explicit** determines whether the explicit result is returned or whether it is left in the unevaluated form.The unevaluated form will be also maintained if the function does not know how to obtain $C^{-1}\ \exp^T C$ from the given exp.

The shortcut for **FCChargeConjugateTransposed** is **FCCCT**.

5.22.1 See also

Overview, SpinorChainTranspose, DiracGamma, Spinor.

5.22.2 Examples

```
GA[\[Mu], \[Nu], \[Rho]]

FCChargeConjugateTransposed[%]
```

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}$$

$$C\ (\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho})^T\ C^{-1}$$

```
FCChargeConjugateTransposed[GA[\[Mu], \[Nu], \[Rho]], Explicit -> True]
```

$$-\bar{\gamma}^{\rho}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\mu}$$

```
GA[5]

FCCCT[%]

% // Explicit
```

$$\bar{\gamma}^5$$

$$C\left(\bar{\gamma}^5\right)^T C^{-1}$$

$$\bar{\gamma}^5$$

5.23 FCDiracIsolate

FCDiracIsolate[exp] wraps chains of Dirac matrices into heads specified by the user.

5.23.1 See also

Overview, DiracGamma, Spinor.

5.23.2 Examples

FCDiracIsolate provides an easy way to extract the Dirac structures present in the expression (e.g. an amplitude)

```
amp = (Spinor[Momentum[p2], SMP["m_u"], 1] . (-I GA[\[Mu]] SMP["g_s"]
SUNTF[{Glu2},
         Col3, Col5]) . (GS[-k1 + p2] + SMP["m_u"]) . (-I GA[\[Nu]]
SMP["g_s"] SUNTF[{Glu4},
         Col5, Col1]) . Spinor[Momentum[p1], SMP["m_u"], 1] FAD[{k1 - p2,
       SMP["m_u"]}, Dimension -> 4] FV[Polarization[k1, I], \[Mu]]
FV[Polarization[k2, -I],
      \[Nu]] + Spinor[Momentum[p2], SMP["m_u"], 1] . (-I GA[\[Nu]]
      SMP["g_s"] SUNTF[{Glu4},
         Col3, Col5]) . (GS[k2 + p2] + SMP["m_u"]) . (-I GA[\[Mu]]
SMP["g_s"] SUNTF[{Glu2},
         Col5, Col1]) . Spinor[Momentum[p1], SMP["m_u"], 1] FAD[{-k2 - p2,
SMP["m_u"]},
      Dimension -> 4] FV[Polarization[k1, I], \[Mu]] FV[Polarization[k2,
       -I], \[Nu]] - Spinor[Momentum[p2], SMP["m_u"], 1] . (-I GA[Lor3]
       SMP["g_s"] SUNTF[{Glu5},
         Col3, Col1]) . Spinor[Momentum[p1], SMP["m_u"], 1] FAD[-k1 + k2,
      Dimension -> 4] FV[Polarization[k1, I], \[Mu]] FV[Polarization[k2,
-I],
      \[Nu]] MT[Lor3, Lor4] (FV[2 k1 - k2, \[Nu]] MT[Lor4, \[Mu]] + FV[-k1
      + 2 k2, \[Mu]] MT[Lor4,
         \[Nu]] + FV[-k1 - k2, Lor4] MT[\[Mu], \[Nu]]) SMP["g_s"]
         SUNF[Glu2, Glu4, Glu5])
```

$$\frac{\bar{\varepsilon}^{\mu}(\mathrm{k1})\bar{\varepsilon}^{*\nu}(\mathrm{k2})\left(\varphi(\overline{\mathrm{p2}},m_u)\right).\left(-ig_s\bar{\gamma}^{\mu}T^{\mathrm{Glu2}}_{\mathrm{Col3}\,\mathrm{Col5}}\right).\left(\bar{\gamma}\cdot\left(\overline{\mathrm{p2}}-\overline{\mathrm{k1}}\right)+m_u\right).\left(-ig_s\bar{\gamma}^{\nu}T^{\mathrm{Glu4}}_{\mathrm{Col5}\,\mathrm{Col1}}\right).\left(\varphi(\overline{\mathrm{p1}},m_u)\right)}{(\overline{\mathrm{k1}}-\overline{\mathrm{p2}})^2-m_u^2}$$
$$+\frac{\bar{\varepsilon}^{\mu}(\mathrm{k1})\bar{\varepsilon}^{*\nu}(\mathrm{k2})\left(\varphi(\overline{\mathrm{p2}},m_u)\right).\left(-ig_s\bar{\gamma}^{\nu}T^{\mathrm{Glu4}}_{\mathrm{Col3}\,\mathrm{Col5}}\right).\left(\bar{\gamma}\cdot\left(\overline{\mathrm{k2}}+\overline{\mathrm{p2}}\right)+m_u\right).\left(-ig_s\bar{\gamma}^{\mu}T^{\mathrm{Glu2}}_{\mathrm{Col5}\,\mathrm{Col1}}\right).\left(\varphi(\overline{\mathrm{p1}},m_u)\right)}{(-\overline{\mathrm{k2}}-\overline{\mathrm{p2}})^2-m_u^2}$$
$$-\frac{1}{(\overline{\mathrm{k2}}-\overline{\mathrm{k1}})^2}g_s\bar{g}^{\mathrm{Lor3}\,\mathrm{Lor4}}\bar{\varepsilon}^{\mu}(\mathrm{k1})\bar{\varepsilon}^{*\nu}(\mathrm{k2})f^{\mathrm{Glu2}\,\mathrm{Glu4}\,\mathrm{Glu5}}\left(\bar{g}^{\mathrm{Lor4}\mu}\left(2\overline{\mathrm{k1}}-\overline{\mathrm{k2}}\right)^{\nu}+\bar{g}^{\mathrm{Lor4}\nu}\left(2\overline{\mathrm{k2}}-\overline{\mathrm{k1}}\right)^{\mu}\right.$$
$$\left.+\bar{g}^{\mu\nu}\left(-\overline{\mathrm{k1}}-\overline{\mathrm{k2}}\right)^{\mathrm{Lor4}}\right)\left(\varphi(\overline{\mathrm{p2}},m_u)\right).\left(-ig_s\bar{\gamma}^{\mathrm{Lor3}}T^{\mathrm{Glu5}}_{\mathrm{Col3}\,\mathrm{Col1}}\right).\left(\varphi(\overline{\mathrm{p1}},m_u)\right)$$

```
ampIso = FCDiracIsolate[amp, Head -> diracS]
```

$$-\frac{g_s^2\bar{\varepsilon}^{\mu}(\mathrm{k1})\bar{\varepsilon}^{*\nu}(\mathrm{k2})T^{\mathrm{Glu2}}_{\mathrm{Col5}\,\mathrm{Col1}}T^{\mathrm{Glu4}}_{\mathrm{Col3}\,\mathrm{Col5}}\,\mathrm{diracS}\left(\left(\varphi(\overline{\mathrm{p2}},m_u)\right).\bar{\gamma}^{\nu}.\left(\bar{\gamma}\cdot\left(\overline{\mathrm{k2}}+\overline{\mathrm{p2}}\right)+m_u\right).\bar{\gamma}^{\mu}.\left(\varphi(\overline{\mathrm{p1}},m_u)\right)\right)}{(-\overline{\mathrm{k2}}-\overline{\mathrm{p2}})^2-m_u^2}$$
$$-\frac{g_s^2\bar{\varepsilon}^{\mu}(\mathrm{k1})\bar{\varepsilon}^{*\nu}(\mathrm{k2})T^{\mathrm{Glu4}}_{\mathrm{Col5}\,\mathrm{Col1}}T^{\mathrm{Glu2}}_{\mathrm{Col3}\,\mathrm{Col5}}\,\mathrm{diracS}\left(\left(\varphi(\overline{\mathrm{p2}},m_u)\right).\bar{\gamma}^{\mu}.\left(\bar{\gamma}\cdot\left(\overline{\mathrm{p2}}-\overline{\mathrm{k1}}\right)+m_u\right).\bar{\gamma}^{\nu}.\left(\varphi(\overline{\mathrm{p1}},m_u)\right)\right)}{(\overline{\mathrm{k1}}-\overline{\mathrm{p2}})^2-m_u^2}$$
$$+\frac{1}{(\overline{\mathrm{k2}}-\overline{\mathrm{k1}})^2}ig_s^2\bar{g}^{\mathrm{Lor3}\,\mathrm{Lor4}}\bar{\varepsilon}^{\mu}(\mathrm{k1})\bar{\varepsilon}^{*\nu}(\mathrm{k2})T^{\mathrm{Glu5}}_{\mathrm{Col3}\,\mathrm{Col1}}f^{\mathrm{Glu2}\,\mathrm{Glu4}\,\mathrm{Glu5}}\left(\bar{g}^{\mu\nu}\left(-\left(\overline{\mathrm{k1}}+\overline{\mathrm{k2}}\right)^{\mathrm{Lor4}}\right)\right.$$
$$\left.-\bar{g}^{\mathrm{Lor4}\nu}\left(\overline{\mathrm{k1}}-2\overline{\mathrm{k2}}\right)^{\mu}+\bar{g}^{\mathrm{Lor4}\mu}\left(2\overline{\mathrm{k1}}-\overline{\mathrm{k2}}\right)^{\nu}\right)\,\mathrm{diracS}\left(\left(\varphi(\overline{\mathrm{p2}},m_u)\right).\bar{\gamma}^{\mathrm{Lor3}}.\left(\varphi(\overline{\mathrm{p1}},m_u)\right)\right)$$

Now that all Dirac structures are wrapped into the head **`diracS`** it is easy to extract them to a separate list

```
Cases2[ampIso, diracS]
```

$$\left\{\mathrm{diracS}\left(\left(\varphi(\overline{\mathrm{p2}},m_u)\right).\bar{\gamma}^{\mathrm{Lor3}}.\left(\varphi(\overline{\mathrm{p1}},m_u)\right)\right),\mathrm{diracS}\left(\left(\varphi(\overline{\mathrm{p2}},m_u)\right).\bar{\gamma}^{\mu}.\left(\bar{\gamma}\cdot\left(\overline{\mathrm{p2}}-\overline{\mathrm{k1}}\right)+m_u\right).\bar{\gamma}^{\nu}.\left(\varphi(\overline{\mathrm{p1}},\right.\right.\right.$$
$$\left.\left.\left.m_u)\right)\right),\mathrm{diracS}\left(\left(\varphi(\overline{\mathrm{p2}},m_u)\right).\bar{\gamma}^{\nu}.\left(\bar{\gamma}\cdot\left(\overline{\mathrm{k2}}+\overline{\mathrm{p2}}\right)+m_u\right).\bar{\gamma}^{\mu}.\left(\varphi(\overline{\mathrm{p1}},m_u)\right)\right)\right\}$$

This way we obtain a sorted list of all unique Dirac structures in **`amp`**.

```
ClearAll[amp, ampIso, diracS]
```

5.24 FCGetDiracGammaScheme

`FCGetDiracGammaScheme[]` shows the currently used scheme for handling Dirac matrices in D dimensions.

5.24.1 See also

Overview, FCSetDiracGammaScheme, DiracTrace.

5.24.2 Examples

```
FCSetDiracGammaScheme["BMHV"]

FCGetDiracGammaScheme[]

% // FullForm
```

BMHV

BMHV

BMHV

```
FCSetDiracGammaScheme["NDR"]

FCGetDiracGammaScheme[]

% // FullForm
```

NDR

NDR

NDR

5.25 FCSetDiracGammaScheme

FCSetDiracGammaScheme[scheme] allows you to specify how Dirac matrices will be handled in **D** dimensions. This is mainly relevant to the treatment of the 5th Dirac matrix γ^5, which is not well-defined in dimensional regularization.

Following schemes are supported:

"NDR" - This is the default value. In the naive dimensional regularization (also known as conventional dimensional regularization or CDR) γ^5 is assumed to anticommute with all Dirac matrices in D dimensions. Hence, every Dirac trace can be rewritten in such a way, that it contains either just one or not a single γ^5 matrix. The latter traces are obviously unambiguous. The traces with one γ^5 are not well-defined in this scheme. It usually depends on the physics of the process, whether and how they can contribute to the final result. Therefore, FeynCalc will keep such traces unevaluated, leaving it to the user to decide how to treat them. Notice that traces with an odd number of the usual Dirac matrices and one γ^5, that vanish in 4 dimensions, will be also put to zero in this scheme.

"NDR-Discard" - This is a special version of the NDR scheme. The Dirac algebra is evaluated in the same way as with "NDR", but the remaining traces with one γ^5 are put to zero. This assumes that such traces do not contribute to the final result, which is obviously true only for specific calculations.

"BMHV" - The Breitenlohner-Maison extension of the t'Hooft-Veltman scheme. This scheme introduces Dirac and Lorentz tensors living in 4, D or $D-4$ dimensions, while γ^5 is a purely 4-dimensional object. BMHV is algebraically consistent but often suffers from nonconservation of currents in the final results. The conservation must be then enforced by introducing finite counter-terms. The counter-terms are to be supplied by the user, since FeynCalc does not do this automatically.

"Larin" - Special prescription developed by S. Larin, also known as the Larin-Gorishny-Atkyampo-DelBurgo scheme. Essentially, it is a shortcut (mostly used in QCD) for obtaining the same results as in BMHV but without the necessity to deal with tensors from different dimensions. In this scheme γ^5 is treated as nonanticommuting, while Dirac traces are still cyclic. If a chain of Dirac matrices contains a single γ^5, it is essentially left untouched. When computing the trace of such a chain, the cyclicity is used to put γ^5 to the very end of the chain. Then, the trace is evaluated using the Moch-Vermaseren-Vogt formula, Eq.(10) from arXiv:1506.04517. If a chain contains more than one γ^5, all but one γ^5 will be eliminated using the replacement $\gamma_\mu\gamma^5 \to i/6\varepsilon_{\mu\nu\rho\sigma}\gamma^\nu\gamma^\rho\gamma^\sigma$. This way every trace with multiple occurrences of γ^5 can be converted to a linear combination of traces with a single γ^5. Such traces are then treated as described above. Notice that Levi-Civita tensors generated during the calculation of traces are D-dimensional. For example, a product of two such tensors with all their indices contracted yields a polynomial in D's. This scheme is often used for performance reasons and is assumed to give the same results as the BMHV scheme. However, this is not a rigorous statement and so when in doubt it might be better to use BMHV instead.

5.25.1 See also

Overview, FCGetDiracGammaScheme, DiracTrace.

5.25.2 Examples

In NDR chiral traces remain unevaluated. You decide how to treat them.

```
FCSetDiracGammaScheme["NDR"]

DiracTrace[GAD[\[Mu], \[Nu], \[Rho], \[Sigma], \[Tau], \[Kappa], 5]]

DiracSimplify[%]
```

NDR

$$\text{tr}\left(\gamma^{\mu}.\gamma^{\nu}.\gamma^{\rho}.\gamma^{\sigma}.\gamma^{\tau}.\gamma^{\kappa}.\bar{\gamma}^{5}\right)$$

$$\text{tr}\left(\gamma^{\mu}.\gamma^{\nu}.\gamma^{\rho}.\gamma^{\sigma}.\gamma^{\tau}.\gamma^{\kappa}.\bar{\gamma}^{5}\right)$$

If you know that such traces do not contribute, use NDR-Discard scheme to put them to zero

```
FCSetDiracGammaScheme["NDR-Discard"]

DiracSimplify[DiracTrace[GAD[\[Mu], \[Nu], \[Rho], \[Sigma], \[Tau],
\[Kappa], 5]]]
```

NDR-Discard

$$0$$

In BMHV chiral traces are algebraically well-defined

```
FCSetDiracGammaScheme["BMHV"]

res1 = DiracSimplify[DiracTrace[GAD[\[Mu], \[Nu], \[Rho], \[Sigma], \[Tau],
\[Kappa], 5]]]
```

BMHV

$$\begin{aligned}&-4ig^{\kappa\mu}\bar{\epsilon}^{\nu\rho\sigma\tau}+4ig^{\kappa\nu}\bar{\epsilon}^{\mu\rho\sigma\tau}-4ig^{\kappa\rho}\bar{\epsilon}^{\mu\nu\sigma\tau}+4ig^{\kappa\sigma}\bar{\epsilon}^{\mu\nu\rho\tau}-4ig^{\kappa\tau}\bar{\epsilon}^{\mu\nu\rho\sigma}\\&+4ig^{\mu\nu}\bar{\epsilon}^{\kappa\rho\sigma\tau}-4ig^{\mu\rho}\bar{\epsilon}^{\kappa\nu\sigma\tau}+4ig^{\mu\sigma}\bar{\epsilon}^{\kappa\nu\rho\tau}-4ig^{\mu\tau}\bar{\epsilon}^{\kappa\nu\rho\sigma}+4ig^{\nu\rho}\bar{\epsilon}^{\kappa\mu\sigma\tau}\\&-4ig^{\nu\sigma}\bar{\epsilon}^{\kappa\mu\rho\tau}+4ig^{\nu\tau}\bar{\epsilon}^{\kappa\mu\rho\sigma}+4ig^{\rho\sigma}\bar{\epsilon}^{\kappa\mu\nu\tau}-4ig^{\rho\tau}\bar{\epsilon}^{\kappa\mu\nu\sigma}+4ig^{\sigma\tau}\bar{\epsilon}^{\kappa\mu\nu\rho}\end{aligned}$$

Larin's scheme reproduces the results of the BMHV scheme, but this may not be immediately obvious

```
FCSetDiracGammaScheme["Larin"]

res2 = DiracSimplify[DiracTrace[GAD[\[Mu], \[Nu], \[Rho], \[Sigma], \[Tau],
\[Kappa], 5]]]
```

Larin

$$4ig^{\mu\nu}\epsilon^{\kappa\rho\sigma\tau} - 4ig^{\mu\rho}\epsilon^{\kappa\nu\sigma\tau} + 4ig^{\mu\sigma}\epsilon^{\kappa\nu\rho\tau} - 4ig^{\mu\tau}\epsilon^{\kappa\nu\rho\sigma} + 4ig^{\nu\rho}\epsilon^{\kappa\mu\sigma\tau} \\ - 4ig^{\nu\sigma}\epsilon^{\kappa\mu\rho\tau} + 4ig^{\nu\tau}\epsilon^{\kappa\mu\rho\sigma} + 4ig^{\rho\sigma}\epsilon^{\kappa\mu\nu\tau} - 4ig^{\rho\tau}\epsilon^{\kappa\mu\nu\sigma} + 4ig^{\sigma\tau}\epsilon^{\kappa\mu\nu\rho}$$

Owing to Schouten identities, proving the equivalence of chiral traces is not so simple, especially for many terms. **FCSchoutenBruteForce** can be helpful here

```
diff = ChangeDimension[res1 - res2, D]

Contract[FV[p1, \[Mu]] FV[p2, \[Nu]] FV[p3, \[Rho]] FV[p4, \[Sigma]] FV[p5,
\[Tau]] FV[p6, \[Kappa]] diff]

FCSchoutenBruteForce[%, {}, {}]
```

$$-4ig^{\kappa\mu}\epsilon^{\nu\rho\sigma\tau} + 4ig^{\kappa\nu}\epsilon^{\mu\rho\sigma\tau} - 4ig^{\kappa\rho}\epsilon^{\mu\nu\sigma\tau} + 4ig^{\kappa\sigma}\epsilon^{\mu\nu\rho\tau} - 4ig^{\kappa\tau}\epsilon^{\mu\nu\rho\sigma}$$

$$-4i\left(\overline{\text{p1}}\cdot\overline{\text{p6}}\right)\bar{\epsilon}^{\overline{\text{p2}}\,\overline{\text{p3}}\,\overline{\text{p4}}\,\overline{\text{p5}}} + 4i\left(\overline{\text{p2}}\cdot\overline{\text{p6}}\right)\bar{\epsilon}^{\overline{\text{p1}}\,\overline{\text{p3}}\,\overline{\text{p4}}\,\overline{\text{p5}}} - 4i\left(\overline{\text{p3}}\cdot\overline{\text{p6}}\right)\bar{\epsilon}^{\overline{\text{p1}}\,\overline{\text{p2}}\,\overline{\text{p4}}\,\overline{\text{p5}}} \\ + 4i\left(\overline{\text{p4}}\cdot\overline{\text{p6}}\right)\bar{\epsilon}^{\overline{\text{p1}}\,\overline{\text{p2}}\,\overline{\text{p3}}\,\overline{\text{p5}}} - 4i\left(\overline{\text{p5}}\cdot\overline{\text{p6}}\right)\bar{\epsilon}^{\overline{\text{p1}}\,\overline{\text{p2}}\,\overline{\text{p3}}\,\overline{\text{p4}}}$$

FCSchoutenBruteForce: The following rule was applied: $\bar{\epsilon}^{\overline{\text{p2}}\,\overline{\text{p3}}\,\overline{\text{p4}}\,\overline{\text{p5}}}\left(\overline{\text{p1}}\cdot\overline{\text{p6}}\right)$:
$\to \bar{\epsilon}^{\overline{\text{p1}}\,\overline{\text{p3}}\,\overline{\text{p4}}\,\overline{\text{p5}}}\left(\overline{\text{p2}}\cdot\overline{\text{p6}}\right) - \bar{\epsilon}^{\overline{\text{p1}}\,\overline{\text{p2}}\,\overline{\text{p4}}\,\overline{\text{p5}}}\left(\overline{\text{p3}}\cdot\overline{\text{p6}}\right) + \bar{\epsilon}^{\overline{\text{p1}}\,\overline{\text{p2}}\,\overline{\text{p3}}\,\overline{\text{p5}}}\left(\overline{\text{p4}}\cdot\overline{\text{p6}}\right) - \bar{\epsilon}^{\overline{\text{p1}}\,\overline{\text{p2}}\,\overline{\text{p3}}\,\overline{\text{p4}}}\left(\overline{\text{p5}}\cdot\overline{\text{p6}}\right)$

FCSchoutenBruteForce: The numbers of terms in the expression decreased by: 5

FCSchoutenBruteForce: Current length of the expression: 0

0

```
FCSetDiracGammaScheme["NDR"]
```

NDR

5.26 GordonSimplify

`GordonSimplify[exp]` rewrites spinor chains describing a vector or an axial-vector current using Gordon identities.

5.26.1 See also

Overview, DiracGamma, Spinor, SpinorChainTrick.

5.26.2 Examples

```
SpinorUBar[p1, m1] . GA[\[Mu]] . SpinorU[p2, m2]

GordonSimplify[%]
```

$$\bar{u}(\text{p1},\text{m1}).\bar{\gamma}^{\mu}.u(\text{p2},\text{m2})$$

$$\frac{\left(\overline{\text{p1}}+\overline{\text{p2}}\right)^{\mu}\left(\varphi(\overline{\text{p1}},\text{m1})\right).\left(\varphi(\overline{\text{p2}},\text{m2})\right)}{\text{m1}+\text{m2}}+\frac{i\left(\varphi(\overline{\text{p1}},\text{m1})\right).\sigma^{\mu\overline{\text{p1}}-\overline{\text{p2}}}.\left(\varphi(\overline{\text{p2}},\text{m2})\right)}{\text{m1}+\text{m2}}$$

```
SpinorUBar[p1, m1] . GA[\[Mu], 5] . SpinorV[p2, m2]

GordonSimplify[%]
```

$$\bar{u}(\text{p1},\text{m1}).\bar{\gamma}^{\mu}.\bar{\gamma}^{5}.v(\text{p2},\text{m2})$$

$$\frac{\left(\overline{\text{p1}}+\overline{\text{p2}}\right)^{\mu}\left(\varphi(\overline{\text{p1}},\text{m1})\right).\bar{\gamma}^{5}.\left(\varphi(-\overline{\text{p2}},\text{m2})\right)}{\text{m1}+\text{m2}}+\frac{i\left(\varphi(\overline{\text{p1}},\text{m1})\right).\sigma^{\mu\overline{\text{p1}}-\overline{\text{p2}}}.\bar{\gamma}^{5}.\left(\varphi(-\overline{\text{p2}},\text{m2})\right)}{\text{m1}+\text{m2}}$$

Relations involving projectors can be used to trade the right projector for a left one

```
SpinorVBar[p1, m1] . GA[\[Mu], 6] . SpinorV[p2, m2]

GordonSimplify[%]
```

$$\bar{v}(\text{p1},\text{m1}).\bar{\gamma}^{\mu}.\bar{\gamma}^{6}.v(\text{p2},\text{m2})$$

$$-\frac{i\left(\varphi(-\overline{\text{p1}},\text{m1})\right).\sigma^{\mu\overline{\text{p1}}-\overline{\text{p2}}}.\bar{\gamma}^{6}.\left(\varphi(-\overline{\text{p2}},\text{m2})\right)}{\text{m1}}-\frac{\text{m2}\left(\varphi(-\overline{\text{p1}},\text{m1})\right).\bar{\gamma}^{\mu}.\bar{\gamma}^{7}.\left(\varphi(-\overline{\text{p2}},\text{m2})\right)}{\text{m1}}$$
$$-\frac{\left(\overline{\text{p1}}+\overline{\text{p2}}\right)^{\mu}\left(\varphi(-\overline{\text{p1}},\text{m1})\right).\bar{\gamma}^{6}.\left(\varphi(-\overline{\text{p2}},\text{m2})\right)}{\text{m1}}$$

Use the **`Select`** option to achieve the opposite

```
ex = SpinorVBar[p1, m1] . GA[\[Mu], 7] . SpinorV[p2, m2]

GordonSimplify[ex]
```

$$\bar{v}(\text{p1},\text{m1}).\bar{\gamma}^{\mu}.\bar{\gamma}^{7}.v(\text{p2},\text{m2})$$

$$\left(\varphi(-\overline{\text{p1}},\text{m1})\right).\bar{\gamma}^{\mu}.\bar{\gamma}^{7}.\left(\varphi(-\overline{\text{p2}},\text{m2})\right)$$

```
GordonSimplify[ex, Select -> {{Spinor[__], DiracGamma[__], GA[7],
Spinor[__]}}]
```

$$-\frac{i\left(\varphi(-\overline{\text{p1}},\text{m1})\right).\sigma^{\mu\overline{\text{p1}}-\overline{\text{p2}}}.\bar{\gamma}^{7}.\left(\varphi(-\overline{\text{p2}},\text{m2})\right)}{\text{m1}}-\frac{\text{m2}\left(\varphi(-\overline{\text{p1}},\text{m1})\right).\bar{\gamma}^{\mu}.\bar{\gamma}^{6}.\left(\varphi(-\overline{\text{p2}},\text{m2})\right)}{\text{m1}}$$
$$-\frac{\left(\overline{\text{p1}}+\overline{\text{p2}}\right)^{\mu}\left(\varphi(-\overline{\text{p1}},\text{m1})\right).\bar{\gamma}^{7}.\left(\varphi(-\overline{\text{p2}},\text{m2})\right)}{\text{m1}}$$

We can choose between having expressions proportional to $1/m_1$ (mass of the first spinor) or $1/m_2$ (mass of the second spinor)

```
GordonSimplify[SpinorVBar[p1, m1] . GA[\[Mu], 6] . SpinorV[p2, m2], Inverse
-> First]
```

$$-\frac{i\left(\varphi(-\overline{\text{p1}},\text{m1})\right).\sigma^{\mu\overline{\text{p1}}-\overline{\text{p2}}}.\bar{\gamma}^{6}.\left(\varphi(-\overline{\text{p2}},\text{m2})\right)}{\text{m1}}-\frac{\text{m2}\left(\varphi(-\overline{\text{p1}},\text{m1})\right).\bar{\gamma}^{\mu}.\bar{\gamma}^{7}.\left(\varphi(-\overline{\text{p2}},\text{m2})\right)}{\text{m1}}$$
$$-\frac{\left(\overline{\text{p1}}+\overline{\text{p2}}\right)^{\mu}\left(\varphi(-\overline{\text{p1}},\text{m1})\right).\bar{\gamma}^{6}.\left(\varphi(-\overline{\text{p2}},\text{m2})\right)}{\text{m1}}$$

```
GordonSimplify[SpinorVBar[p1, m1] . GA[\[Mu], 6] . SpinorV[p2, m2], Inverse
-> Last]
```

$$-\frac{i\left(\varphi(-\overline{\text{p1}},\text{m1})\right).\sigma^{\mu\,\overline{\text{p1}}-\overline{\text{p2}}}.\bar{\gamma}^{7}.\left(\varphi(-\overline{\text{p2}},\text{m2})\right)}{\text{m2}}-\frac{\text{m1}\left(\varphi(-\overline{\text{p1}},\text{m1})\right).\bar{\gamma}^{\mu}.\bar{\gamma}^{7}.\left(\varphi(-\overline{\text{p2}},\text{m2})\right)}{\text{m2}}$$
$$-\frac{\left(\overline{\text{p1}}+\overline{\text{p2}}\right)^{\mu}\left(\varphi(-\overline{\text{p1}},\text{m1})\right).\bar{\gamma}^{7}.\left(\varphi(-\overline{\text{p2}},\text{m2})\right)}{\text{m2}}$$

In D-dimensions chiral Gordon identities are scheme dependent!

```
ex = SpinorVBarD[p1, m1] . GAD[\[Mu], 5] . SpinorVD[p2, m2]
```

$$\bar{v}(\text{p1},\text{m1}).\gamma^{\mu}.\bar{\gamma}^{5}.v(\text{p2},\text{m2})$$

```
FCGetDiracGammaScheme[]

GordonSimplify[ex]
```

NDR

$$-\frac{(\text{p1}+\text{p2})^{\mu}(\varphi(-\text{p1},\text{m1})).\bar{\gamma}^{5}.(\varphi(-\text{p2},\text{m2}))}{\text{m1}-\text{m2}}-\frac{i(\varphi(-\text{p1},\text{m1})).\sigma^{\mu\ \text{p1}-\text{p2}}.\bar{\gamma}^{5}.(\varphi(-\text{p2},\text{m2}))}{\text{m1}-\text{m2}}$$

```
FCSetDiracGammaScheme["BMHV"]

GordonSimplify[ex]
```

BMHV

$$-\frac{i(\varphi(-\text{p1},\text{m1})).\sigma^{\mu\ \text{p1}-\text{p2}}.\bar{\gamma}^{5}.(\varphi(-\text{p2},\text{m2}))}{\text{m1}-\text{m2}}-\frac{(\text{p1}+\text{p2})^{\mu}(\varphi(-\text{p1},\text{m1})).\bar{\gamma}^{5}.(\varphi(-\text{p2},\text{m2}))}{\text{m1}-\text{m2}}$$
$$+\frac{2(\varphi(-\text{p1},\text{m1})).\gamma^{\mu}.\left(\hat{\gamma}\cdot\hat{\text{p2}}\right).\bar{\gamma}^{5}.(\varphi(-\text{p2},\text{m2}))}{\text{m1}-\text{m2}}$$

```
FCSetDiracGammaScheme["NDR"]
```

NDR

5.27 SirlinSimplify

`SirlinSimplify[exp]` simplifies spinor chains that contain Dirac matrices using relations derived by A. Sirlin in Nuclear Physics B192 (1981) 93-99. Contrary to the original paper, the sign of the Levi-Civita tensor is chosen as $\varepsilon^{0123} = 1$ which is the standard choice in FeynCalc.

5.27.1 See also

Overview, DiracGamma, Spinor, SpinorChainTrick.

5.27.2 Examples

```
SpinorUBar[p3, m3] . GA[\[Mu], \[Rho], \[Nu], 7] . SpinorU[p1, m1]
SpinorUBar[p4, m4] . GA[\[Mu], \[Tau], \[Nu], 7] . SpinorU[p2, m2]

SirlinSimplify[%]
```

$$\bar{u}(\text{p3},\text{m3}).\bar{\gamma}^{\mu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\nu}.\bar{\gamma}^{7}.u(\text{p1},\text{m1})\bar{u}(\text{p4},\text{m4}).\bar{\gamma}^{\mu}.\bar{\gamma}^{\tau}.\bar{\gamma}^{\nu}.\bar{\gamma}^{7}.u(\text{p2},\text{m2})$$

$$4\bar{g}^{\rho\tau}\left(\varphi(\overline{\text{p3}},\text{m3})\right).\bar{\gamma}^{\text{liS29}}.\bar{\gamma}^{7}.\left(\varphi(\overline{\text{p1}},\text{m1})\right)\left(\varphi(\overline{\text{p4}},\text{m4})\right).\bar{\gamma}^{\text{liS29}}.\bar{\gamma}^{7}.\left(\varphi(\overline{\text{p2}},\text{m2})\right)$$

5.28 SpinorChainEvaluate

`SpinorChainEvaluate[exp]` explicitly evaluates suitable spinor chains, i.e. it replaces a **`DOT[Spinor[...], ..., Spinor[...]]`** with a scalar quantity without a DOT.

5.28.1 See also

Overview

5.28.2 Examples

```
ex = SpinorUBar[p, m] . SpinorU[p, m]

SpinorChainEvaluate[ex]
```

$$\bar{u}(p,m).u(p,m)$$

$$2m$$

```
SpinorChainEvaluate[ex, DiracSpinorNormalization -> "Nonrelativistic"]
```

$$\frac{m}{p^0}$$

```
SpinorChainEvaluate[ex, DiracSpinorNormalization -> "Rest"]
```

$$1$$

```
ex = SpinorUBarD[p, m] . GA[5] . SpinorUD[p, m]

SpinorChainEvaluate[ex]
```

$$\bar{u}(p,m).\bar{\gamma}^5.u(p,m)$$

$$0$$

```
FCSetDiracGammaScheme["BMHV"]

SpinorChainEvaluate[ex]
```

BMHV

$$(\varphi(p,m)).\bar{\gamma}^5.(\varphi(p,m))$$

5.29 SpinorChainChiralSplit

`SpinorChainChiralSplit[exp]` introduces chiral projectors in spinor chains that contain no γ^5.

5.29.1 See also

Overview, DiracSubstitute67, DiracGamma, ToDiracGamma67.

5.29.2 Examples

```
SpinorUBar[p1, m1] . GSD[p] . SpinorV[p2, m2]

SpinorChainChiralSplit[%]
```

$$\bar{u}(\text{p1},\text{m1}).(\gamma\cdot p).v(\text{p2},\text{m2})$$

$$\left(\varphi(\overline{\text{p1}},\text{m1})\right).(\gamma\cdot p).\bar{\gamma}^6.\left(\varphi(-\overline{\text{p2}},\text{m2})\right)+\left(\varphi(\overline{\text{p1}},\text{m1})\right).(\gamma\cdot p).\bar{\gamma}^7.\left(\varphi(-\overline{\text{p2}},\text{m2})\right)$$

5.30 SpinorChainTranspose

SpinorChainTranspose[exp] transposes particular spinor chains in exp, which effectively switches the u and v spinors and reverses the order of the Dirac matrices using charge conjugation operator. This operation is often required in calculations that involve Majorana particles. By default, the function will tranpose all chains of the form $\bar{v}.x.u$ and $\bar{v}.x.v$. A different or more fine grained choice can be obtained via the option **Select**.

5.30.1 See also

Overview, FCChargeConjugateTransposed, DiracGamma, Spinor.

5.30.2 Examples

```
SpinorVBarD[p1, m1] . GAD[\[Mu]] . (GSD[p] + m) . GAD[\[Mu]] . SpinorUD[p2,
m2]

SpinorChainTranspose[%]
```

$$\bar{v}(\text{p1},\text{m1}).\gamma^{\mu}.(m+\gamma\cdot p).\gamma^{\mu}.u(\text{p2},\text{m2})$$

$$-(\varphi(-\text{p2},\text{m2})).\gamma^{\mu}.(m-\gamma\cdot p).\gamma^{\mu}.(\varphi(\text{p1},\text{m1}))$$

```
SpinorUBarD[p1, m1] . GAD[\[Mu]] . (GSD[p] + m) . GAD[\[Mu]] . SpinorVD[p2,
m2]

SpinorChainTranspose[%]
```

$$\bar{u}(\text{p1},\text{m1}).\gamma^{\mu}.(m+\gamma\cdot p).\gamma^{\mu}.v(\text{p2},\text{m2})$$

$$(\varphi(\text{p1},\text{m1})).\gamma^{\mu}.(m+\gamma\cdot p).\gamma^{\mu}.(\varphi(-\text{p2},\text{m2}))$$

```
SpinorUBarD[p1, m1] . GAD[\[Mu]] . (GSD[p] + m) . GAD[\[Mu]] . SpinorVD[p2,
m2]

SpinorChainTranspose[%, Select -> {{SpinorUBarD[_, _], SpinorVD[_, _]}}]
```

$$\bar{u}(\mathrm{p1},\mathrm{m1}).\gamma^{\mu}.(m+\gamma\cdot p).\gamma^{\mu}.v(\mathrm{p2},\mathrm{m2})$$

$$-(\varphi(\mathrm{p2},\mathrm{m2})).\gamma^{\mu}.(m-\gamma\cdot p).\gamma^{\mu}.(\varphi(-\mathrm{p1},\mathrm{m1}))$$

5.31 SpinorChainTrick

`SpinorChainTrick[exp]` applies several simplifications to products of spinor chains.

5.31.1 See also

Overview, FCCanonicalizeDummyIndices, DiracGamma, Spinor.

5.31.2 Examples

```
a SpinorUBar[p1, m1] . GA[\[Mu]] . SpinorU[p2, m2] SpinorVBar[p1, m1] .
GA[\[Mu]] . SpinorV[p4, m4] +
   b SpinorUBar[p1, m1] . GA[\[Nu]] . SpinorU[p2, m2] SpinorVBar[p1, m1] .
   GA[\[Nu]] . SpinorV[p4, m4]

SpinorChainTrick[%]
```

$$a\bar{u}(\mathrm{p1},\mathrm{m1}).\bar{\gamma}^{\mu}.u(\mathrm{p2},\mathrm{m2})\bar{v}(\mathrm{p1},\mathrm{m1}).\bar{\gamma}^{\mu}.v(\mathrm{p4},\mathrm{m4})+b\bar{u}(\mathrm{p1},\mathrm{m1}).\bar{\gamma}^{\nu}.u(\mathrm{p2},\mathrm{m2})\bar{v}(\mathrm{p1},\mathrm{m1}).\bar{\gamma}^{\nu}.v(\mathrm{p4},\mathrm{m4})$$

$$(a+b)\left(\varphi(\overline{\mathrm{p1}},\mathrm{m1})\right).\bar{\gamma}^{\mathrm{FCGV(LI191)}}.\left(\varphi(\overline{\mathrm{p2}},\mathrm{m2})\right)\left(\varphi(-\overline{\mathrm{p1}},\mathrm{m1})\right).\bar{\gamma}^{\mathrm{FCGV(LI191)}}.\left(\varphi(-\overline{\mathrm{p4}},\mathrm{m4})\right)$$

```
SpinorUBar[p1, m1] . GAE[\[Mu]] . SpinorU[p2, m2] SpinorVBar[p1, m1] .
GA[\[Mu]] . SpinorV[p4, m4]

SpinorChainTrick[%]
```

$$\bar{u}(\mathrm{p1},\mathrm{m1}).\hat{\gamma}^{\mu}.u(\mathrm{p2},\mathrm{m2})\bar{v}(\mathrm{p1},\mathrm{m1}).\bar{\gamma}^{\mu}.v(\mathrm{p4},\mathrm{m4})$$

$$0$$

5.32 ToDiracGamma67

`ToDiracGamma67[exp]` substitutes $\frac{1}{2}\left(1+\gamma^5\right)$ and $\frac{1}{2}\left(1-\gamma^5\right)$ by the chirality projectors γ^6 and γ^7.

5.32.1 See also

Overview, DiracSubstitute5, DiracGamma, ToDiracGamma67.

5.32.2 Examples

```
GA[\[Mu]] . (1/2 + GA[5]/2) . GA[\[Nu]]

ToDiracGamma67[%]
```

$$\bar{\gamma}^{\mu}.\left(\frac{\bar{\gamma}^5}{2}+\frac{1}{2}\right).\bar{\gamma}^{\nu}$$

$$\bar{\gamma}^{\mu}.\bar{\gamma}^6.\bar{\gamma}^{\nu}$$

When the option **`All`** is set to **`True`**, also standalone γ^5 will be replaced

```
GA[\[Mu], 5, \[Nu]]

ToDiracGamma67[%, All -> True]
```

$$\bar{\gamma}^{\mu}.\bar{\gamma}^5.\bar{\gamma}^{\nu}$$

$$\bar{\gamma}^{\mu}.\left(\bar{\gamma}^6-\bar{\gamma}^7\right).\bar{\gamma}^{\nu}$$

5.33 ToDiracSigma

`ToDiracSigma[exp, x, y]` substitutes the neighboring Dirac matrices x and y by **`DiracSigma`** and the metric tensor.

5.33.1 See also

Overview, DiracGamma, DiracSigma, DiracSigmaExplicit.

5.33.2 Examples

```
GA[\[Mu], \[Nu]]

ToDiracSigma[%, GA[\[Mu]], GA[\[Nu]]]
```

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}$$

$$\bar{g}^{\mu\nu} - i\sigma^{\mu\nu}$$

```
GA[\[Mu], \[Nu], \[Alpha], \[Beta], \[Rho], \[Sigma]]

ToDiracSigma[%, GA[\[Alpha]], GA[\[Beta]]]
```

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\alpha}.\bar{\gamma}^{\beta}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\sigma}$$

$$\bar{g}^{\alpha\beta}\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\sigma} - i\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\sigma^{\alpha\beta}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\sigma}$$

5.34 ToLarin

ToLarin[exp] substitutes $\gamma^{\mu}\gamma^{5}$ with $-\frac{I}{6}\varepsilon^{\mu\nu\lambda\sigma}\gamma^{\nu}\gamma^{\lambda}\gamma^{\sigma}$.

5.34.1 See also

Overview, Eps, DiracGamma.

5.34.2 Examples

```
GAD[\[Mu], \[Nu]] . GA[5]

ToLarin[%]
```

$$\gamma^{\mu}.\gamma^{\nu}.\bar{\gamma}^{5}$$

$$-\frac{1}{6}i\gamma^{\mu}.\gamma^{\text{du19}}.\gamma^{\text{du20}}.\gamma^{\text{du21}}\epsilon^{\nu\ \text{du19}\ \text{du20}\ \text{du21}}$$

6 Pauli algebra

6.1 FCGetPauliSigmaScheme

FCGetPauliSigmaScheme[] shows the currently used scheme for handling Pauli matrices in $D-1$ dimensions. For more details see the documentation of **FCSetPauliSigmaScheme**.

6.1.1 See also

Overview, PauliSigma, FCSetPauliSigmaScheme.

6.1.2 Examples

```
FCGetPauliSigmaScheme[]
```

None

6.2 FCSetPauliSigmaScheme

FCSetPauliSigmaScheme[scheme] allows you to specify how Pauli matrices will be handled in $D-1$ dimensions.

This is mainly related to the commutator of two Pauli matrices, which involves a Levi-Civita tensor. The latter is not a well-defined quantity in $D-1$ dimensions. Following schemes are supported:

- **"None"** - This is the default value. The anticommutator relation is not applied to $D-1$ dimensional Pauli matrices.
- **"Naive"** - Naively apply the commutator relation in $D-1$-dimensions, i.e. $\{\sigma^i, \sigma^j\} = 2i\varepsilon^{ijk}\sigma^k$. The Levi-Civita tensor lives in $D-1$-dimensions, so that a contraction of two such tensors which have all indices in common yields $(D-3)(D-2)(D-1)$.

6.2.1 See also

Overview, PauliSigma, FCGetPauliSigmaScheme.

6.2.2 Examples

```
FCGetPauliSigmaScheme[]
```

None

```
CSID[i, j, k]

PauliSimplify[%, PauliReduce -> True]
```

$$\sigma^i.\sigma^j.\sigma^k$$

$$\sigma^i.\sigma^j.\sigma^k$$

```
FCSetPauliSigmaScheme["Naive"];
```

```
FCGetPauliSigmaScheme[]
```

Naive

```
ex = PauliSimplify[CSID[i, j, k], PauliReduce -> True]
```

$$i\epsilon^{ijk} + D\sigma^i\delta^{jk} - D\sigma^j\delta^{ik} - 3\sigma^i\delta^{jk} + 3\sigma^j\delta^{ik} + \sigma^k\delta^{ij}$$

```
ex // FCE // StandardForm

(*I CLCD[i, j, k] + CSID[k] KDD[i, j] + 3 CSID[j] KDD[i, k] - D CSID[j]
KDD[i, k] - 3 CSID[i] KDD[j, k] + D CSID[i] KDD[j, k]*)
```

```
FCSetPauliSigmaScheme["None"];
```

6.3 FCPauliIsolate

`FCPauliIsolate[exp]` wraps chains of Pauli matrices into heads specified by the user.

6.3.1 See also

Overview

6.3.2 Examples

```
FCPauliIsolate[y SI[i] + x PauliXi[-I] . SIS[p1] . PauliEta[I] .
PauliEta[-I] . SIS[p2] . PauliXi[I], Head -> pChain]
```

$$y\ \text{pChain}\left(\bar{\sigma}^{i}\right) + x\ \text{pChain}\left(\xi^{\dagger}.\left(\bar{\sigma}\cdot\overline{\text{p1}}\right).\eta\right)\ \text{pChain}\left(\eta^{\dagger}.\left(\bar{\sigma}\cdot\overline{\text{p2}}\right).\xi\right)$$

6.4 PauliChainJoin

`PauliChainJoin[exp]` joins chains of Pauli matrices with explicit Pauli indices wrapped with a head **`PauliChain`**.

6.4.1 See also

Overview, PauliChain, PCHN, PauliIndex, PauliIndexDelta, DIDelta, PauliChainCombine, PauliChainExpand, PauliChainFactor.

6.4.2 Examples

```
PCHN[PauliXi[-I], i] PCHN[CSID[a] . CSID[b], i, j] PCHN[j, PauliEta[I]]

PauliChainJoin[%]
```

$$(\eta)_{j}\left(\xi^{\dagger}\right)_{i}\left(\sigma^{a}.\sigma^{b}\right)_{ij}$$

$$\xi^{\dagger}.\sigma^{a}.\sigma^{b}.\eta$$

6.5 PauliChainCombine

`PauliChainCombine[exp]` is (nearly) the inverse operation to **`PauliChainExpand`**.

6.5.1 See also

Overview, PauliChain, PCHN, PauliIndex, PauliIndexDelta, DIDelta, PauliChainJoin, PauliChainExpand, PauliChainFactor.

6.5.2 Examples

```
(PCHN[CSISD[q], Dir3, Dir4] FAD[{k, me}])/(2 CSPD[q, q]) + 1/(2 CSPD[q,
q])*
    FAD[k, {k - q, me}] (-2 DCHN[CSISD[q], Dir3, Dir4] CSPD[q, q] + 2
DCHN[1, Dir3, Dir4]*
       me CSPD[q, q] + DCHN[CSISD[q], Dir3, Dir4] (-me^2 + CSPD[q, q]))

PauliChainCombine[%]
```

$$\frac{\left(q^2-\text{me}^2\right)\left((1)_{\text{Dir3 Dir4}}\sigma\cdot q\right)+2\,\text{me}q^2(1)_{\text{Dir3 Dir4}}-2q^2\left((1)_{\text{Dir3 Dir4}}\sigma\cdot q\right)}{2q^2k^2.\left((k-q)^2-\text{me}^2\right)}+\frac{(\sigma\cdot q)_{\text{Dir3 Dir4}}}{2q^2\left(k^2-\text{me}^2\right)}$$

$$\frac{(1)_{\text{Dir3 Dir4}}\left(q^2-\text{me}^2\right)\sigma\cdot q+2\,\text{me}q^2(1)_{\text{Dir3 Dir4}}-2q^2(1)_{\text{Dir3 Dir4}}\sigma\cdot q}{2q^2k^2.\left((k-q)^2-\text{me}^2\right)}+\frac{(\sigma\cdot q)_{\text{Dir3 Dir4}}}{2q^2\left(k^2-\text{me}^2\right)}$$

6.6 PauliChainExpand

PauliChainExpand[exp] expands all Pauli chains with explicit indices using linearity, e.g. **PCHN[CSIS[p1]+CSIS[p2]+m, i, j]** becomes **PCHN[CSIS[p1], i, j]+PCHN[CSIS[p2], i, j]+m*PCHN[1, i, j]**.

6.6.1 See also

Overview, PauliChain, PCHN, PauliIndex, PauliIndexDelta, DIDelta, PauliChainJoin, PauliChainCombine, PauliChainFactor.

6.6.2 Examples

```
PCHN[(CSIS[p] + m) . CSI[a], i, j]

PauliChainExpand[%]
```

$$\left(\left(\overline{\sigma}\cdot\overline{p}+m\right).\overline{\sigma}^a\right)_{ij}$$

$$m\left(\overline{\sigma}^a\right)_{ij}+\left(\left(\overline{\sigma}\cdot\overline{p}\right).\overline{\sigma}^a\right)_{ij}$$

6.7 PauliChainFactor

PauliChainFactor[exp] factors out all expressions inside a **PauliChain** to which the chain doesn't apply. For example, all objects that are not Pauli matrices can be safely factored out from every Pauli chain.

6.7.1 See also

Overview, PauliChain, PCHN, PauliIndex, PauliIndexDelta, DIDelta, PauliChainJoin, PauliChainCombine, PauliChainExpand.

6.7.2 Examples

```
PCHN[CV[p, \[Nu]] CSI[a] . CSI[b] . CSI[a], i, j]

PauliChainFactor[%]
```

$$\left(\overline{p}^{\nu}\overline{\sigma}^{a}.\overline{\sigma}^{b}.\overline{\sigma}^{a}\right)_{ij}$$

$$\overline{p}^{\nu}\left(\overline{\sigma}^{a}.\overline{\sigma}^{b}.\overline{\sigma}^{a}\right)_{ij}$$

6.8 PauliOrder

PauliOrder[exp] orders the Pauli matrices in **expr** alphabetically.

PauliOrder[exp, orderlist] orders the Pauli matrices in expr according to **orderlist**.

6.8.1 See also

Overview

6.8.2 Examples

```
CSI[k, j, i]

PauliOrder[%]
```

$$\overline{\sigma}^{k}.\overline{\sigma}^{j}.\overline{\sigma}^{i}$$

$$2\overline{\sigma}^{i}\bar{\delta}^{jk} - 2\overline{\sigma}^{j}\bar{\delta}^{ik} + 2\overline{\sigma}^{k}\bar{\delta}^{ij} - \overline{\sigma}^{i}.\overline{\sigma}^{j}.\overline{\sigma}^{k}$$

```
CSID[i, j, k]
```

```
PauliOrder[%]
```

$$\sigma^i.\sigma^j.\sigma^k$$

$$\sigma^i.\sigma^j.\sigma^k$$

```
PauliOrder[%%, {j, i, k}]
```

$$2\sigma^k\delta^{ij} - \sigma^j.\sigma^i.\sigma^k$$

6.9 PauliSigmaCombine

`PauliSigmaCombine[exp]` is (nearly) the inverse operation to PauliSigmaExpand.

6.9.1 See also

Overview, PauliSigmaExpand.

6.9.2 Examples

```
SIS[p] + SIS[q]
```

```
PauliSigmaCombine[%]
```

$$\bar{\sigma}\cdot\overline{p} + \bar{\sigma}\cdot\overline{q}$$

$$\bar{\sigma}\cdot(\overline{p} + \overline{q})$$

```
PauliXi[-I] . (SIS[p1 + p2] + SIS[q]) . PauliEta[I]
```

```
PauliSigmaCombine[%]
```

$$\xi^\dagger.\left(\bar{\sigma}\cdot\left(\overline{\text{p1}} + \overline{\text{p2}}\right) + \bar{\sigma}\cdot\overline{q}\right).\eta$$

$$\xi^\dagger.\left(\bar{\sigma}\cdot\left(\overline{\text{p1}} + \overline{\text{p2}} + \overline{q}\right)\right).\eta$$

6.10 PauliSigmaExpand

PauliSigmaExpand[exp] expands all **PauliSigma[Momentum[a+b+..]]** in **exp** into **(PauliSigma[Momentum[a]] + PauliSigma[Momentum[b]] + ...)**.

6.10.1 See also

Overview, PauliSigmaCombine.

6.10.2 Examples

```
SIS[q] . SIS[p - q]

PauliSigmaExpand[%]
```

$$(\bar{\sigma}\cdot\overline{q})\,.\,(\bar{\sigma}\cdot(\overline{p}-\overline{q}))$$

$$(\bar{\sigma}\cdot\overline{q})\,.\,(\bar{\sigma}\cdot\overline{p}-\bar{\sigma}\cdot\overline{q})$$

```
SIS[a + b] . SIS[c + d]

PauliSigmaExpand[%, Momentum -> {a}]

PauliSigmaExpand[%%, Momentum -> All]
```

$$(\bar{\sigma}\cdot(\overline{a}+\overline{b}))\,.\,(\bar{\sigma}\cdot(\overline{c}+\overline{d}))$$

$$(\bar{\sigma}\cdot\overline{a}+\bar{\sigma}\cdot\overline{b})\,.\,(\bar{\sigma}\cdot(\overline{c}+\overline{d}))$$

$$(\bar{\sigma}\cdot\overline{a}+\bar{\sigma}\cdot\overline{b})\,.\,(\bar{\sigma}\cdot\overline{c}+\bar{\sigma}\cdot\overline{d})$$

6.11 PauliSimplify

PauliSimplify[exp] simplifies products of Pauli matrices and expands non-commutative products. Double indices and vectors are contracted. The order of the Pauli matrices is not changed.

6.11.1 See also

Overview, PauliSigma, PauliTrick.

6.11.2 Examples

```
CSIS[p1] . CSI[i] . CSIS[p2]

PauliSimplify[%]
```

$$(\overline{\sigma} \cdot \overline{\text{p1}}) .\overline{\sigma}^i. (\overline{\sigma} \cdot \overline{\text{p2}})$$

$$(\overline{\sigma} \cdot \overline{\text{p1}}) .\overline{\sigma}^i. (\overline{\sigma} \cdot \overline{\text{p2}})$$

```
CSIS[p] . CSI[i, j, k] . CSIS[p]

PauliSimplify[%]
```

$$(\overline{\sigma} \cdot \overline{p}) .\overline{\sigma}^i.\overline{\sigma}^j.\overline{\sigma}^k. (\overline{\sigma} \cdot \overline{p})$$

$$-\overline{p}^2\overline{\sigma}^i.\overline{\sigma}^j.\overline{\sigma}^k + 2\overline{p}^k\overline{\sigma}^i.\overline{\sigma}^j. (\overline{\sigma} \cdot \overline{p}) - 2\overline{p}^j\overline{\sigma}^i.\overline{\sigma}^k. (\overline{\sigma} \cdot \overline{p}) + 2\overline{p}^i\overline{\sigma}^j.\overline{\sigma}^k. (\overline{\sigma} \cdot \overline{p})$$

```
PauliSimplify[CSIS[p] . CSI[i, j, k] . CSIS[p], PauliReduce -> False]
```

$$-\overline{p}^2\overline{\sigma}^i.\overline{\sigma}^j.\overline{\sigma}^k + 2\overline{p}^k\overline{\sigma}^i.\overline{\sigma}^j. (\overline{\sigma} \cdot \overline{p}) - 2\overline{p}^j\overline{\sigma}^i.\overline{\sigma}^k. (\overline{\sigma} \cdot \overline{p}) + 2\overline{p}^i\overline{\sigma}^j.\overline{\sigma}^k. (\overline{\sigma} \cdot \overline{p})$$

```
CSID[i, j, i]

PauliSimplify[%]
```

$$\sigma^i.\sigma^j.\sigma^i$$

$$3\sigma^j - D\sigma^j$$

```
CSID[i, j, k, l, m, i]

PauliSimplify[%]
```

$$\sigma^i.\sigma^j.\sigma^k.\sigma^l.\sigma^m.\sigma^i$$

$$D\sigma^j.\sigma^k.\sigma^l.\sigma^m - 3\sigma^j.\sigma^k.\sigma^l.\sigma^m + 2\sigma^j.\sigma^k.\sigma^m.\sigma^l - 2\sigma^j.\sigma^l.\sigma^m.\sigma^k + 2\sigma^k.\sigma^l.\sigma^m.\sigma^j$$

6.12 PauliTrace

PauliTrace[exp] is the head of Pauli traces. By default the trace is not evaluated. The evaluation occurs only when the option **PauliTraceEvaluate** is set to **True**. It is recommended to use **PauliSimplify**, which will automatically evaluate all Pauli traces in the input expression.

6.12.1 See also

Overview, PauliSimplify.

6.12.2 Examples

```
PauliTrace[CSI[i, j, k, l]]
```

$$\text{tr}\left(\overline{\sigma}^i.\overline{\sigma}^j.\overline{\sigma}^k.\overline{\sigma}^l\right)$$

```
PauliTrace[CSI[i, j, k, l], PauliTraceEvaluate -> True]
```

$$2\left(\bar{\delta}^{il}\bar{\delta}^{jk} - \bar{\delta}^{ik}\bar{\delta}^{jl} + \bar{\delta}^{ij}\bar{\delta}^{kl}\right)$$

```
PauliTrace[CSI[i, j, k, l]]

% // PauliSimplify
```

$$\text{tr}\left(\overline{\sigma}^i.\overline{\sigma}^j.\overline{\sigma}^k.\overline{\sigma}^l\right)$$

$$2\bar{\delta}^{il}\bar{\delta}^{jk} - 2\bar{\delta}^{ik}\bar{\delta}^{jl} + 2\bar{\delta}^{ij}\bar{\delta}^{kl}$$

6.13 PauliTrick

PauliTrick[exp] contracts σ matrices with each other and performs several simplifications (no expansion, use **PauliSimplify** for this).

6.13.1 See also

Overview, PauliSigma, PauliSimplify.

6.13.2 Examples

```
CSIS[p1] . CSI[i] . CSIS[p2]

PauliTrick[%] // Contract
```

$$\left(\overline{\sigma}\cdot\overline{\text{p1}}\right).\overline{\sigma}^{i}.\left(\overline{\sigma}\cdot\overline{\text{p2}}\right)$$

$$\left(\overline{\sigma}\cdot\overline{\text{p1}}\right).\overline{\sigma}^{i}.\left(\overline{\sigma}\cdot\overline{\text{p2}}\right)$$

```
CSID[i, j, i]

PauliTrick[%] // Contract
```

$$\sigma^{i}.\sigma^{j}.\sigma^{i}$$

$$-\left((D-3)\sigma^{j}\right)$$

```
CSIS[p] . CSI[j] . CSIS[p] . CSIS[i]

PauliTrick[%] // Contract // EpsEvaluate // FCCanonicalizeDummyIndices

PauliTrick[%%, PauliReduce -> False]
```

$$\left(\overline{\sigma}\cdot\overline{p}\right).\overline{\sigma}^{j}.\left(\overline{\sigma}\cdot\overline{p}\right).\left(\overline{\sigma}\cdot\overline{i}\right)$$

$$2\overline{p}^{j}\left(\overline{\sigma}\cdot\overline{p}\right).\left(\overline{\sigma}\cdot\overline{i}\right)-\overline{p}^{2}\overline{\sigma}^{j}.\left(\overline{\sigma}\cdot\overline{i}\right)$$

$$2\overline{p}^{j}\left(\overline{\sigma}\cdot\overline{p}\right).\left(\overline{\sigma}\cdot\overline{i}\right)-\overline{p}^{2}\overline{\sigma}^{j}.\left(\overline{\sigma}\cdot\overline{i}\right)$$

7 Algebra of noncommutative objects

7.1 AntiCommutator

AntiCommutator[x, y] = c defines the anti-commutator of the non commuting objects **x** and **y**.

7.1.1 See also

Overview, Commutator, CommutatorExplicit, DeclareNonCommutative, DotSimplify.

7.1.2 Examples

This declares **a** and **b** as noncommutative variables.

```
DeclareNonCommutative[a, b]

AntiCommutator[a, b]

CommutatorExplicit[%]
```

$$\{a, b\}$$

$$a.b + b.a$$

```
CommutatorExplicit[AntiCommutator[a + b, a - 2 b ]]
```

$$(a - 2b).(a + b) + (a + b).(a - 2b)$$

```
DotSimplify[AntiCommutator[a + b, a - 2 b ]]
```

$$-a.b - b.a + 2a.a - 4b.b$$

```
DeclareNonCommutative[c, d, ct, dt]
```

Defining **{c, d} = z** results in replacements of **c.d** by **z-d.c.**

```
AntiCommutator[c, d] = z

DotSimplify[ d . c . d ]
```

$$z$$

$$dz - d.d.c$$

```
AntiCommutator[dt, ct] = zt
```

$$\text{zt}$$

```
DotSimplify[dt . ct . dt]
```

$$\text{dt zt} - \text{ct.dt.dt}$$

```
UnDeclareNonCommutative[a, b, c, d, ct, dt]

UnDeclareAllAntiCommutators[]
```

7.2 Calc

Calc[exp] performs several simplifications that involve **Contract**, **DiracSimplify**, **SUNSimplify**, **DotSimplify**, **EpsEvaluate**, **ExpandScalarProduct**, **PowerSimplify**, **Expand2** and **Trick**.

7.2.1 See also

Overview, Trick, DiracSimplify, DiracTrick.

7.2.2 Examples

This calculates $\gamma^\mu \gamma_\mu$ in 4 dimensions and g^ν_ν in D dimensions.

```
Calc[GA[\[Mu], \[Mu]]]
```

$$4$$

```
Calc[ MTD[\[Nu], \[Nu]]]
```

$$D$$

This simplifies $f_{abc} f_{abe}$

```
Calc[SUNF[a, b, c] SUNF[a, b, e]]
```

$$C_A \delta^{ce}$$

```
FV[p + r, \[Mu]] MT[\[Mu], \[Nu]] FV[q - p, \[Nu]]

Calc[%]
```

$$\bar{g}^{\mu\nu} (\bar{q} - \bar{p})^{\nu} (\bar{p} + \bar{r})^{\mu}$$

$$\bar{p} \cdot \bar{q} - \bar{p} \cdot \bar{r} - \bar{p}^2 + \bar{q} \cdot \bar{r}$$

```
GluonVertex[{p, li1}, {q, li2}, {-p - q, li3}]

Calc[% FVD[p, li1] FVD[q, li2] FVD[-p - q, li3]]
```

$$V^{\text{li1 li2 li3}}(p, q, -p - q)$$

$$0$$

7.3 Trick

`Trick[exp]` performs several basic simplifications without expansion. **`Trick[exp]`** uses **`Contract`**, **`DotSimplify`** and **`SUNDeltaContract`**.

7.3.1 See also

Overview, Calc, Contract, DiracTrick, DotSimplify, DiracTrick.

7.3.2 Examples

This calculates $g^{\mu\nu}\gamma_\mu$ and g^ν_ν in D dimensions.

```
Trick[{GA[\[Mu]] MT[\[Mu], \[Nu]], MTD[\[Nu], \[Nu]]}]
```

$$\{\bar{\gamma}^\nu, D\}$$

```
FV[p + r, \[Mu]] MT[\[Mu], \[Nu]] FV[q - p, \[Nu]]

Trick[%]
```

$$\bar{g}^{\mu\nu}\,(\bar{q}-\bar{p})^\nu\,(\bar{p}+\bar{r})^\mu$$

$$\bar{p}\cdot\bar{q}-\bar{p}\cdot\bar{r}-\bar{p}^2+\bar{q}\cdot\bar{r}$$

```
Trick[c . b . a . GA[d] . GA[e]]
```

$$abc\bar{\gamma}^d.\bar{\gamma}^e$$

```
Trick[c . b . a . GA[d] . GA[e]] // FCE // StandardForm

(*a b c GA[d] . GA[e]*)
```

7.4 Commutator

`Commutator[x, y] = c` defines the commutator between the (non-commuting) objects **`x`** and **`y`**.

7.4.1 See also

Overview, AntiCommutator, CommutatorExplicit, DeclareNonCommutative, DotSimplify.

7.4.2 Examples

```
DeclareNonCommutative[a, b, c, d]
```

```
Commutator[a, b]

CommutatorExplicit[%]
```

$$[a, b]$$

$$a.b - b.a$$

```
DotSimplify[Commutator[a + b, c + d]]

UnDeclareNonCommutative[a, b, c, d]
```

$$a.c - c.a + a.d - d.a + b.c - c.b + b.d - d.b$$

Verify the Jacobi identity.

```
\[Chi] = Commutator; DeclareNonCommutative[x, y, z];
```

```
\[Chi][x, \[Chi][y, z]] + \[Chi][y, \[Chi][z, x]] + \[Chi][z, \[Chi][x, y]]

DotSimplify[%]
```

$$[x, [y, z]] + [y, [z, x]] + [z, [x, y]]$$

$$0$$

```
Clear[\[Chi]]

UnDeclareNonCommutative[x, y, z]
```

7.5 CommutatorExplicit

`CommutatorExplicit[exp]` substitutes any **`Commutator`** and **`AntiCommutator`** in **`exp`** by their definitions.

7.5.1 See also

Overview, Calc, DotSimplify.

7.5.2 Examples

```
DeclareNonCommutative[a, b, c, d]
```

```
Commutator[a, b]
```

```
CommutatorExplicit[%]
```

$$[a, b]$$

$$a.b - b.a$$

```
AntiCommutator[a - c, b - d]
```

```
CommutatorExplicit[%]
```

$$\{a - c,\ b - d\}$$

$$(a - c).(b - d) + (b - d).(a - c)$$

```
CommutatorExplicit[AntiCommutator[a - c, b - d]] // DotSimplify
```

$$a.b + b.a - a.d - d.a - b.c - c.b + c.d + d.c$$

```
UnDeclareNonCommutative[a, b, c, d]
```

7.6 CommutatorOrder

`CommutatorOrder[exp]` orders any **`Commutator`** and **`AntiCommutator`** lexicographically.

7.6.1 See also

Overview, Commutator, AntiCommutator.

7.6.2 Examples

```
Commutator[a, b] + Commutator[b, a]
```

```
CommutatorOrder[%]
```

$$[a, b] + [b, a]$$

$$0$$

7.7 DeclareNonCommutative

DeclareNonCommutative[a, b, ...] declares **a, b, ...** to be non-commutative, i.e., **DataType[a, b, ..., NonCommutative]** is set to **True**.

7.7.1 See also

Overview, DataType, UnDeclareNonCommutative.

7.7.2 Examples

As a side effect of **DeclareNonCommutative**, **x** is declared to be of data type **NonCommutative**.

```
DeclareNonCommutative[x]
```

```
DataType[x, NonCommutative]
```

True

```
DeclareNonCommutative[y, z]
```

```
DataType[a, x, y, z, NonCommutative]
```

{False, True, True, True}

```
UnDeclareNonCommutative[x, y, z]
```

```
DataType[a, x, y, z, NonCommutative]
```

{False, False, False, False}

7.8 DotExpand

DotExpand[exp] expands dot products in **exp**.

7.8.1 See also

Overview, DOT, DotSimplify, DeclareNonCommutative, UnDeclareNonCommutative

7.8.2 Examples

```
DOT[a x + b y + c z, d + e + f]

DotExpand[%]
```

$$(ax + by + cz).(d + e + f)$$

$$adx + aex + afx + bdy + bey + bfy + cdz + cez + cfz$$

```
DeclareNonCommutative /@ {a, b, c, d, e, f};

DotExpand[DOT[a x + b y + c z, d + e + f]]
```

$$xa.d + xa.e + xa.f + yb.d + yb.e + yb.f + zc.d + zc.e + zc.f$$

```
UnDeclareNonCommutative /@ {a, b, c, d, e, f};

DotExpand[DOT[a x + b y + c z, d + e + f]]
```

$$adx + aex + afx + bdy + bey + bfy + cdz + cez + cfz$$

7.9 DotSimplify

DotSimplify[exp] expands and reorders noncommutative terms in exp. Simplifying relations may be specified by the option **DotSimplifyRelations** or by **Commutator** and **AntiCommutator** definitions. Whether exp is expanded noncommutatively depends on the option **Expanding**.

7.9.1 See also

Overview, AntiCommutator, Commutator, Calc.

7.9.2 Examples

```
UnDeclareAllCommutators[]

UnDeclareAllAntiCommutators[]
```

```
GA[\[Mu]] . (2 GS[p] - GS[q]) . GA[\[Nu]]

DotSimplify[%]
```

$$\bar{\gamma}^{\mu}.\left(2\bar{\gamma}\cdot\overline{p}-\bar{\gamma}\cdot\overline{q}\right).\bar{\gamma}^{\nu}$$

$$2\bar{\gamma}^{\mu}.\left(\bar{\gamma}\cdot\overline{p}\right).\bar{\gamma}^{\nu}-\bar{\gamma}^{\mu}.\left(\bar{\gamma}\cdot\overline{q}\right).\bar{\gamma}^{\nu}$$

```
DeclareNonCommutative[a, b, c]

a . (b - z c) . a

DotSimplify[%]
```

$$a.(b-cz).a$$

$$a.b.a-za.c.a$$

```
Commutator[a, c] = 1

DotSimplify[a . (b - z c) . a]
```

$$1$$

$$a.b.a-z(c.a.a+a)$$

```
Commutator[a, c] =.

DotSimplify[a . (b - z c) . a]
```

$$a.b.a-za.c.a$$

```
AntiCommutator[b, a] = c

DotSimplify[a . (b - z c) . a]
```

$$c$$

$$-a.a.b - za.c.a + a.c$$

```
AntiCommutator[b, a] =.

DotSimplify[a . (b - z c) . a, DotSimplifyRelations -> {a . c -> 1/z}]
```

$$a.b.a - a$$

```
UnDeclareNonCommutative[a, b, c]

DeclareNonCommutative[x]

DotSimplify[x . x . x]
```

$$x.x.x$$

```
DotSimplify[x . x . x, DotPower -> True]

UnDeclareNonCommutative[x]
```

$$x^3$$

Check some relations between noncommutative expressions involving two operators Q and P

```
DeclareNonCommutative[Q, P]
```

```
lhs = (Q . Commutator[Q, P] + Commutator[Q, P] . Q)/2

rhs = Commutator[Q, Q . P + P . Q]/2

DotSimplify[lhs - rhs]

% // ExpandAll
```

$$\frac{1}{2}(Q.[Q,P]+[Q,P].Q)$$

$$\frac{1}{2}[Q,P.Q+Q.P]$$

$$\frac{1}{2}(P.Q.Q-Q.Q.P)+\frac{1}{2}(Q.Q.P-P.Q.Q)$$

$$0$$

```
Commutator[Q, P] = I;
```

Introduce the dilation operator D from the affine quantization and verify that $[Q,D] = i\hbar$ (cf. arXiv:2108.10713)

```
DOp = (Q . P + P . Q)/2;
```

```
Commutator[Q, DOp]

% // DotSimplify // ExpandAll
```

$$\left[Q,\frac{1}{2}(P.Q+Q.P)\right]$$

$$iQ$$

```
UnDeclareAllCommutators[]

UnDeclareAllAntiCommutators[]
```

7.10 FCMatrixIsolate

`FCMatrixIsolate[exp]` wraps the occurring Dirac, Pauli and color objects into heads specified by the user.

7.10.1 See also

Overview, FCDiracIsolate, FCColorIsolate, FCPauliIsolate.

7.10.2 Examples

```
ex = -e eQ gs Spinor[Momentum[k2], mu, 1] . GS[Polarization[k1, -I,
       Transversality -> True]] . (mu + GS[k1 + k2]) . GS[Polarization[p2,
       I]] . Spinor[Momentum[p1], mu, 1] FAD[{-k1 - k2, mu}, Dimension ->
       4]*
    SUNTF[{Glu3}, Col4, Col1] - e eQ gs DCHN[Spinor[Momentum[k2], mu,
      1], i] DCHN[GS[Polarization[p2, I]] . (mu + GS[k2 - p2]) .
      GS[Polarization[k1,
        -I, Transversality -> True]], i, j] DCHN[Spinor[Momentum[p1], mu,
       1], j]*
    FAD[{-k2 + p2, mu}, Dimension -> 4] SUNTF[{Glu3}, Col4, Col1]
```

$$-\frac{e\ \mathrm{eQ}\ \mathrm{gs} T^{\mathrm{Glu3}}_{\mathrm{Col4}\,\mathrm{Col1}} \left(\varphi(\overline{\mathrm{k2}}, \mathrm{mu})\right) . (\bar{\gamma} \cdot \bar{\varepsilon}^*(\mathrm{k1})) . \left(\bar{\gamma} \cdot \left(\overline{\mathrm{k1}} + \overline{\mathrm{k2}}\right) + \mathrm{mu}\right) . (\bar{\gamma} \cdot \bar{\varepsilon}(\mathrm{p2})) . \left(\varphi(\overline{\mathrm{p1}}, \mathrm{mu})\right)}{(-\overline{\mathrm{k1}} - \overline{\mathrm{k2}})^2 - \mathrm{mu}^2}$$
$$-\frac{e\ \mathrm{eQ}\ \mathrm{gs} T^{\mathrm{Glu3}}_{\mathrm{Col4}\,\mathrm{Col1}} \left(\varphi(\overline{\mathrm{k2}}, \mathrm{mu})\right)_i \left(\varphi(\overline{\mathrm{p1}}, \mathrm{mu})\right)_j \left((\bar{\gamma} \cdot \bar{\varepsilon}(\mathrm{p2})) . \left(\bar{\gamma} \cdot \left(\overline{\mathrm{k2}} - \overline{\mathrm{p2}}\right) + \mathrm{mu}\right) . (\bar{\gamma} \cdot \bar{\varepsilon}^*(\mathrm{k1}))\right)_{ij}}{(\overline{\mathrm{p2}} - \overline{\mathrm{k2}})^2 - \mathrm{mu}^2}$$

```
FCMatrixIsolate[ex, FCDiracIsolate -> {dch}, FCColorIsolate -> {cch},
  FCPauliIsolate -> {pch}, Head -> re, FCE -> True]
```

$$\begin{aligned}
&\mathrm{cch}\left(T^{\mathrm{Glu3}}_{\mathrm{Col4}\,\mathrm{Col1}}\right)\ \mathrm{re}\left(-\frac{e\ \mathrm{eQ}\ \mathrm{gs}}{(\overline{\mathrm{p2}} - \overline{\mathrm{k2}})^2 - \mathrm{mu}^2}\right)\ \mathrm{dch}\left(\left(\varphi(\overline{\mathrm{k2}}, \mathrm{mu})\right)_i (\varphi(\overline{\mathrm{p1}},\right.\\
&\left.\mathrm{mu})\right)_j \left((\bar{\gamma} \cdot \bar{\varepsilon}(\mathrm{p2})) . \left(\bar{\gamma} \cdot \left(\overline{\mathrm{k2}} - \overline{\mathrm{p2}}\right) + \mathrm{mu}\right) . (\bar{\gamma} \cdot \bar{\varepsilon}^*(\mathrm{k1}))\right)_{ij}\Big)\\
&+ \mathrm{cch}\left(T^{\mathrm{Glu3}}_{\mathrm{Col4}\,\mathrm{Col1}}\right)\ \mathrm{re}\left(-\frac{e\ \mathrm{eQ}\ \mathrm{gs}}{(-\overline{\mathrm{k1}} - \overline{\mathrm{k2}})^2 - \mathrm{mu}^2}\right)\ \mathrm{dch}\left(\left(\varphi(\overline{\mathrm{k2}},\right.\right.\\
&\left.\left.\mathrm{mu})\right) . (\bar{\gamma} \cdot \bar{\varepsilon}^*(\mathrm{k1})) . \left(\bar{\gamma} \cdot \left(\overline{\mathrm{k1}} + \overline{\mathrm{k2}}\right) + \mathrm{mu}\right) . (\bar{\gamma} \cdot \bar{\varepsilon}(\mathrm{p2})) . \left(\varphi(\overline{\mathrm{p1}}, \mathrm{mu})\right)\right)
\end{aligned}$$

7.11 FCMatrixProduct

`FCMatrixProduct[mat1, mat2, ...]` can be used to obtain products of matrices with entries containing noncommutative symbols. Using the usual **`Dot`** on such matrices would otherwise destroy the original ordering.

The resulting expression can be then further simplified using **`DotSimplify`**.

7.11.1 See also

Overview, DataType, DeclareNonCommutative, UnDeclareNonCommutative.

7.11.2 Examples

Generic matrices

Consider two generic 2×2-matrices containing noncommutative heads

```
DeclareNonCommutative[opA, opB, opC, opD]
```

```
mat[1] = {{opA[1], opB[1]}, {opC[1], opD[1]}}

mat[2] = {{opA[2], opB[2]}, {opC[2], opD[2]}}
```

$$\begin{pmatrix} \text{opA}(1) & \text{opB}(1) \\ \text{opC}(1) & \text{opD}(1) \end{pmatrix}$$

$$\begin{pmatrix} \text{opA}(2) & \text{opB}(2) \\ \text{opC}(2) & \text{opD}(2) \end{pmatrix}$$

Using the usual **Dot** product the elements of the resulting matrix are now multiplied with each other commutatively. Hence, the result is incorrect.

```
mat[1] . mat[2]
```

$$\begin{pmatrix} \text{opA}(1)\ \text{opA}(2) + \text{opB}(1)\ \text{opC}(2) & \text{opA}(1)\ \text{opB}(2) + \text{opB}(1)\ \text{opD}(2) \\ \text{opA}(2)\ \text{opC}(1) + \text{opC}(2)\ \text{opD}(1) & \text{opB}(2)\ \text{opC}(1) + \text{opD}(1)\ \text{opD}(2) \end{pmatrix}$$

With **FCMatrixProduct** the proper ordering is preserved

```
FCMatrixProduct[mat[1], mat[2]]
```

$$\begin{pmatrix} \text{opA}(1).\text{opA}(2) + \text{opB}(1).\text{opC}(2) & \text{opA}(1).\text{opB}(2) + \text{opB}(1).\text{opD}(2) \\ \text{opC}(1).\text{opA}(2) + \text{opD}(1).\text{opC}(2) & \text{opC}(1).\text{opB}(2) + \text{opD}(1).\text{opD}(2) \end{pmatrix}$$

We can also multiply more than two matrices at once

```
mat[3] = {{opA[3], opB[3]}, {opC[3], opD[3]}}
```

$$\begin{pmatrix} \text{opA}(3) & \text{opB}(3) \\ \text{opC}(3) & \text{opD}(3) \end{pmatrix}$$

```
out = FCMatrixProduct[mat[1], mat[2], mat[3]]
```

$$\begin{pmatrix} \text{opB}(1).(\text{opC}(2).\text{opA}(3) + \text{opD}(2).\text{opC}(3)) + \text{opA}(1).(\text{opA}(2).\text{opA}(3) + \text{opB}(2).\text{opC}(3)) & \text{opA}(1).(\text{opA}(2).\text{opB}(\\ \text{opC}(1).(\text{opA}(2).\text{opA}(3) + \text{opB}(2).\text{opC}(3)) + \text{opD}(1).(\text{opC}(2).\text{opA}(3) + \text{opD}(2).\text{opC}(3)) & \text{opC}(1).(\text{opA}(2).\text{opB}(\end{pmatrix}$$

Now use **`DotSimplify`** to expand noncommutative products

```
DotSimplify[out]
```

$$\begin{pmatrix} \text{opA}(1).\text{opB}(2).\text{opC}(3) + \text{opB}(1).\text{opC}(2).\text{opA}(3) + \text{opA}(1).\text{opA}(2).\text{opA}(3) + \text{opB}(1).\text{opD}(2).\text{opC}(3) & \text{opA}(1).\text{o} \\ \text{opD}(1).\text{opC}(2).\text{opA}(3) + \text{opC}(1).\text{opA}(2).\text{opA}(3) + \text{opC}(1).\text{opB}(2).\text{opC}(3) + \text{opD}(1).\text{opD}(2).\text{opC}(3) & \text{opC}(1).\text{op} \end{pmatrix}$$

Dirac matrices in terms of Pauli matrices

Let us define Dirac matrices in the Dirac basis in terms of Pauli matrices

```
gamma[0] = {{1, 0}, {0, -1}};

gamma[i_] := {{0, CSI[i]}, {-CSI[i], 0}};
```

and express $\gamma^i \gamma^j \gamma^i$ as a 2×2-matrix

```
FCMatrixProduct[gamma[i], gamma[j], gamma[i]]

DotSimplify[%]
```

$$\begin{pmatrix} 0.\left(\overline{\sigma}^j.\left(-\overline{\sigma}^i\right) + 0.0\right) + \overline{\sigma}^i.\left(0.\left(-\overline{\sigma}^i\right) + \left(-\overline{\sigma}^j\right).0\right) & 0.\left(0.\overline{\sigma}^i + \overline{\sigma}^j.0\right) + \overline{\sigma}^i.\left(\left(-\overline{\sigma}^j\right).\overline{\sigma}^i + 0.0\right) \\ 0.\left(0.\left(-\overline{\sigma}^i\right) + \left(-\overline{\sigma}^j\right).0\right) + \left(-\overline{\sigma}^i\right).\left(\overline{\sigma}^j.\left(-\overline{\sigma}^i\right) + 0.0\right) & 0.\left(\left(-\overline{\sigma}^j\right).\overline{\sigma}^i + 0.0\right) + \left(-\overline{\sigma}^i\right).\left(0.\overline{\sigma}^i + \overline{\sigma}^j.0\right) \end{pmatrix}$$

$$\begin{pmatrix} 0 & -\overline{\sigma}^i.\overline{\sigma}^j.\overline{\sigma}^i \\ \overline{\sigma}^i.\overline{\sigma}^j.\overline{\sigma}^i & 0 \end{pmatrix}$$

7.12 FCTraceExpand

`FCTraceExpand[exp]` expands traces of Dirac and $SU(N)$ matrices using linearity of the trace. The traces themselves are not evaluated.

7.12.1 See also

Overview, DiracTrace, SUNTrace.

7.12.2 Examples

```
ex = DiracTrace[GA[\[Mu]] . (GS[p1] + m1) . GA[\[Nu]] . (GS[p2] + m2) .
GA[\[Rho]] +x]
```

$$\text{tr}\left(\bar{\gamma}^{\mu}.\left(\bar{\gamma}\cdot\overline{\text{p1}}+\text{m1}\right).\bar{\gamma}^{\nu}.\left(\bar{\gamma}\cdot\overline{\text{p2}}+\text{m2}\right).\bar{\gamma}^{\rho}+x\right)$$

```
FCTraceExpand[ex]
```

$$\begin{aligned}&\text{m1 m2 tr}\left(\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}\right)+\text{m1 tr}\left(\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\left(\bar{\gamma}\cdot\overline{\text{p2}}\right).\bar{\gamma}^{\rho}\right)\\&+\text{m2 tr}\left(\bar{\gamma}^{\mu}.\left(\bar{\gamma}\cdot\overline{\text{p1}}\right).\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}\right)+\text{tr}\left(\bar{\gamma}^{\mu}.\left(\bar{\gamma}\cdot\overline{\text{p1}}\right).\bar{\gamma}^{\nu}.\left(\bar{\gamma}\cdot\overline{\text{p2}}\right).\bar{\gamma}^{\rho}\right)+\text{tr}(1)x\end{aligned}$$

```
FCTraceExpand[ex, DotSimplify -> False]
```

$$\text{tr}\left(\bar{\gamma}^{\mu}.\left(\bar{\gamma}\cdot\overline{\text{p1}}+\text{m1}\right).\bar{\gamma}^{\nu}.\left(\bar{\gamma}\cdot\overline{\text{p2}}+\text{m2}\right).\bar{\gamma}^{\rho}\right)+\text{tr}(1)x$$

```
FCTraceExpand[ex, DiracTrace -> False]
```

$$\text{tr}\left(\bar{\gamma}^{\mu}.\left(\bar{\gamma}\cdot\overline{\text{p1}}+\text{m1}\right).\bar{\gamma}^{\nu}.\left(\bar{\gamma}\cdot\overline{\text{p2}}+\text{m2}\right).\bar{\gamma}^{\rho}+x\right)$$

```
a*DiracTrace[GA[\[Mu]] . (GS[p1] + m1) . GA[\[Nu]]] +
b*DiracTrace[GA[\[Mu]] . (GS[p2] + m2) . GA[\[Nu]]]

FCTraceExpand[%, Momentum -> {p1}]
```

$$a\ \text{tr}\left(\bar{\gamma}^{\mu}.\left(\bar{\gamma}\cdot\overline{\text{p1}}+\text{m1}\right).\bar{\gamma}^{\nu}\right)+b\ \text{tr}\left(\bar{\gamma}^{\mu}.\left(\bar{\gamma}\cdot\overline{\text{p2}}+\text{m2}\right).\bar{\gamma}^{\nu}\right)$$

$$a\left(\text{m1 tr}\left(\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}\right)+\text{tr}\left(\bar{\gamma}^{\mu}.\left(\bar{\gamma}\cdot\overline{\text{p1}}\right).\bar{\gamma}^{\nu}\right)\right)+b\ \text{tr}\left(\bar{\gamma}^{\mu}.\left(\bar{\gamma}\cdot\overline{\text{p2}}+\text{m2}\right).\bar{\gamma}^{\nu}\right)$$

At the moment **SUNTrace** automatically expands its content, so here **FCTraceExpand** is not needed. However, this may change in future.

```
ex = SUNTrace[SUNT[i, j, k] + SUNT[l, m, n]]
```

$$\text{tr}(T^i.T^j.T^k) + \text{tr}(T^l.T^m.T^n)$$

```
FCTraceExpand[ex]
```

$$\text{tr}(T^i.T^j.T^k) + \text{tr}(T^l.T^m.T^n)$$

```
FCTraceExpand[ex, SUNTrace -> False]
```

$$\text{tr}(T^i.T^j.T^k) + \text{tr}(T^l.T^m.T^n)$$

7.13 FCTraceFactor

`FCTraceFactor[expr]` factors out all expressions inside a trace to which the trace doesn't apply. For example, all objects that are not Dirac matrices can be safely factored out from every Dirac trace.

7.13.1 See also

Overview, DiracTrace, SUNTrace.

7.13.2 Examples

Pull constants out of the Dirac trace

```
FCTraceFactor[DiracTrace[c1 . (c2*(GS[p1] + M)) . GA[\[Mu]] . (c3*(GS[p2] +
M2))]]
```

$$\text{c1 c2 c3 tr}\left(\left(\bar{\gamma}\cdot\overline{\text{p1}}+M\right).\bar{\gamma}^{\mu}.\left(\bar{\gamma}\cdot\overline{\text{p2}}+\text{M2}\right)\right)$$

7.14 NonCommFreeQ

`NonCommFreeQ[exp]` yields **`True`** if **`exp`** contains no non-commutative objects (i.e. those objects which are listed in **`$NonComm`**) or only non-commutative objects inside **`DiracTrace`**s or **`SUNTrace`**s.

7.14.1 See also

Overview, $NonComm, NonCommQ, DiracTrace, SUNTrace.

7.14.2 Examples

7.15 NonCommQ

NonCommQ[exp] yields **True** if **exp** contains non-commutative objects (i.e. those objects which are listed in **$NonComm**) not inside **DiracTrace**s or **SUNTrace**s.

7.15.1 See also

Overview, $NonComm, NonCommFreeQ, DiracTrace, SUNTrace.

7.15.2 Examples

```
NonCommQ[xx + yy]
```

False

```
NonCommQ[GA[\[Mu]] . GS[p + m] . GA[\[Mu]]]
```

True

```
NonCommQ[DCHN[GA[\[Mu]], i, j]]
```

True

7.16 NonCommHeadQ

NonCommHeadQ[exp] yields **True** if the head of exp is a non-commutative object or **Dot**.

7.16.1 See also

Overview, DataType, DeclareNonCommutative, UnDeclareNonCommutative, NonCommFreeQ, NonCommQ

7.16.2 Examples

```
NonCommHeadQ[GA[mu]]
```

True

```
NonCommHeadQ[GA[mu, nu, mu]]
```

True

```
NonCommHeadQ[FV[p, mu]]
```

False

```
NonCommHeadQ[FCI[SUNT[a]]]
```

True

```
NonCommHeadQ[FCI[SUNTF[a, i, j]]]
```

False

7.17 TR

TR[exp] calculates the Dirac trace of **exp**. Depending on the setting of the option **SUNTrace** also a trace over $SU(N)$ objects is performed.

The Mathematica build-in function **Tr** is overloaded to call **TR** if any of **DiracGamma**, **GA**, **GAD**, **GS** or **GSD** are in the expression.

Tr[list] finds the trace of the matrix or tensor list.

Tr[list, f] finds a generalized trace, combining terms with f instead of **Plus**.

Tr[list, f, n] goes down to level **n** in **list**.

Tr[expression] calculates the **DiracTrace**, i.e., **TR[expression]** if any of **DiracGamma**, **GA**, **GAD**, **GS** or **GSD** is present in expression.

7.17.1 See also

Overview, DiracSimplify, DiracTrace, FermionSpinSum, SUNTrace.

7.17.2 Examples

```
GA[\[Mu], \[Nu]]
```

```
TR[%]
```

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}$$

$$4\bar{g}^{\mu\nu}$$

```
TR[(GSD[p] + m) . GAD[\[Mu]] . (GSD[q] - m) . GAD[\[Nu]]]
```

$$-4\left(m^2 g^{\mu\nu} + g^{\mu\nu}(p\cdot q) - p^{\nu}q^{\mu} - p^{\mu}q^{\nu}\right)$$

```
TR[GA[\[Mu], \[Nu], \[Rho], \[Sigma], 5]]
```

$$-4i\bar{\epsilon}^{\mu\nu\rho\sigma}$$

```
TR[GS[p, q, r, s]]
```

$$4\left((\overline{p}\cdot\overline{s})(\overline{q}\cdot\overline{r}) - (\overline{p}\cdot\overline{r})(\overline{q}\cdot\overline{s}) + (\overline{p}\cdot\overline{q})(\overline{r}\cdot\overline{s})\right)$$

```
TR[(GS[p] + m) . GA[\[Mu]] . (GS[q] + m) . GA[\[Mu]], Factoring -> True]
```

$$8\left(2m^2 - \overline{p}\cdot\overline{q}\right)$$

```
TR[GA[\[Alpha], \[Beta]], FCE -> True]
```

$$4\bar{g}^{\alpha\beta}$$

```
GA[\[Mu], \[Nu]] SUNT[b] . SUNT[c] SUNDelta[c, b]
```

```
TR[%, SUNTrace -> False, SUNNToCACF -> True]
```

```
TR[%%, SUNTrace -> True, SUNNToCACF -> True]
```

$$\delta^{bc} T^b . T^c \bar{\gamma}^\mu . \bar{\gamma}^\nu$$

$$4 C_F \bar{g}^{\mu\nu}$$

$$4 C_F \bar{g}^{\mu\nu}$$

```
TR[1, SUNTrace -> False, SUNNToCACF -> True]
```

$$4$$

```
TR[1, SUNTrace -> True, SUNNToCACF -> True]
```

$$4$$

```
Tr[ GA[m, n]]
```

$$4 \bar{g}^{mn}$$

7.18 Tr2

If **exp** contains **DiracTrace**s, **Tr2[exp]** simplifies **exp** and does the Dirac traces unless more that 4 gamma matrices and **DiracGamma[5]** occur. **Tr2[exp]** also separates the color-structure, and takes the color trace if **Tf** occurs in **exp**. If **exp** does not contain **DiracTrace**s, **Tr2[exp]** takes the Dirac trace.

7.18.1 See also

Overview, Tr, Tf, DiracTrace, DiracSimplify, SUNTrace.

7.18.2 Examples

7.19 UnDeclareAllAntiCommutators

UnDeclareAllAntiCommutators[] undeclares all user-defined anticommutators.

7.19.1 See also

Overview, AntiCommutator, CommutatorExplicit, DeclareNonCommutative, DotSimplify.

7.19.2 Examples

```
DeclareNonCommutative[a, b, c, d]

AntiCommutator[a, b] = x1;

AntiCommutator[c, d] = x2;

DotSimplify[a . b . c . d]
```

$$b.a.d.c - \text{x2}\, b.a - \text{x1}\, d.c + \text{x1}\ \text{x2}$$

```
UnDeclareAllAntiCommutators[]

DotSimplify[a . b . c . d]
```

$$a.b.c.d$$

7.20 UnDeclareAllCommutators

`UnDeclareAllCommutators[]` undeclares all user-defined commutators.

7.20.1 See also

Overview, Commutator, CommutatorExplicit, DeclareNonCommutative, DotSimplify.

7.20.2 Examples

```
DeclareNonCommutative[a, b, c, d]

Commutator[a, b] = x1;

Commutator[c, d] = x2;

DotSimplify[a . b . c . d]
```

$$b.a.d.c + \text{x2}\, b.a + \text{x1}\, d.c + \text{x1}\ \text{x2}$$

```
UnDeclareAllCommutators[]

DotSimplify[a . b . c . d]
```

$$a.b.c.d$$

7.21 UnDeclareAntiCommutator

UnDeclareAntiCommutator[a, b] undeclares the value assigned to the anticommutator of **a** and **b**.

7.21.1 See also

Overview, Commutator, CommutatorExplicit, DeclareNonCommutative, DotSimplify.

7.21.2 Examples

```
AntiCommutator[QuantumField[FCPartialD[LorentzIndex[xxx_]], A],
QuantumField[A]] = 0;
```

```
QuantumField[A] . QuantumField[A] . LeftPartialD[\[Nu]]

ExpandPartialD[%]
```

$$A.A.\overleftarrow{\partial}_{\nu}$$

$$0$$

```
UnDeclareAntiCommutator[QuantumField[FCPartialD[LorentzIndex[xxx_]], A],
QuantumField[A]];
```

```
ExpandPartialD[QuantumField[A] . QuantumField[A] . LeftPartialD[\[Nu]]]
```

$$A.\left(\left(\partial_{\nu}A\right)\right)+\left(\left(\partial_{\nu}A\right)\right).A$$

7.22 UnDeclareCommutator

UnDeclareCommutator[a, b] undeclares the value assigned to the commutator of **a** and **b**.

7.22.1 See also

Overview, Commutator, CommutatorExplicit, DeclareNonCommutative, DotSimplify.

7.22.2 Examples

```
Commutator[QuantumField[FCPartialD[LorentzIndex[xxx_]], A],
QuantumField[A]] = 0;
```

```
QuantumField[A] . QuantumField[A] . LeftPartialD[\[Nu]] . QuantumField[A] .
QuantumField[A] . LeftPartialD[\[Nu]]

ExpandPartialD[%]
```

$$A.A.\overleftarrow{\partial}_\nu.A.A.\overleftarrow{\partial}_\nu$$

$$6A.A.((\partial_\nu A)).((\partial_\nu A)) + A.(\partial_\nu\partial_\nu A).A.A + (\partial_\nu\partial_\nu A).A.A.A$$

```
UnDeclareCommutator[QuantumField[FCPartialD[LorentzIndex[xxx_]], A],
QuantumField[A]];
```

```
QuantumField[A] . QuantumField[A] . LeftPartialD[\[Nu]] . QuantumField[A] .
QuantumField[A] . LeftPartialD[\[Nu]]

ExpandPartialD[%]
```

$$A.A.\overleftarrow{\partial}_\nu.A.A.\overleftarrow{\partial}_\nu$$

$A.((\partial_\nu A)).A.((\partial_\nu A))$
$+A.((\partial_\nu A)).((\partial_\nu A)).A+((\partial_\nu A)).A.A.((\partial_\nu A))+((\partial_\nu A)).A.((\partial_\nu A)).A+2((\partial_\nu A)).((\partial_\nu A)).A.A+A.(\partial_\nu\partial_\nu A).A.A+($

7.23 UnDeclareNonCommutative

`UnDeclareNonCommutative[a, b, ...]` undeclares **`a, b, ...`** to be noncommutative, i.e., **`DataType[a, b, ..., NonCommutative]`** is set to **`False`**.

7.23.1 See also

Overview, DataType, DeclareNonCommutative.

7.23.2 Examples

```
DeclareNonCommutative[x]
```

As a side-effect of DeclareNonCommutative x is declared to be of DataType NonCommutative.

```
DataType[x, NonCommutative]
```

True

The inverse operation is UnDeclareNonCommutative.

```
UnDeclareNonCommutative[x]

DataType[x, NonCommutative]
```

False

```
DeclareNonCommutative[y, z]

DataType[y, z, NonCommutative]
```

{True, True}

```
UnDeclareNonCommutative[y, z]

DataType[y, z, NonCommutative]
```

{False, False}

8 $SU(N)$ algebra

8.1 FCColorIsolate

FCColorIsolate[exp] wraps colored objects (**SUNT**, **SUNF** etc.) into heads specified by the user.

8.1.1 See also

Overview, SUNT, SUNF.

8.1.2 Examples

FCColorIsolate provides an easy way to extract the color structures present in the expression (e.g. an amplitude)

```
amp = (Spinor[Momentum[p2], SMP["m_u"], 1] . (-I GA[\[Mu]] SMP["g_s"]
SUNTF[{Glu2},
        Col3, Col5]) . (GS[-k1 + p2] + SMP["m_u"]) . (-I GA[\[Nu]]
SMP["g_s"] SUNTF[{Glu4},
        Col5, Col1]) . Spinor[Momentum[p1], SMP["m_u"], 1] FAD[{k1 - p2,
      SMP["m_u"]}, Dimension -> 4] FV[Polarization[k1, I], \[Mu]]
FV[Polarization[k2,
      -I], \[Nu]] + Spinor[Momentum[p2], SMP["m_u"], 1] . (-I GA[\[Nu]]
      SMP["g_s"] SUNTF[{Glu4},
        Col3, Col5]) . (GS[k2 + p2] + SMP["m_u"]) . (-I GA[\[Mu]]
SMP["g_s"] SUNTF[{Glu2},
        Col5, Col1]) . Spinor[Momentum[p1], SMP["m_u"], 1] FAD[{-k2 - p2,
SMP["m_u"]},
      Dimension -> 4] FV[Polarization[k1, I], \[Mu]] FV[Polarization[k2,
-I],
      \[Nu]] - Spinor[Momentum[p2], SMP["m_u"], 1] . (-I GA[Lor3]
      SMP["g_s"] SUNTF[{Glu5},
        Col3, Col1]) . Spinor[Momentum[p1], SMP["m_u"], 1] FAD[-k1 + k2,
      Dimension -> 4] FV[Polarization[k1, I], \[Mu]] FV[Polarization[k2,
-I], \[Nu]] MT[Lor3,
      Lor4] (FV[2 k1 - k2, \[Nu]] MT[Lor4, \[Mu]] + FV[-k1 + 2 k2, \[Mu]]
MT[Lor4, \[Nu]] + FV[-k1 - k2,
        Lor4] MT[\[Mu], \[Nu]]) SMP["g_s"] SUNF[Glu2, Glu4, Glu5])
```

$$\frac{\bar{\varepsilon}^{\mu}(\text{k1})\bar{\varepsilon}^{*\nu}(\text{k2})\left(\varphi(\overline{\text{p2}},m_u)\right).\left(-ig_s\bar{\gamma}^{\mu}T^{\text{Glu2}}_{\text{Col3 Col5}}\right).\left(\bar{\gamma}\cdot\left(\overline{\text{p2}}-\overline{\text{k1}}\right)+m_u\right).\left(-ig_s\bar{\gamma}^{\nu}T^{\text{Glu4}}_{\text{Col5 Col1}}\right).\left(\varphi(\overline{\text{p1}},m_u)\right)}{(\overline{\text{k1}}-\overline{\text{p2}})^2-m_u^2}$$
$$+\frac{\bar{\varepsilon}^{\mu}(\text{k1})\bar{\varepsilon}^{*\nu}(\text{k2})\left(\varphi(\overline{\text{p2}},m_u)\right).\left(-ig_s\bar{\gamma}^{\nu}T^{\text{Glu4}}_{\text{Col3 Col5}}\right).\left(\bar{\gamma}\cdot\left(\overline{\text{k2}}+\overline{\text{p2}}\right)+m_u\right).\left(-ig_s\bar{\gamma}^{\mu}T^{\text{Glu2}}_{\text{Col5 Col1}}\right).\left(\varphi(\overline{\text{p1}},m_u)\right)}{(-\overline{\text{k2}}-\overline{\text{p2}})^2-m_u^2}$$
$$-\frac{1}{(\overline{\text{k2}}-\overline{\text{k1}})^2}g_s\bar{g}^{\text{Lor3 Lor4}}\bar{\varepsilon}^{\mu}(\text{k1})\bar{\varepsilon}^{*\nu}(\text{k2})f^{\text{Glu2 Glu4 Glu5}}\left(\bar{g}^{\text{Lor4}\mu}\left(2\overline{\text{k1}}-\overline{\text{k2}}\right)^{\nu}+\bar{g}^{\text{Lor4}\nu}\left(2\overline{\text{k2}}-\overline{\text{k1}}\right)^{\mu}\right.$$
$$\left.+\bar{g}^{\mu\nu}\left(-\overline{\text{k1}}-\overline{\text{k2}}\right)^{\text{Lor4}}\right)\left(\varphi(\overline{\text{p2}},m_u)\right).\left(-ig_s\bar{\gamma}^{\text{Lor3}}T^{\text{Glu5}}_{\text{Col3 Col1}}\right).\left(\varphi(\overline{\text{p1}},m_u)\right)$$

```
ampIso = FCColorIsolate[amp, Head -> colorS]
```

$$-\frac{g_s^2\bar{\varepsilon}^{\mu}(\text{k1})\bar{\varepsilon}^{*\nu}(\text{k2})\,\text{colorS}\left(T^{\text{Glu4}}_{\text{Col5 Col1}}T^{\text{Glu2}}_{\text{Col3 Col5}}\right)\left(\varphi(\overline{\text{p2}},m_u)\right).\bar{\gamma}^{\mu}.\left(\bar{\gamma}\cdot\left(\overline{\text{p2}}-\overline{\text{k1}}\right)+m_u\right).\bar{\gamma}^{\nu}.\left(\varphi(\overline{\text{p1}},m_u)\right)}{(\overline{\text{k1}}-\overline{\text{p2}})^2-m_u^2}$$
$$-\frac{g_s^2\bar{\varepsilon}^{\mu}(\text{k1})\bar{\varepsilon}^{*\nu}(\text{k2})\,\text{colorS}\left(T^{\text{Glu2}}_{\text{Col5 Col1}}T^{\text{Glu4}}_{\text{Col3 Col5}}\right)\left(\varphi(\overline{\text{p2}},m_u)\right).\bar{\gamma}^{\nu}.\left(\bar{\gamma}\cdot\left(\overline{\text{k2}}+\overline{\text{p2}}\right)+m_u\right).\bar{\gamma}^{\mu}.\left(\varphi(\overline{\text{p1}},m_u)\right)}{(-\overline{\text{k2}}-\overline{\text{p2}})^2-m_u^2}$$
$$+\frac{1}{(\overline{\text{k2}}-\overline{\text{k1}})^2}ig_s^2\bar{g}^{\text{Lor3 Lor4}}\bar{\varepsilon}^{\mu}(\text{k1})\bar{\varepsilon}^{*\nu}(\text{k2})\left(\bar{g}^{\mu\nu}\left(-\overline{\text{k1}}-\overline{\text{k2}}\right)^{\text{Lor4}}+\bar{g}^{\text{Lor4}\nu}\left(2\overline{\text{k2}}-\overline{\text{k1}}\right)^{\mu}\right.$$
$$\left.+\bar{g}^{\text{Lor4}\mu}\left(2\overline{\text{k1}}-\overline{\text{k2}}\right)^{\nu}\right)\,\text{colorS}\left(T^{\text{Glu5}}_{\text{Col3 Col1}}f^{\text{Glu2 Glu4 Glu5}}\right)\left(\varphi(\overline{\text{p2}},m_u)\right).\bar{\gamma}^{\text{Lor3}}.\left(\varphi(\overline{\text{p1}},m_u)\right)$$

Now that all color structures are wrapped into the head **colorS** it is easy to extract them to a separate list

```
Cases2[ampIso, colorS]
```

$$\left\{\text{colorS}\left(T^{\text{Glu2}}_{\text{Col5 Col1}}T^{\text{Glu4}}_{\text{Col3 Col5}}\right),\text{colorS}\left(T^{\text{Glu4}}_{\text{Col5 Col1}}T^{\text{Glu2}}_{\text{Col3 Col5}}\right),\text{colorS}\left(T^{\text{Glu5}}_{\text{Col3 Col1}}f^{\text{Glu2 Glu4 Glu5}}\right)\right\}$$

This way we obtain a sorted list of all unique color structures in **amp**.

```
ClearAll[amp, ampIso, colorS]
```

8.2 CalcColorFactor

CalcColorFactor[exp] calculates the color factor of exp. **CalcColorFactor** is useful for application on FeynArts produced amplitudes.

8.2.1 See also

Overview, SUNSimplify.

8.2.2 Examples

```
CalcColorFactor[SUNF[a, b, c] SUNF[a, b, d]]
```

$$C_A \delta^{cd}$$

8.3 SUNDeltaContract

SUNDeltaContract[exp] substitutes for all **SUNDelta** in **exp SUNDeltaContract**, contracts the adjoint SU(N) indices and resubstitutes **SUNDelta**. **SUNDeltaContract[i, j]** is the Kronecker-delta for SU(N) in the adjoint representation with contraction properties. It wraps the head SUNIndex around its arguments.

8.3.1 See also

Overview, SUNDelta, SUNIndex.

8.3.2 Examples

```
SUNDelta[SUNIndex[a], SUNIndex[b]]^2

SUNDeltaContract[%]
```

$$\left(\delta^{ab}\right)^2$$

$$N^2 - 1$$

8.4 SUNFDeltaContract

SUNFDeltaContract[exp] substitutes for all **SUNFDelta** in exp **SUNFDeltaContract**, contracts the fundamental SU(N) indices and resubstitutes **SUNFDelta**.**SUNFDeltaContract[i, j]** is the Kronecker-delta for SU(N) in the fundamental representation with contraction properties. It wraps the head **SUNFIndex** around its arguments.

8.4.1 See also

Overview, SUNFDelta, SUNFIndex.

8.4.2 Examples

```
SUNFDelta[SUNFIndex[a], SUNFIndex[b]]^2

SUNFDeltaContract[%]
```

$$\delta_{ab}^2$$

$$N$$

8.5 SUNSimplify

SUNSimplify[exp] simplifies color algebraic expressions involving color matrices with implicit (**SUNT**) or explicit fundamental indices (**SUNTF**) as well as structure constants (**SUND**, **SUNF**) and Kronecker deltas (**SD**, **SDF**).

If the option **Explicit** is set to **True** (default is **False**), the structure constants will be rewritten in terms of traces. However, since traces with 2 or 3 color matrices are by default converted back into structure constants, you must also set the option **SUNTraceEvaluate** to **False** (default is **Automatic**) in order to have unevaluated color traces in the output.

8.5.1 See also

Overview, SUNTrace, SUNT, SUNTF, SUNF, SUND, SUNTraceEvaluate.

8.5.2 Examples

```
SUNDelta[a, b] SUNDelta[b, c]

SUNSimplify[%]
```

$$\delta^{ab}\delta^{bc}$$

$$\delta^{ac}$$

```
SUNT[a] . SUNT[a]

SUNSimplify[%]
```

$$T^a.T^a$$

$$C_F$$

```
SUNSimplify[SUNT[a] . SUNT[a], SUNNToCACF -> False]
```

$$\frac{N^2-1}{2N}$$

```
SUNF[a, r, s] SUNF[b, r, s]

SUNSimplify[%]
```

$$f^{ars}\, f^{brs}$$

$$C_A \delta^{ab}$$

```
SUNF[a, b, c]  SUNF[a, b, c]

SUNSimplify[%]
```

$$\left(f^{abc}\right)^2$$

$$2C_A^2 C_F$$

```
SUNF[a, b, c] SUNF[d, b, c]

SUNSimplify[%]
```

$$f^{abc}\, f^{dbc}$$

$$C_A \delta^{ad}$$

```
SUNF[a, b, c] SUND[d, b, c]

SUNSimplify[%, Explicit -> True]
```

$$d^{bcd}\, f^{abc}$$

$$0$$

```
SUND[a, b, c] SUND[a, b, c]
```

```
SUNSimplify[%, SUNNToCACF -> False] // Factor2
```

$$\left(d^{abc}\right)^2$$

$$\frac{\left(1-N^2\right)\left(4-N^2\right)}{N}$$

```
SUNSimplify[SUND[a, b, c] SUND[e, b, c], SUNNToCACF -> False] // Simplify
```

$$\frac{(N-2)(N+2)\delta^{ae}}{N}$$

```
SUNSimplify[SUNF[a, b, c], Explicit -> True]
```

$$f^{abc}$$

```
SUNSimplify[SUNF[a, b, c], Explicit -> True, SUNTraceEvaluate -> False]
```

$$2i\ \mathrm{tr}\left(T^a.T^c.T^b\right)-2i\ \mathrm{tr}\left(T^a.T^b.T^c\right)$$

```
SUNSimplify[SUND[a, b, c], Explicit -> True]
```

$$d^{abc}$$

```
SUNSimplify[SUND[a, b, c], Explicit -> True, SUNTraceEvaluate -> False]
```

$$2\ \mathrm{tr}\left(T^a.T^b.T^c\right)+2\ \mathrm{tr}\left(T^a.T^c.T^b\right)$$

```
SUNF[a, b, c] SUNT[c, b, a]
```

```
SUNSimplify[%]
```

$$f^{abc}T^c.T^b.T^a$$

$$-\frac{1}{2}iC_AC_F$$

```
SUNF[a, b, e] SUNF[c, d, e] + SUNF[a, b, z] SUNF[c, d, z]
SUNSimplify[%, SUNIndexNames -> {j}]
```

$$f^{abe} f^{cde} + f^{abz} f^{cdz}$$

$$2 f^{abj} f^{cdj}$$

```
SUNSimplify[1 - SD[i, i]]
```

$$2 - C_A^2$$

```
SUNSimplify[SUNF[a, b, c] SUND[d, b, c]]
```

$$0$$

```
SUNSimplify[SUNF[a, b, c] SUND[a, b, d]]
```

$$0$$

```
SUNSimplify[SUNF[a, b, c] SUND[a, d, c]]
```

$$0$$

```
SUNSimplify[SUND[a, b, c] SUND[d, b, c]]
```

$$-\left(\left(4 - C_A^2\right) \delta^{ad} \left(C_A - 2 C_F\right)\right)$$

```
SUNSimplify[SUNTrace[SUNT[i1, i2, i1, i2]], FCE -> True]
```

$$-\frac{C_F}{2}$$

8.6 SUNTrace

SUNTrace[exp] is the head of color traces. By default the trace is not evaluated. The evaluation occurs only when the option **SUNTraceEvaluate** is set to **True**. It is recommended to use **SUNSimplify**, which will automatically evaluate all color traces involving 2 or 3 matrices in the input expression.

8.6.1 See also

Overview, SUNSimplify, SUNT, SUNTF, SUNF, SUND, SUNTraceEvaluate.

8.6.2 Examples

```
SUNTrace[SUNT[a, b]]
```

$$\text{tr}\left(T^a.T^b\right)$$

```
SUNTrace[SUNT[a, b], SUNTraceEvaluate -> True]
```

$$\frac{\delta^{ab}}{2}$$

```
SUNTrace[SUNT[a, b]] // SUNSimplify
```

$$\frac{\delta^{ab}}{2}$$

```
SUNTrace[SUNT[a, b, c]] // SUNSimplify
```

$$\frac{d^{abc}}{4}+\frac{1}{4}if^{abc}$$

```
SUNTrace[SUNT[a, b, c, d]] // SUNSimplify[#, SUNTraceEvaluate -> True,
SUNIndexNames -> {j}] &
```

$$\begin{aligned}&\frac{1}{4}\delta^{ad}\left(C_A-2C_F\right)\delta^{bc}-\frac{1}{4}\delta^{ac}\left(C_A-2C_F\right)\delta^{bd}+\frac{1}{4}\delta^{ab}\left(C_A-2C_F\right)\delta^{cd}\\&-\frac{1}{8}if^{adj}d^{bcj}+\frac{1}{8}id^{adj}f^{bcj}+\frac{1}{8}d^{adj}d^{bcj}-\frac{1}{8}d^{bdj}d^{acj}+\frac{1}{8}d^{cdj}d^{abj}\end{aligned}$$

```
SUNTrace[SUNT[a, b, c, a, b, c]] // SUNSimplify
```

$$\frac{1}{4}\left(C_A^2+1\right) C_F \left(C_A-2 C_F\right)$$

9 Loop integrals

9.1 A0

A0[m^2] is the Passarino-Veltman one-point integral A_0..

9.1.1 See also

Overview, B0, C0, D0, PaVe.

9.1.2 Examples

By default A_0 is not expressed in terms of B_0.

```
A0[m^2]
```

$$\mathrm{A}_0\left(m^2\right)$$

```
SetOptions[A0, A0ToB0 -> True];
A0[m^2]
```

$$-\frac{2m^2\ \mathrm{B}_0\left(0, m^2, m^2\right)}{2-D}$$

```
SetOptions[A0, A0ToB0 -> False];
```

According to the rules of dimensional regularization $A_0(0)$ is set to 0.

```
A0[0]
```

$$0$$

```
A0[SmallVariable[M^2]]
```

$$0$$

9.2 A00

A00[m^2] is the Passarino-Veltman coefficient function A_{00}, i.e. the coefficient function multiplying $g^{\mu\nu}$. The argument is a scalar and has mass dimension 2.

9.2.1 See also

Overview, A0, B0, C0, D0, PaVe.

9.2.2 Examples

A_{00} get automatically reduced to A_0

```
A00[m^2]
```

$$\frac{m^2\ \mathrm{A}_0\left(m^2\right)}{D}$$

According to the rules of dimensional regularization $A_{00}(0)$ is set to 0.

```
A00[0]
```

$$0$$

9.3 Apart2

Apart2[expr] partial fractions propagators of the form $1/[(q^2 - m1^2)(q^2 - m2^2)]$.

9.3.1 See also

Overview, FAD, FeynAmpDenominator, ApartFF.

9.3.2 Examples

```
FAD[{q, m}, {q, M}, q - p]

Apart2[%]

StandardForm[FCE[%]]
```

$$\frac{1}{(q^2-m^2)\,.\,(q^2-M^2)\,.(q-p)^2}$$

$$\frac{\frac{1}{(q^2-m^2).(q-p)^2}-\frac{1}{(q^2-M^2).(q-p)^2}}{m^2-M^2}$$

$$\frac{\text{FAD}[\{q,m\},-p+q]-\text{FAD}[\{q,M\},-p+q]}{m^2-M^2}$$

Apart2 can also handle Cartesian propagators with square roots. To disable this mode use *textSqrttotextFalse*

```
int = CFAD[{{k, 0}, {+m^2, -1}, 1}, {{k - p, 0}, {0, -1}, 1}] GFAD[{{DE -
Sqrt[CSPD[k, k]], 1}, 1}]

int // FeynAmpDenominatorCombine // Apart2
```

$$\frac{1}{(\text{DE}-\sqrt{k^2}+i\eta)(k^2+m^2-i\eta).((k-p)^2-i\eta)}$$

$$\frac{\frac{\text{DE}}{(k^2+m^2-i\eta).((k-p)^2-i\eta)}+\frac{1}{(\text{DE}-\sqrt{k^2}+i\eta).((k-p)^2-i\eta)}+\frac{\sqrt{k^2}}{(k^2+m^2-i\eta).((k-p)^2-i\eta)}}{\text{DE}^2+m^2}$$

9.4 ApartFF

`ApartFF[amp, {q1, q2, ...}]` partial fractions loop integrals by decomposing them into simpler integrals that contain only linearly independent propagators. It uses **`FCApart`** as a backend and is equally suitable for 1-loop and multi-loop integrals.

`FCApart` implements an algorithm based on arXiv:1204.2314 by F. Feng that seems to employ a variety Leinartas's algorithm (cf. arXiv:1206.4740). Unlike Feng's $Apart that is applicable to general multivariate polynomials, **`FCApart`** is tailored to work only with FeynCalc's **`FeynAmpDenominator`**, **`Pair`** and **`CartesianPair`** symbols, i.e. it is less general in this respect.

`ApartFF[amp * extraPiece1, extraPiece2, {q1, q2, ...}]` is a special working mode of **`ApartFF`**, where the final result of partial fractioning **`amp*extraPiece1`** is multiplied by **`extraPiece2`**. It is understood, that **`extraPiece1*extraPiece2`** should be unity, e. g. when **`extraPiece1`** is an **`FAD`**, while extraPiece is an **`SPD`** inverse to it. This mode should be useful for nonstandard integrals where the desired partial fraction decomposition can be performed only after multiplying **`amp`** with **`extraPiece1`**.

9.4.1 See also

Overview, FCApart, FeynAmpDenominatorSimplify.

9.4.2 Examples

```
FCClearScalarProducts[]
```

```
SPD[q, q] FAD[{q, m}]

ApartFF[%, {q}]
```

$$\frac{q^2}{q^2 - m^2}$$

$$\frac{m^2}{q^2 - m^2}$$

```
SPD[q, p] SPD[q, r] FAD[{q}, {q - p}, {q - r}]

ApartFF[%, {q}]
```

$$\frac{(p \cdot q)(q \cdot r)}{q^2.(q-p)^2.(q-r)^2}$$

$$\frac{p^2 r^2}{4q^2.(q-r)^2.(q-p)^2} + \frac{p^2 + 2(q \cdot r) + 2r^2}{4q^2.(-p+q+r)^2} + -\frac{p^2}{4q^2.(q-p)^2} - \frac{r^2}{4q^2.(q-r)^2}$$

```
FAD[{q}, {q - p}, {q + p}]

ApartFF[%, {q}]
```

$$\frac{1}{q^2.(q-p)^2.(p+q)^2}$$

$$\frac{1}{p^2 q^2.(q-p)^2} - \frac{1}{p^2 q^2.(q-2p)^2}$$

```
SPD[p, q1] SPD[p, q2]^2 FAD[{q1, m}, {q2, m}, q1 - p, q2 - p, q1 - q2]

ApartFF[%, {q1, q2}]
```

$$\frac{(p\cdot \text{q1})(p\cdot \text{q2})^2}{\left(\text{q1}^2-m^2\right).\left(\text{q2}^2-m^2\right).(\text{q1}-p)^2.(\text{q2}-p)^2.(\text{q1}-\text{q2})^2}$$

$$\begin{aligned}&\frac{\left(m^2+p^2\right)^3}{8\left(\text{q1}^2-m^2\right).\left(\text{q2}^2-m^2\right).(\text{q2}-p)^2.(\text{q1}-\text{q2})^2.(\text{q1}-p)^2}\\&-\frac{\left(m^2+p^2\right)^2}{4\left(\text{q1}^2-m^2\right).\left(\text{q2}^2-m^2\right).(\text{q1}-\text{q2})^2.(\text{q1}-p)^2}\\&+\frac{\left(m^2+p^2\right)\left(m^2+2p^2\right)}{4\,\text{q2}^2.\text{q1}^2.\left((\text{q1}-p)^2-m^2\right).(\text{q1}-\text{q2})^2}-\frac{\left(m^2+p^2\right)(p\cdot \text{q1})}{4\left(\text{q1}^2-m^2\right).\left(\text{q2}^2-m^2\right).(\text{q2}-p)^2.(\text{q1}-\text{q2})^2}\\&-\frac{\left(m^2+p^2\right)(p\cdot \text{q1})}{4\,\text{q2}^2.\text{q1}^2.(\text{q1}-\text{q2})^2.\left((\text{q2}-p)^2-m^2\right)}-\frac{p\cdot \text{q1}}{4\left(\text{q2}^2-m^2\right).(\text{q1}-\text{q2})^2.(\text{q1}-p)^2}\\&-\frac{m^2+p\cdot \text{q1}+p^2}{4\left(\text{q1}^2-m^2\right).(\text{q2}-p)^2.(\text{q1}-\text{q2})^2}+\frac{m^2+2(p\cdot \text{q1})+p^2}{8\left(\text{q1}^2-m^2\right).\left(\text{q2}^2-m^2\right).(\text{q1}-\text{q2})^2}\end{aligned}$$

```
SPD[q, p] FAD[{q, m}, {q - p, 0}]

ApartFF[%, {q}]
```

$$\frac{p\cdot q}{\left(q^2-m^2\right).(q-p)^2}$$

$$\frac{m^2+p^2}{2q^2.\left((q-p)^2-m^2\right)}-\frac{1}{2\left(q^2-m^2\right)}$$

If the propagators should not be altered via momentum shifts (e.g. because they belong to a previously identified topology), use the option **`FDS->False`**

```
int = SPD[q2, p] SPD[q1, p] FAD[{q1, m}, {q2, m}, q1 - p, q2 - p, q2 - q1]
```

$$\frac{(p\cdot \text{q1})(p\cdot \text{q2})}{\left(\text{q1}^2-m^2\right).\left(\text{q2}^2-m^2\right).(\text{q1}-p)^2.(\text{q2}-p)^2.(\text{q2}-\text{q1})^2}$$

```
ApartFF[int, {q1, q2}]
```

$$\frac{\left(m^2+p^2\right)^2}{4\left(\text{q1}^2-m^2\right).\left(\text{q2}^2-m^2\right).(\text{q2}-p)^2.(\text{q1}-\text{q2})^2.(\text{q1}-p)^2}$$
$$-\frac{m^2+p^2}{2\left(\text{q1}^2-m^2\right).\left(\text{q2}^2-m^2\right).(\text{q1}-\text{q2})^2.(\text{q1}-p)^2}+\frac{m^2+p^2}{2\,\text{q2}^2.\text{q1}^2.\left((\text{q1}-p)^2-m^2\right).(\text{q1}-\text{q2})^2}$$
$$-\frac{1}{2\left(\text{q1}^2-m^2\right).(\text{q2}-p)^2.(\text{q1}-\text{q2})^2}+\frac{1}{4\left(\text{q1}^2-m^2\right).\left(\text{q2}^2-m^2\right).(\text{q1}-\text{q2})^2}$$

```
ApartFF[int, {q1, q2}, FDS -> False]
```

$$\frac{\left(m^2+p^2\right)^2}{4\left(\text{q2}^2-m^2\right).(\text{q2}-p)^2.(\text{q2}-\text{q1})^2.\left(\text{q1}^2-m^2\right).(\text{q1}-p)^2}$$
$$-\frac{m^2+p^2}{4\left(\text{q1}^2-m^2\right).(\text{q1}-p)^2.\left(\text{q2}^2-m^2\right).(\text{q2}-\text{q1})^2}+\frac{m^2+p^2}{4\left(\text{q1}^2-m^2\right).(\text{q1}-p)^2.(\text{q2}-p)^2.(\text{q2}-\text{q1})^2}$$
$$-\frac{m^2+p^2}{4\left(\text{q1}^2-m^2\right).\left(\text{q2}^2-m^2\right).(\text{q2}-p)^2.(\text{q2}-\text{q1})^2}+\frac{m^2+p^2}{4(\text{q1}-p)^2.\left(\text{q2}^2-m^2\right).(\text{q2}-p)^2.(\text{q2}-\text{q1})^2}$$
$$-\frac{1}{4(\text{q2}-\text{q1})^2.\left(\text{q1}^2-m^2\right).(\text{q2}-p)^2}-\frac{1}{4(\text{q2}-\text{q1})^2.(\text{q1}-p)^2.\left(\text{q2}^2-m^2\right)}$$
$$+\frac{1}{4(\text{q2}-\text{q1})^2.\left(\text{q1}^2-m^2\right).\left(\text{q2}^2-m^2\right)}+\frac{1}{4(\text{q2}-\text{q1})^2.(\text{q1}-p)^2.(\text{q2}-p)^2}$$

If the partial fractioning should be performed only w. r. t. the denominators but not numerators, use the option **`Numerator->False`**

```
int = FAD[k, p - k, {k, m}] SPD[p, k]
```

$$\frac{k\cdot p}{k^2.(p-k)^2.\left(k^2-m^2\right)}$$

```
ApartFF[int, {k}]
```

$$\frac{m^2+p^2}{2m^2k^2.\left((k-p)^2-m^2\right)}+-\frac{1}{2m^2\left(k^2-m^2\right)}-\frac{p^2}{2m^2k^2.(k-p)^2}$$

```
ApartFF[int, {k}, Numerator -> False]
```

$$\frac{k \cdot p}{m^2 k^2 . ((k-p)^2 - m^2)} - \frac{k \cdot p}{m^2 k^2 . (k-p)^2}$$

Using the option **FeynAmpDenominator ->False** we can specify that integrals without numerators should not be partial fractioned

```
int = FAD[k, p - k, {k, m}] (SPD[q] + SPD[p, k])
```

$$\frac{k \cdot p + q^2}{k^2 . (p-k)^2 . (k^2 - m^2)}$$

```
ApartFF[int, {k}]
```

$$\frac{m^2 + p^2 + 2q^2}{2m^2 k^2 . ((k-p)^2 - m^2)} + -\frac{1}{2m^2 (k^2 - m^2)} - \frac{p^2 + 2q^2}{2m^2 k^2 . (k-p)^2}$$

```
ApartFF[int, {k}, FeynAmpDenominator -> False]
```

$$\frac{q^2}{k^2 . (p-k)^2 . (k^2 - m^2)} + \frac{m^2 + p^2}{2m^2 k^2 . ((k-p)^2 - m^2)} + -\frac{1}{2m^2 (k^2 - m^2)} - \frac{p^2}{2m^2 k^2 . (k-p)^2}$$

The **extraPiece**-trick is useful for cases where a direct partial fractioning is not possible

```
int = (SFAD[{{0, k . l}}, p - k] SPD[k, p])
```

$$\frac{k \cdot p}{(k \cdot l + i\eta) . ((p-k)^2 + i\eta)}$$

Here **ApartFF** cannot do anything

```
ApartFF[int, {k}]
```

$$\frac{k \cdot p}{(k \cdot l + i\eta) . ((p-k)^2 + i\eta)}$$

Multiplying the integral with unity **FAD[k]*SPD[k]** we can cast into a more desirable form

```
ApartFF[int FAD[k], SPD[k], {k}] // ApartFF[#, {k}] &
```

$$\frac{k^2}{2(k\cdot l+i\eta).((p-k)^2+i\eta)}+\frac{p^2}{2(k\cdot l+i\eta).((k-p)^2+i\eta)}$$

Here we need a second call to **ApartFF** since the first execution doesn't drop scaleless integrals or perform any shifts in the denominators.

9.5 B0

B0[pp, ma^2, mb^2] is the Passarino-Veltman two-point integral B_0. All arguments are scalars and have dimension mass squared. If the option **BReduce** is set to **True**, certain **B0**'s are reduced to **A0**'s. Setting the option **B0Unique** to **True** simplifies **B0[a, 0, a]** and **B0[0, 0, a]**.

9.5.1 See also

Overview, B1, B00, B11, PaVe.

9.5.2 Examples

```
B0[SP[p, p], m^2, m^2]
```

$$\mathrm{B}_0\left(\overline{p}^2, m^2, m^2\right)$$

```
B0[0, 0, m^2, B0Unique -> True, B0Real -> True]
```

$$\mathrm{B}_0\left(0, m^2, m^2\right)+1$$

```
B0[m^2, 0, m^2, B0Unique -> True, B0Real -> True]
```

$$\mathrm{B}_0\left(0, m^2, m^2\right)+2$$

```
B0[0, m^2, m^2]
```

$$\mathrm{B}_0\left(0, m^2, m^2\right)$$

9.6 B00

B00[pp, ma^2, mb^2] is the Passarino-Veltman B_{00}-function, i.e., the coefficient function of the metric tensor. All arguments are scalars and have dimension mass squared.

9.6.1 See also

Overview, B0, B1, PaVe.

9.6.2 Examples

```
B00[SPD[p], m^2, M^2]
```

$$\frac{\left(m^2-2mM+M^2-p^2\right)\left(m^2+2mM+M^2-p^2\right)\ \mathrm{B}_0\left(p^2,m^2,M^2\right)}{4(1-D)p^2} + \frac{\mathrm{A}_0\left(M^2\right)\left(m^2-M^2-p^2\right)}{4(1-D)p^2} - \frac{\mathrm{A}_0\left(m^2\right)\left(m^2-M^2+p^2\right)}{4(1-D)p^2}$$

```
B00[SPD[p], m^2, m^2]
```

$$-\frac{\left(4m^2-p^2\right)\ \mathrm{B}_0\left(p^2,m^2,m^2\right)}{4(1-D)} - \frac{\mathrm{A}_0\left(m^2\right)}{2(1-D)}$$

```
B00[SPD[p], m^2, M^2, BReduce -> False]
```

$$\mathrm{B}_{00}\left(p^2,m^2,M^2\right)$$

```
B00[0, m^2, m^2]
```

$$-\frac{m^2\ \mathrm{B}_0\left(0,m^2,m^2\right)}{1-D} - \frac{\mathrm{A}_0\left(m^2\right)}{2(1-D)}$$

```
B00[SmallVariable[M^2], m^2, m^2]
```

$$-\frac{m^2\ \mathrm{B}_0\left(M^2,m^2,m^2\right)}{1-D} - \frac{\mathrm{A}_0\left(m^2\right)}{2(1-D)}$$

9.7 B1

`B1[pp, ma^2, mb^2]` the Passarino-Veltman B_1-function. All arguments are scalars and have dimension mass squared.

9.7.1 See also

Overview, B0, B00, B11, PaVe, PaVeReduce.

9.7.2 Examples

```
B1[SPD[p], m^2, M^2]
```

$$-\frac{\left(m^2-M^2+p^2\right)\ \mathrm{B}_0\left(p^2,m^2,M^2\right)}{2p^2}+\frac{\mathrm{A}_0\left(m^2\right)}{2p^2}-\frac{\mathrm{A}_0\left(M^2\right)}{2p^2}$$

```
B1[SPD[p], m^2, M^2, BReduce -> False]
```

$$\mathrm{B}_1\left(p^2,m^2,M^2\right)$$

```
B1[SP[p], m^2, m^2]
```

$$-\frac{1}{2}\ \mathrm{B}_0\left(\overline{p}^2,m^2,m^2\right)$$

```
B1[SPD[p], m^2, m^2, BReduce -> False]
```

$$\mathrm{B}_1\left(p^2,m^2,m^2\right)$$

```
B1[m^2, m^2, 0]
```

$$\frac{\mathrm{A}_0\left(m^2\right)}{2m^2}-\mathrm{B}_0\left(m^2,0,m^2\right)$$

```
B1[m^2, m^2, 0, BReduce -> False]
```

$$\mathrm{B}_1\left(m^2,m^2,0\right)$$

```
B1[0, 0, m^2]
```

$$\mathrm{B}_1\left(0, 0, m^2\right)$$

```
B1[pp, SmallVariable[SMP["m_e"]^2], Subsuperscript[m, 2, 2]]
```

$$-\frac{(\mathrm{pp} - m_2^2)\ \mathrm{B}_0\left(\mathrm{pp}, m_e^2, m_2^2\right)}{2\ \mathrm{pp}} - \frac{\mathrm{A}_0(m_2^2)}{2\ \mathrm{pp}}$$

9.8 B11

`B11[pp, ma^2, mb^2]` is the Passarino-Veltman B_{11}-function, i.e. the coefficient function of $p^\mu p^\nu$. All arguments are scalars and have dimension mass squared.

9.8.1 See also

Overview, B0, B00, B1, PaVe.

9.8.2 Examples

```
B11[SPD[p], m^2, M^2]
```

$$-\frac{\left(Dm^4 - 2Dm^2M^2 + 2Dm^2p^2 + DM^4 - 2DM^2p^2 + Dp^4 - 4m^2p^2\right)\ \mathrm{B}_0\left(p^2, m^2, M^2\right)}{4(1-D)p^4} + \frac{D\ \mathrm{A}_0\left(m^2\right)\left(m^2 - M^2 + p^2\right)}{4(1-D)p^4} - \frac{\mathrm{A}_0\left(M^2\right)\left(Dm^2 - DM^2 + 3Dp^2 - 4p^2\right)}{4(1-D)p^4}$$

```
B11[SPD[p], m^2, M^2, BReduce -> False]
```

$$\mathrm{B}_{11}\left(p^2, m^2, M^2\right)$$

```
B11[SPD[p], m^2, m^2]
```

$$\frac{\left(4m^2 - Dp^2\right)\ \mathrm{B}_0\left(p^2, m^2, m^2\right)}{4(1-D)p^2} + \frac{(2-D)\ \mathrm{A}_0\left(m^2\right)}{2(1-D)p^2}$$

```
B11[SPD[p], m^2, m^2, BReduce -> False]
```

$$\mathrm{B}_{11}\left(p^2, m^2, m^2\right)$$

```
B11[0, m^2, m^2]
```

$$\frac{1}{3}\,\mathrm{B}_0\left(0, m^2, m^2\right)$$

```
B11[0, m^2, m^2, BReduce -> False]
```

$$\mathrm{B}_{11}\left(0, m^2, m^2\right)$$

```
B11[SmallVariable[M^2], m^2, m^2]
```

$$\frac{m^2\,\mathrm{B}_0\left(M^2, m^2, m^2\right)}{(1-D)M^2} + \frac{(2-D)\,\mathrm{A}_0\left(m^2\right)}{2(1-D)M^2}$$

```
B11[SmallVariable[M^2], m^2, m^2, BReduce -> False]
```

$$\mathrm{B}_{11}\left(M^2, m^2, m^2\right)$$

9.9 C0

`C0[p10, p12, p20, m1^2, m2^2, m3^2]` is the scalar Passarino-Veltman C_0 function. The convention for the arguments is that if the denominator of the integrand has the form $([q^2 - m1^2][(q+p1)^2 - m2^2][(q+p2)^2 - m3^2])$, the first three arguments of C0 are the scalar products $p10 = p1^2$, $p12 = (p1-p2).(p1-p2)$, $p20 = p2^2$.

9.9.1 See also

Overview, B0, D0, PaVe, PaVeOrder.

9.9.2 Examples

```
C0[a, b, c, m12, m22, m32]
```

$$\mathrm{C}_0(a, b, c, \mathrm{m12}, \mathrm{m22}, \mathrm{m32})$$

```
C0[b, a, c, m32, m22, m12] // PaVeOrder
```

$$\mathrm{C}_0(a, b, c, \mathrm{m12}, \mathrm{m22}, \mathrm{m32})$$

```
PaVeOrder[C0[b, a, c, m32, m22, m12], PaVeOrderList -> {c, a}]
```

$$\mathrm{C}_0(c, a, b, \mathrm{m32}, \mathrm{m12}, \mathrm{m22})$$

9.10 CTdec

`CTdec[{{qi, a}, {qj, b}, ...}, {p1, p2, ...}]` or **`CTdec[exp, {{qi, a}, {qj, b}, ...}, {p1, p2, ...}]`** calculates the tensorial decomposition formulas for Cartesian integrals. The more common ones are saved in TIDL.

9.10.1 See also

Overview, Tdec, TIDL, TID.

9.10.2 Examples

Check that $\int d^{D-1}q\, f(p,q) q^i = \frac{p^i}{p^2} \int d^{D-1}q\, f(p,q) p\cdot q$

```
CTdec[{{q, i}}, {p}]
```

$$\left\{\{\mathrm{X1} \to p\cdot q, \mathrm{X2} \to p^2\}, \frac{\mathrm{X1}\, p^i}{\mathrm{X2}}\right\}$$

```
%[[2]] /. %[[1]]
```

$$\frac{p^i (p\cdot q)}{p^2}$$

```
CTdec[{{q, i}}, {p}, List -> False]
```

$$\frac{p^i (p \cdot q)}{p^2}$$

This calculates integral transformation for any $\int d^{D-1}q_1 d^{D-1}q_2 d^{D-1}q_3 f(p, q_1, q_2, q_3) q_1^i q_2^j q_3^k$.

```
CTdec[{{Subscript[q, 1], i}, {Subscript[q, 2], j}, {Subscript[q, 3], k}},
{p}, List -> False]
```

$$\frac{p^k \delta^{ij} (p \cdot q_3) \left((p \cdot q_1)(p \cdot q_2) - p^2 (q_1 \cdot q_2)\right)}{(2-D)p^4}$$
$$+ \frac{p^j \delta^{ik} (p \cdot q_2) \left((p \cdot q_1)(p \cdot q_3) - p^2 (q_1 \cdot q_3)\right)}{(2-D)p^4} + \frac{p^i \delta^{jk} (p \cdot q_1) \left((p \cdot q_2)(p \cdot q_3) - p^2 (q_2 \cdot q_3)\right)}{(2-D)p^4}$$
$$- \frac{p^i p^j p^k \left((D-1)(p \cdot q_1)(p \cdot q_2)(p \cdot q_3) + 2(p \cdot q_1)(p \cdot q_2)(p \cdot q_3) - p^2 (q_1 \cdot q_2)(p \cdot q_3) - p^2 (q_1 \cdot q_3)(p \cdot q_2) - p^2 (q\right.}{(2-D)p^6}$$

```
Contract[% CVD[p, i] CVD[p, j] CVD[p, k]] // Factor
```

$$(p \cdot q_1)(p \cdot q_2)(p \cdot q_3)$$

9.11 Tdec

`Tdec[{{qi, mu}, {qj, nu}, ...}, {p1, p2, ...}]` calculates the tensorial decomposition formulas for Lorentzian integrals. The more common ones are saved in **`TIDL`**.

The automatic symmetrization of the tensor basis is done using Alexey Pak's algorithm described in arXiv:1111.0868.

9.11.1 See also

Overview, TID, TIDL, OneLoopSimplify.

9.11.2 Examples

Check that $\int d^D f(p,q) q^\mu = \frac{p^\mu}{p^2} \int d^D f(p,q) p \cdot q$

```
Tdec[{q, \[Mu]}, {p}]

%[[2]] /. %[[1]]
```

$$\left\{\{\text{X1} \to p \cdot q, \text{X2} \to p^2\}, \frac{\text{X1} p^\mu}{\text{X2}}\right\}$$

$$\frac{p^\mu (p \cdot q)}{p^2}$$

This calculates integral transformation for any $\int d^D q_1 d^D q_2 d^D q_3 \, f(p, q_1, q_2, q_3) q_1^\mu q_2^\nu q_3^\rho$.

```
Tdec[{{Subscript[q, 1], \[Mu]}, {Subscript[q, 2], \[Nu]}, {Subscript[q, 3],
\[Rho]}}, {p}, List -> False]

Contract[% FVD[p, \[Mu]] FVD[p, \[Nu]] FVD[p, \[Rho]]] // Factor
```

$$\frac{p^\rho g^{\mu\nu} (p \cdot q_3) \left((p \cdot q_1)(p \cdot q_2) - p^2 (q_1 \cdot q_2)\right)}{(1-D) p^4}$$
$$+ \frac{p^\nu g^{\mu\rho} (p \cdot q_2) \left((p \cdot q_1)(p \cdot q_3) - p^2 (q_1 \cdot q_3)\right)}{(1-D) p^4} + \frac{p^\mu g^{\nu\rho} (p \cdot q_1) \left((p \cdot q_2)(p \cdot q_3) - p^2 (q_2 \cdot q_3)\right)}{(1-D) p^4}$$
$$- \frac{p^\mu p^\nu p^\rho \left(D (p \cdot q_1)(p \cdot q_2)(p \cdot q_3) + 2 (p \cdot q_1)(p \cdot q_2)(p \cdot q_3) - p^2 (q_1 \cdot q_2)(p \cdot q_3) - p^2 (q_1 \cdot q_3)(p \cdot q_2) - p^2 (q_2 \cdot q_3\right.}{(1-D) p^6}$$

$$(p \cdot q_1)(p \cdot q_2)(p \cdot q_3)$$

9.12 D0

`D0[p10, p12, p23, p30, p20, p13, m1^2, m2^2, m3^2, m4^2 ]` is the Passarino-Veltman D_0 function. The convention for the arguments is that if the denominator of the integrand has the form $([q^2 - m1^2][(q+p1)^2 - m2^2][(q+p2)^2 - m3^2][(q+p3)^2 - m4^2])$, the first six arguments of **`D0`** are the scalar products $p10 = p1^2$, $p12 = (p1-p2)^2$, $p23 = (p2-p3)^2$, $p30 = p3^2$, $p20 = p2^2$, $p13 = (p1-p3)^2$.

9.12.1 See also

Overview, B0, C0, PaVe, PaVeOrder.

9.12.2 Examples

```
D0[p10, p12, p23, p30, p20, p13, m1^2, m2^2, m3^2, m4^2]
```

$$\mathrm{D_0}\left(\mathrm{p10}, \mathrm{p12}, \mathrm{p23}, \mathrm{p30}, \mathrm{p20}, \mathrm{p13}, \mathrm{m1}^2, \mathrm{m2}^2, \mathrm{m3}^2, \mathrm{m4}^2\right)$$

```
PaVeOrder[D0[p10, p12, p23, p30, p20, p13, m1^2, m2^2, m3^2, m4^2],
PaVeOrderList -> {p13, p20}]
```

$$\mathrm{D_0}\left(\mathrm{p10}, \mathrm{p30}, \mathrm{p23}, \mathrm{p12}, \mathrm{p13}, \mathrm{p20}, \mathrm{m2}^2, \mathrm{m1}^2, \mathrm{m4}^2, \mathrm{m3}^2\right)$$

```
PaVeOrder[%]
```

$$\mathrm{D_0}\left(\mathrm{p10}, \mathrm{p12}, \mathrm{p23}, \mathrm{p30}, \mathrm{p20}, \mathrm{p13}, \mathrm{m1}^2, \mathrm{m2}^2, \mathrm{m3}^2, \mathrm{m4}^2\right)$$

9.13 DB0

DB0[p2, m1^2, m2^2] is the derivative of the two-point function **B0[p2, m1^2, m2^2]** with respect to **p2**.

9.13.1 See also

Overview, B0.

9.13.2 Examples

```
D[B0[Subscript[p, 2], Subscript[m, 1]^2, Subscript[m, 2]^2], Subscript[p,
2]]
```

$$\mathrm{DB0}\left(p_2, m_1^2, m_2^2\right)$$

9.14 DB1

DB1[p2, m1^2, m2^2] is the derivative of **B1[p2, m1^2, m2^2]** with respect to **p2**.

9.14.1 See also

Overview, B1.

9.14.2 Examples

```
D[B1[Subscript[p, 2], Subscript[m, 1]^2, Subscript[m, 2]^2], Subscript[p,
2]]
```

$$\frac{\left(m_1^2-m_2^2+p_2\right)\ \mathrm{B}_0\left(p_2,m_1^2,m_2^2\right)}{2p_2^2}-\frac{\mathrm{B}_0\left(p_2,m_1^2,m_2^2\right)}{2p_2}-\frac{\left(m_1^2-m_2^2+p_2\right)\ \mathrm{DB0}\left(p_2,m_1^2,m_2^2\right)}{2p_2}-\frac{\mathrm{A}_0\left(m_1^2\right)}{2p_2^2}+\frac{\mathrm{A}_0\left(m_2^2\right)}{2p_2^2}$$

9.15 FCApart

FCApart[expr, {q1, q2, ...}] is an internal function that partial fractions a loop integral (that depends on **q1**,**q2**, ...) into integrals that contain only linearly independent propagators. The algorithm is largely based on arXiv:1204.2314 by F.Feng. **FCApart** is meant to be applied to single loop integrals only. If you need to perform partial fractioning on an expression that contains multiple loop integrals, use **ApartFF**.

There is actually no reason, why one would want to apply **FCApart** instead of **ApartFF**, except for cases, where **FCApart** is called from a different package that interacts with FeynCalc.

9.15.1 See also

Overview, ApartFF, FeynAmpDenominatorSimplify.

9.15.2 Examples

```
SPD[q, q] FAD[{q, m}]

FCApart[%, {q}]
```

$$\frac{q^2}{q^2-m^2}$$

$$\frac{m^2}{q^2-m^2}$$

```
SPD[q, p] SPD[q, r] FAD[{q}, {q - p}, {q - r}]

FCApart[%, {q}]
```

$$\frac{(p\cdot q)(q\cdot r)}{q^2.(q-p)^2.(q-r)^2}$$

$$\frac{p^2r^2}{4q^2.(q-p)^2.(q-r)^2}+\frac{p^2+2r^2}{4q^2.(-p+q+r)^2}+\frac{q\cdot r}{2q^2.(-p+q+r)^2}+-\frac{p^2}{4q^2.(q-p)^2}-\frac{r^2}{4q^2.(q-r)^2}$$

```
SPD[p, q1] SPD[p, q2]^2 FAD[{q1, m}, {q2, m}, q1 - p, q2 - p, q1 - q2]

FCApart[%, {q1, q2}]
```

$$\frac{(p\cdot \text{q1})(p\cdot \text{q2})^2}{\left(\text{q1}^2-m^2\right).\left(\text{q2}^2-m^2\right).(\text{q1}-p)^2.(\text{q2}-p)^2.(\text{q1}-\text{q2})^2}$$

$$\begin{aligned}&\frac{\left(m^2+p^2\right)^3}{8\left(\text{q1}^2-m^2\right).\left(\text{q2}^2-m^2\right).(\text{q1}-p)^2.(\text{q1}-\text{q2})^2.(\text{q2}-p)^2}\\&-\frac{\left(m^2+p^2\right)^2}{4\left(\text{q1}^2-m^2\right).\left(\text{q2}^2-m^2\right).(\text{q1}-p)^2.(\text{q1}-\text{q2})^2}\\&-\frac{m^2+p^2}{4\left(\text{q1}^2-m^2\right).(\text{q1}-\text{q2})^2.(\text{q2}-p)^2}+\frac{m^2+p^2}{8\left(\text{q1}^2-m^2\right).\left(\text{q2}^2-m^2\right).(\text{q1}-\text{q2})^2}\\&+\frac{\left(m^2+p^2\right)\left(m^2+2p^2\right)}{4\,\text{q1}^2.\text{q2}^2.\left((\text{q1}-p)^2-m^2\right).(\text{q1}-\text{q2})^2}-\frac{\left(m^2+p^2\right)(p\cdot \text{q1})}{4\,\text{q1}^2.\text{q2}^2.(\text{q1}-\text{q2})^2.\left((\text{q2}-p)^2-m^2\right)}\\&-\frac{\left(m^2+p^2\right)(p\cdot \text{q1})}{4\left(\text{q1}^2-m^2\right).\left(\text{q2}^2-m^2\right).(\text{q1}-\text{q2})^2.(\text{q2}-p)^2}-\frac{p\cdot \text{q1}}{4\left(\text{q1}^2-m^2\right).(\text{q1}-\text{q2})^2.(\text{q2}-p)^2}\\&+\frac{p\cdot \text{q1}}{4\left(\text{q1}^2-m^2\right).\left(\text{q2}^2-m^2\right).(\text{q1}-\text{q2})^2}-\frac{p\cdot \text{q1}}{4\left(\text{q2}^2-m^2\right).(\text{q1}-p)^2.(\text{q1}-\text{q2})^2}\end{aligned}$$

9.16 FCClausen

`FCClausen[x, y]` gives the Clausen function with arguments **x** and **y**.

9.16.1 See also

Overview

9.16.2 Examples

```
FCClausen[2, x]
```

$$\text{Cl}_2(x)$$

```
FCClausen[2, x] // Explicit
```

$$\frac{1}{2}i\left(\text{Li}_2\left(e^{-ix}\right)-\text{Li}_2\left(e^{ix}\right)\right)$$

```
FCClausen[2, 1.3]
```

$$\text{Cl}_2(1.3)$$

```
FCClausen[2, 1.3] // N
```

$$0.989703+0.i$$

```
FCClausen[2, Pi/2, Explicit -> True]
```

$$C$$

```
FCClausen[2, Pi/2, Explicit -> True] // StandardForm

(*Catalan*)
```

9.17 FCGramMatrix

`FCGramMatrix[{p1, p2, ...}]` creates a Gram matrix from the given list of momenta.

9.17.1 See also

Overview, FCGramDeterminant.

9.17.2 Examples

```
FCGramMatrix[{p1, p2}]
```

$$\begin{pmatrix} 2\,\mathrm{p1}^2 & 2(\mathrm{p1}\cdot\,\mathrm{p2}) \\ 2(\mathrm{p1}\cdot\,\mathrm{p2}) & 2\,\mathrm{p2}^2 \end{pmatrix}$$

```
FCGramMatrix[{p1, p2, p3}]
```

$$\begin{pmatrix} 2\,\mathrm{p1}^2 & 2(\mathrm{p1}\cdot\,\mathrm{p2}) & 2(\mathrm{p1}\cdot\,\mathrm{p3}) \\ 2(\mathrm{p1}\cdot\,\mathrm{p2}) & 2\,\mathrm{p2}^2 & 2(\mathrm{p2}\cdot\,\mathrm{p3}) \\ 2(\mathrm{p1}\cdot\,\mathrm{p3}) & 2(\mathrm{p2}\cdot\,\mathrm{p3}) & 2\,\mathrm{p3}^2 \end{pmatrix}$$

```
FCGramMatrix[{p1, p2, p3}, Head -> {CartesianPair,
CartesianMomentum},Dimension -> D - 1]

Det[%]
```

$$\begin{pmatrix} 2\,\mathrm{p1}^2 & 2(\mathrm{p1}\cdot\,\mathrm{p2}) & 2(\mathrm{p1}\cdot\,\mathrm{p3}) \\ 2(\mathrm{p1}\cdot\,\mathrm{p2}) & 2\,\mathrm{p2}^2 & 2(\mathrm{p2}\cdot\,\mathrm{p3}) \\ 2(\mathrm{p1}\cdot\,\mathrm{p3}) & 2(\mathrm{p2}\cdot\,\mathrm{p3}) & 2\,\mathrm{p3}^2 \end{pmatrix}$$

$$-8\,\mathrm{p3}^2(\mathrm{p1}\cdot\,\mathrm{p2})^2 - 8\,\mathrm{p1}^2(\mathrm{p2}\cdot\,\mathrm{p3})^2 - 8\,\mathrm{p2}^2(\mathrm{p1}\cdot\,\mathrm{p3})^2 + 8\,\mathrm{p1}^2\,\mathrm{p2}^2\,\mathrm{p3}^2 + 16(\mathrm{p1}\cdot\,\mathrm{p2})(\mathrm{p1}\cdot\,\mathrm{p3})(\mathrm{p2}\cdot\,\mathrm{p3})$$

```
FCGramDeterminant[{p1, p2, p3}, Head -> {CartesianPair, CartesianMomentum},
Dimension -> D - 1]
```

$$-8\,\mathrm{p3}^2(\mathrm{p1}\cdot\,\mathrm{p2})^2 - 8\,\mathrm{p1}^2(\mathrm{p2}\cdot\,\mathrm{p3})^2 - 8\,\mathrm{p2}^2(\mathrm{p1}\cdot\,\mathrm{p3})^2 + 8\,\mathrm{p1}^2\,\mathrm{p2}^2\,\mathrm{p3}^2 + 16(\mathrm{p1}\cdot\,\mathrm{p2})(\mathrm{p1}\cdot\,\mathrm{p3})(\mathrm{p2}\cdot\,\mathrm{p3})$$

9.18 FCGramDeterminant

`FCGramDeterminant[{p1, p2, ...}]` computes the determinant of the Gram matrix created from the given list of momenta.

9.18.1 See also

Overview, FCGramMatrix.

9.18.2 Examples

```
FCGramDeterminant[{p1, p2, p3}]
```

$$-8\,\mathrm{p3}^2(\mathrm{p1}\cdot\,\mathrm{p2})^2-8\,\mathrm{p1}^2(\mathrm{p2}\cdot\,\mathrm{p3})^2-8\,\mathrm{p2}^2(\mathrm{p1}\cdot\,\mathrm{p3})^2+8\,\mathrm{p1}^2\,\mathrm{p2}^2\,\mathrm{p3}^2+16(\mathrm{p1}\cdot\,\mathrm{p2})(\mathrm{p1}\cdot\,\mathrm{p3})(\mathrm{p2}\cdot\,\mathrm{p3})$$

```
FCGramDeterminant[{p1, p2, p3}, Head -> {CartesianPair, CartesianMomentum},
Dimension -> D - 1]
```

$$-8\,\mathrm{p3}^2(\mathrm{p1}\cdot\,\mathrm{p2})^2-8\,\mathrm{p1}^2(\mathrm{p2}\cdot\,\mathrm{p3})^2-8\,\mathrm{p2}^2(\mathrm{p1}\cdot\,\mathrm{p3})^2+8\,\mathrm{p1}^2\,\mathrm{p2}^2\,\mathrm{p3}^2+16(\mathrm{p1}\cdot\,\mathrm{p2})(\mathrm{p1}\cdot\,\mathrm{p3})(\mathrm{p2}\cdot\,\mathrm{p3})$$

9.19 FCDiffEqChangeVariables

`FCDiffEqChangeVariables[mat, x, y, rule, yOfX]` applies a variable transformation from **`x`** to **`y`**described by **`rule`**, where **`yOfX`** denotes $y(x)$. Here **`mat`** is a matrix in the context of differential equations, i.e. it can be either the matrix $\mathcal{A}$ or $\mathcal{B}$ from the pre-canonical $F' = \mathcal{A}F$ or canonical $G' = \varepsilon\mathcal{B}G$ form, or the transformation matrix $\mathcal{T}$ with $F = \mathcal{T}G$.

By default, the transformation also includes the prefactor $1/f'(y)$. This is correct for $\mathcal{A}$ or $\mathcal{B}$ but not for $\mathcal{T}$ matrices. The inclusion of the prefactor can be disabled by setting the option **`Prefactor`** to **`False`**.

9.19.1 See also

Overview.

9.19.2 Examples

```
mat = {{(-2*(-1 + eps))/x, 0, 0, 0}, {0, (1 - eps)/x, 0, 0}, {0, (-2*(-1 +
eps))/(x*(-1 + 4*x)),
    (-2*(-1 + 2*eps))/(-1 + 4*x), 0},  {(-2*(-1 + eps))/(x*(-1 + 4*x)), 0,
0,
    (-1 + eps + 6*x - 8*eps*x)/(x*(-1 + 4*x))}}
```

$$\begin{pmatrix} -\frac{2(\mathrm{eps}-1)}{x} & 0 & 0 & 0 \\ 0 & \frac{1-\mathrm{eps}}{x} & 0 & 0 \\ 0 & -\frac{2(\mathrm{eps}-1)}{x(4x-1)} & -\frac{2(2\,\mathrm{eps}-1)}{4x-1} & 0 \\ -\frac{2(\mathrm{eps}-1)}{x(4x-1)} & 0 & 0 & \frac{-8\,\mathrm{eps}x+\mathrm{eps}+6x-1}{x(4x-1)} \end{pmatrix}$$

```
matNew = FCDiffEqChangeVariables[mat, x, y, x -> (1 - y^2)/4, Sqrt[1 -
4*x], Assumptions -> {y > 0}]
```

$$\begin{pmatrix} -\frac{4(\text{eps}-1)y}{y^2-1} & 0 & 0 & 0 \\ 0 & -\frac{2(\text{eps}-1)y}{y^2-1} & 0 & 0 \\ 0 & \frac{4-4\,\text{eps}}{y-y^3} & \frac{1-2\,\text{eps}}{y} & 0 \\ \frac{4-4\,\text{eps}}{y-y^3} & 0 & 0 & \frac{4\,\text{eps}y^2-2\,\text{eps}-3y^2+1}{y-y^3} \end{pmatrix}$$

Setting the option **Reverse** to **True** allows to undo the transformation.

```
matCheck = FCDiffEqChangeVariables[matNew, x, y, x -> (1 - y^2)/4, Sqrt[1 -
4*x], Reverse -> True]
```

$$\begin{pmatrix} \frac{2-2\,\text{eps}}{x} & 0 & 0 & 0 \\ 0 & \frac{1-\text{eps}}{x} & 0 & 0 \\ 0 & -\frac{2-2\,\text{eps}}{x-4x^2} & \frac{2-4\,\text{eps}}{4x-1} & 0 \\ -\frac{2-2\,\text{eps}}{x-4x^2} & 0 & 0 & \frac{8\,\text{eps}x-\text{eps}-6x+1}{x-4x^2} \end{pmatrix}$$

```
Simplify[matCheck - mat] // Flatten // Union
```

$$\{0\}$$

```
FCDiffEqChangeVariables[mat, x, y, x -> (1 - y^2)/4, Sqrt[1 - 4*x],
Assumptions -> {y > 0},
  Prefactor -> False]
```

$$\begin{pmatrix} \frac{8(\text{eps}-1)}{y^2-1} & 0 & 0 & 0 \\ 0 & \frac{4(\text{eps}-1)}{y^2-1} & 0 & 0 \\ 0 & -\frac{8(\text{eps}-1)}{y^2(y^2-1)} & \frac{4\,\text{eps}-2}{y^2} & 0 \\ -\frac{8(\text{eps}-1)}{y^2(y^2-1)} & 0 & 0 & \frac{\text{eps}(8y^2-4)-6y^2+2}{y^2(y^2-1)} \end{pmatrix}$$

9.20 FCHideEpsilon

FCHideEpsilon[expr] substitutes **1/Epsilon - EulerGamma + Log[4 Pi]** with **SMP["Delta"]**.

9.20.1 See also

Overview, FCShowEpsilon.

9.20.2 Examples

```
1/Epsilon + Log[4 Pi] - EulerGamma

FCHideEpsilon[%]
```

$$\frac{1}{\varepsilon} - \gamma + \log(4\pi)$$

$$\Delta$$

```
1/EpsilonUV + Log[4 Pi] - EulerGamma

FCHideEpsilon[%]
```

$$\frac{1}{\varepsilon_{\text{UV}}} - \gamma + \log(4\pi)$$

$$\Delta_{\text{UV}}$$

```
1/EpsilonIR + Log[4 Pi] - EulerGamma

FCHideEpsilon[%]
```

$$\frac{1}{\varepsilon_{\text{IR}}} - \gamma + \log(4\pi)$$

$$\Delta_{\text{IR}}$$

9.21 FCShowEpsilon

FCShowEpsilon[expr] substitutes **SMP["Delta"]** with **1/Epsilon - EulerGamma + Log[4 Pi]**.

9.21.1 See also

Overview, FCHideEpsilon.

9.21.2 Examples

```
SMP["Delta"]

FCShowEpsilon[%]
```

$$\Delta$$

$$\frac{1}{\varepsilon} - \gamma + \log(4\pi)$$

```
SMP["Delta_UV"]

FCShowEpsilon[%]
```

$$\Delta_{\text{UV}}$$

$$\frac{1}{\varepsilon_{\text{UV}}} - \gamma + \log(4\pi)$$

```
SMP["Delta_IR"]

FCShowEpsilon[%]
```

$$\Delta_{\text{IR}}$$

$$\frac{1}{\varepsilon_{\text{IR}}} - \gamma + \log(4\pi)$$

9.22 FCIntegral

FCIntegral is the head of integrals in a setting of the option **IntegralTable** of **FeynAmpDenominatorSimplify**. Currently implemented only for 2-loop integrals.

9.22.1 See also

Overview, IntegralTable, FeynAmpDenominatorSimplify.

9.22.2 Examples

9.23 FCFeynmanFindDivergences

`FCFeynmanFindDivergences[exp, vars]` identifies UV and IR divergences of the given Feynman parametric integral that arise when different parametric variables approach zero or infinity.

This function employs the analytic regularization algorithm introduced by Erik Panzer in 1403.3385, 1401.4361 and 1506.07243. Its current implementation is very much based on the code of the **`findDivergences`** routine from the Maple package HyperInt by Erik Panzer.

The function returns a list of lists of the form **`{{{x[i], x[j], ...}, {x[k], x[l], ...}, sdd}, ...}`**, where **`{x[i], x[j], ...}`** need to approach zero, while **`{x[k], x[l], ...}`** must tend towards infinity to generate the superficial degree of divergence **`sdd`**.

It is important to apply the function directly to the Feynman parametric integrand obtained e.g. from **`FCFeynmanParametrize`**. If the integrand has already been modified using variable transformations or the Cheng-Wu theorem, the algorithm may not work properly.

Furthermore, divergences that arise inside the integration domain cannot be identified using this method.

The identified divergences can be regularized using the function **`FCFeynmanRegularizeDivergence`**.

9.23.1 See also

Overview, FCFeynmanParametrize, FCFeynmanProjectivize, FCFeynmanRegularizeDivergence.

9.23.2 Examples

```
int = SFAD[l, k + l, {{k, -2 k . q}}]

fpar = FCFeynmanParametrize[int, {k, l}, Names -> x, FCReplaceD -> {D -> 4
- 2 Epsilon}]
```

$$\frac{1}{(l^2+i\eta).((k+l)^2+i\eta).(k^2-2(k\cdot q)+i\eta)}$$

$$\left\{(x(1)x(2)+x(3)x(2)+x(1)x(3))^{3\varepsilon-3}\left(q^2x(1)^2(x(2)+x(3))\right)^{1-2\varepsilon},-\Gamma(2\varepsilon-1),\{x(1),x(2),x(3)\}\right\}$$

This Feynman parametric integral contains logarithmic divergences for $x_1 \to \infty$ and $x_{2,3} \to 0$

```
FCFeynmanFindDivergences[fpar[[1]], x]
```

$$\begin{pmatrix} \{\{\},\{x(1)\}\} & \varepsilon \\ \{\{x(2),x(3)\},\{\}\} & \varepsilon \end{pmatrix}$$

9.24 FCFeynmanRegularizeDivergence

FCFeynmanRegularizeDivergence[exp, div] regularizes the divergence **div** in the Feynman parametric integral **exp**. Provided that all divergences have been regularized in this fashion, upon expanding the integrand around $\varepsilon = 0$ one can safely integrate in the Feynman parameters.

Notice that **div** can be also a list made of divergences found by **FCFeynmanFindDivergences**.

This function uses the method of analytic regularization introduced by Erik Panzer in 1403.3385, 1401.4361 and 1506.07243.

Its current implementation is very much based on the code of the **dimregPartial** routine from the Maple package HyperInt by Erik Panzer.

Here **div** must be of the form **{{x[i], x[j], ...}, {x[k], x[l], ...}, sdd}**, where **{x[i], x[j], ...}** need to approach zero, while **{x[k], x[l], ...}** must tend towards infinity to generate the superficial degree of divergence **sdd**.

9.24.1 See also

Overview, FCFeynmanParametrize, FCFeynmanProjectivize, FCFeynmanFindDivergences.

9.24.2 Examples

```
int = SFAD[l, k + l, {{k, -2 k . q}}]
fpar = FCFeynmanParametrize[int, {k, l}, Names -> x, FCReplaceD -> {D -> 4
- 2 Epsilon}]
```

$$\frac{1}{(l^2+i\eta).((k+l)^2+i\eta).(k^2-2(k\cdot q)+i\eta)}$$

$$\left\{(x(1)x(2)+x(3)x(2)+x(1)x(3))^{3\varepsilon-3}\left(q^2x(1)^2(x(2)+x(3))\right)^{1-2\varepsilon},-\Gamma(2\varepsilon-1),\{x(1),x(2),x(3)\}\right\}$$

This Feynman parametric integral integrand contains logarithmic divergences for $x_1 \to \infty$ and $x_{2,3} \to 0$

```
divs = FCFeynmanFindDivergences[fpar[[1]], x]
```

$$\begin{pmatrix} \{\{\},\{x(1)\}\} & \varepsilon \\ \{\{x(2),x(3)\},\{\}\} & \varepsilon \end{pmatrix}$$

Regularizing the first divergence we obtain

```
intReg = FCFeynmanRegularizeDivergence[fpar[[1]], divs[[1]]]
```

$$-\frac{3(\varepsilon-1)q^2x(1)^2x(2)x(3)(x(2)+x(3))(x(1)x(2)+x(3)x(2)+x(1)x(3))^{3\varepsilon-4}\left(q^2x(1)^2(x(2)+x(3))\right)^{-2\varepsilon}}{\varepsilon}$$

It turns out that there are no further divergences left

```
FCFeynmanFindDivergences[intReg, x]
```

$$\{\}$$

Now one can expand the integrand in **Epsilon** and perform the integration in Feynman parameters order by order in **Epsilon**

```
Series[intReg, {Epsilon, 0, 0}] // Normal
```

$$\frac{3q^2x(1)^2x(2)x(3)(x(2)+x(3))}{\varepsilon(x(1)x(2)+x(3)x(2)+x(1)x(3))^4}$$
$$-\frac{3q^2x(1)^2x(2)x(3)(x(2)+x(3))\left(2\log\left(q^2x(1)^2(x(2)+x(3))\right)-3\log(x(1)x(2)+x(3)x(2)+x(1)x(3))+1\right)}{(x(1)x(2)+x(3)x(2)+x(1)x(3))^4}$$

Here is an example of regularizing two divergences at a time

```
FCFeynmanRegularizeDivergence[(y[1]*(y[1] + y[2] + y[3])^(2*ep)*(y[1]^2 -
4*y[2]*y[3])^(-2 -
        ep))/(x[1] + x[2])^2, {{{{y[2]}, {y[3]}}, -2*ep}, {{{y[3]},
{y[2]}}, -2*ep}}]
```

$$\frac{1}{2\,\mathrm{ep}(x(1)+x(2))^2}y(1)(y(1)+y(2)+y(3))^{2(\mathrm{ep}-1)}\left(y(1)^2-4y(2)y(3)\right)^{-\mathrm{ep}-2}\left(2\,\mathrm{ep}y(1)^2\right.$$
$$\left.+4\,\mathrm{ep}y(2)y(1)+4\,\mathrm{ep}y(3)y(1)+8\,\mathrm{ep}y(2)y(3)-y(2)y(1)-y(3)y(1)-4y(2)y(3)\right)$$

9.25 FCFeynmanParameterJoin

FCFeynmanParameterJoin[{{{prop1, prop2, x}, prop3, y}, ...}, {p1, p2, ...}] joins all propagators in **int** using Feynman parameters but does not integrate over the loop momenta p_i. The function returns **{fpInt, pref, vars}**, where **fpInt** is the piece of the integral that contains a single **GFAD**-type propagator and **pref** is the part containing the **res**. The introduced Feynman parameters are listed in vars. The overall Dirac delta is omitted.

Notice that each inner list must contain exactly thee elements, the first two being propagators (or products of propagators) and the last one denoting the head of Feynman parameter variables used to join those propagators. For example, **{FAD[{p1, m1}], FAD[{p2, m2}], x}**, but also **{FAD[{p1, m1}], FAD[{p2, m2}]FAD[{p3, m3}], x}** or even **{FAD[{p1, m1}]FAD[{p2, m2}], FAD[{p3, m3}]FAD[{p4, m4}], x}** represent valid examples of such lists, while something like **{FAD[{p1, m1}], FAD[{p2, m2}], FAD[{p3, m3}], x}** (4 instead of 3 list elements) is invalid and will not work.

Having obtained an output of **FCFeynmanParameterJoin** (e.g. called **intT**), you should use the following syntax to pass this to **FCFeynmanParametrize**: **FCFeynmanParametrize[intT[[1]], intT[[2]], {lmom1, lmom2, ...}, Variables->intT[[3]]]**

9.25.1 See also

Overview, FCFeynmanParametrize.

9.25.2 Examples

```
testProps = {FAD[{p1, m1}], FAD[{p2, m2}], FAD[{p3, m3}], FAD[{p4, m4}]}
```

$$\left\{\frac{1}{\text{p1}^2-\text{m1}^2},\frac{1}{\text{p2}^2-\text{m2}^2},\frac{1}{\text{p3}^2-\text{m3}^2},\frac{1}{\text{p4}^2-\text{m4}^2}\right\}$$

Let us first join two propagators with each other using Feynman parameters **x[i]**

```
FCFeynmanParameterJoin[{testProps[[1]], testProps[[2]], x}, {p1, p2, p3,
p4}]
```

$$\left\{\frac{1}{\left(\left(\text{p1}^2-\text{m1}^2\right)x(1)+\left(\text{p2}^2-\text{m2}^2\right)x(2)+i\eta\right)^2},1,\{x(1),x(2)\}\right\}$$

Now we can join the resulting propagator with another propagator by introducing another set of Feynman parameters **y[i]**

```
FCFeynmanParameterJoin[{{testProps[[1]], testProps[[2]], x},
testProps[[3]], y},
  {p1, p2, p3, p4}]
```

$$\left\{\frac{1}{\left(\left(-x(1)\,\mathrm{m1}^2+\mathrm{p1}^2x(1)-\mathrm{m2}^2x(2)+\mathrm{p2}^2x(2)\right)y(1)+\left(\mathrm{p3}^2-\mathrm{m3}^2\right)y(2)+i\eta\right)^3},\right.$$
$$\left.2y(1),\{x(1),x(2),y(1),y(2)\}\right\}$$

If needed, this procedure can be nested even further

```
FCFeynmanParameterJoin[{{{testProps[[1]], testProps[[2]], x},
testProps[[3]], y},
   testProps[[4]], z}, {p1, p2, p3, p4}]
```

$$\left\{\frac{1}{\left(\left(-x(1)y(1)\,\mathrm{m1}^2+\mathrm{p1}^2x(1)y(1)-\mathrm{m2}^2x(2)y(1)+\mathrm{p2}^2x(2)y(1)-\mathrm{m3}^2y(2)+\mathrm{p3}^2y(2)\right)z(1)+\left(\mathrm{p4}^2-\mathrm{m4}^2\right)z(2)+i\eta\right)^4}\right.$$
$$\left.6y(1)z(1)^2,\{x(1),x(2),y(1),y(2),z(1),z(2)\}\right\}$$

Notice that **FCFeynmanParametrize**knows how to deal with the output produced by **FCFeynmanParameterJoin**

```
intT = FCFeynmanParameterJoin[{{SFAD[{p1, mg^2}] SFAD[{p3 - p1, mg^2}], 1,
x},
    SFAD[{{0, -2 p1 . q}}] SFAD[{{0, -2 p3 . q}}], y}, {p1, p3}]
```

$$\left\{\frac{1}{\left(\left(-x(1)\,\mathrm{mg}^2-x(2)\,\mathrm{mg}^2+\mathrm{p1}^2x(1)+\mathrm{p1}^2x(2)-2(\mathrm{p1}\cdot\ \mathrm{p3})x(2)+\mathrm{p3}^2x(2)\right)y(1)-2(\mathrm{p1}\cdot q)y(2)-2(\mathrm{p3}\cdot q)y(3)+i\eta\right)}\right.$$
$$\left.6y(1),\{x(1),x(2),y(1),y(2),y(3)\}\right\}$$

```
FCFeynmanParametrize[intT[[1]], intT[[2]], {p1, p3}, Names -> z, Indexed ->
True,
  FCReplaceD -> {D -> 4 - 2 ep}, Simplify -> True, Assumptions -> {mg > 0,
ep > 0},
  FinalSubstitutions -> {FCI@SPD[q] -> qq, mg^2 -> mg2}, Variables ->
intT[[3]]]
```

$$\left\{\frac{\left(x(1)x(2)y(1)^2\right)^{3\,\mathrm{ep}}\left(\mathrm{mg2}x(1)x(2)(x(1)+x(2))y(1)^3+\mathrm{qq}y(1)\left(x(1)y(3)^2+x(2)(y(2)+y(3))^2\right)\right)^{-2\,\mathrm{ep}}}{x(1)^2x(2)^2y(1)^3},\right.$$
$$\left.\Gamma(2\,\mathrm{ep}),\{x(1),x(2),y(1),y(2),y(3)\}\right\}$$

9.26 FCFeynmanParametrize

FCFeynmanParametrize[int, {q1, q2, ...}] introduces Feynman parameters for the multi-loop integral int.

The function returns **{fpInt, pref, vars}**, where **fpInt** is the integrand in Feynman parameters, **pref** is the prefactor free of Feynman parameters and **vars** is the list of integration variables.

If the chosen parametrization contains a Dirac delta multiplying the integrand, it will be omitted unless the option **DiracDelta** is set to True.

By default **FCFeynmanParametrize** uses normalization that is common in multi-loop calculations, i.e. $\frac{1}{i\pi^{D/2}}$ or $\frac{1}{\pi^{D/2}}$ per loop for Minkowski or Euclidean/Cartesian integrals respectively.

If you want to have the standard $\frac{1}{(2\pi)^D}$ normalization or yet another value, please set the option **FeynmanIntegralPrefactor** accordingly. Following values are available

- "MultiLoop1" - default value explained above
- "MultiLoop2" - like the default value but with an extra $e^{\gamma_E \frac{4-D}{2}}$ per loop
- "Textbook" - $\frac{1}{(2\pi)^D}$ per loop
- "Unity" - no extra prefactor multiplying the integral measure
- "LoopTools" - overall prefactor $\frac{1}{i(\pi)^{D/2} r_\Gamma}$ with $r_\Gamma = \frac{\Gamma(3-D/2)\Gamma^2(D/2-1)}{\Gamma(D-3)}$ at 1 loop. This matches the the normalization of 1-loop integrals in LoopTools. For 2 loops and above an extra $\frac{1}{i\pi^{D/2}}$ is added per loop.

The calculation of D-dimensional Minkowski integrals and $D-1$-dimensional Cartesian integrals is straightforward.

To calculate a D-dimensional Euclidean integral (i.e. an integral defined with the Euclidean metric signature $(1,1,1,1)$ you need to write it in terms of **FVD**, **SPD**, **FAD**, **SFAD** etc. and set the option **"Euclidean"** to **True**.

The function can derive different representations of a loop integral. The choice of the representation is controlled by the option **Method**. Following representations are available

- "Feynman" - the standard Feynman representation (default value). Both tensor integrals and integrals with scalar products in the numerator are supported.
- "Lee-Pomeransky" - this representation was first introduced in 1308.6676 by Roman Lee and Andrei Pomeransky. Currently, only scalar integrals without numerators are supported.

FCFeynmanParametrize can also be employed in conjunction with **FCFeynmanParameterJoin**, where one first joins suitable propagators using auxiliary Feynman parameters and then finally integrates out loop momenta.

For a proper analysis of a loop integral one usually needs the **U** and **F** polynomials separately. Since internally **FCFeynmanParametrize** uses **FCFeynmanPrepare**, the information available from the latter is also accessible to **FCFeynmanParametrize**.

By setting the option **FCFeynmanPrepare** to **True**, the output of **FCFeynmanPrepare** will be added the the output of **FCFeynmanParametrize** as the 4th list element.

9.26.1 See also

Overview, FCFeynmanPrepare, FCFeynmanProjectivize, FCFeynmanParameterJoin, SplitSymbolicPowers.

9.26.2 Examples

Feynman representation

1-loop tadpole

```
FCFeynmanParametrize[FAD[{q, m}], {q}, Names -> x]
```

$$\left\{1, -\left(m^2\right)^{\frac{D}{2}-1} \Gamma\left(1-\frac{D}{2}\right), \{\}\right\}$$

Massless 1-loop 2-point function

```
FCFeynmanParametrize[FAD[q, q - p], {q}, Names -> x]
```

$$\left\{(x(1)+x(2))^{2-D}\left(-p^2 x(1) x(2)\right)^{\frac{D}{2}-2}, \Gamma\left(2-\frac{D}{2}\right), \{x(1), x(2)\}\right\}$$

With p^2 replaced by **pp** and **D** set to **4 - 2 Epsilon**

```
FCFeynmanParametrize[FAD[q, q - p], {q}, Names -> x, FinalSubstitutions ->
SPD[p] -> pp,
  FCReplaceD -> {D -> 4 - 2 Epsilon}]
```

$$\{(x(1)+x(2))^{2\varepsilon-2}(-\text{pp}x(1)x(2))^{-\varepsilon}, \Gamma(\varepsilon), \{x(1), x(2)\}\}$$

Standard text-book prefactor of the loop integral measure

```
FCFeynmanParametrize[FAD[q, q - p], {q}, Names -> x, FinalSubstitutions ->
SPD[p] -> pp,
  FCReplaceD -> {D -> 4 - 2 Epsilon}, FeynmanIntegralPrefactor ->
"Textbook"]
```

$$\{(x(1)+x(2))^{2\varepsilon-2}(-\text{pp}x(1)x(2))^{-\varepsilon}, i 2^{2\varepsilon-4}\pi^{\varepsilon-2}\Gamma(\varepsilon), \{x(1), x(2)\}\}$$

Same integral but with the Euclidean metric signature

```
FCFeynmanParametrize[FAD[q, q - p], {q}, Names -> x, FinalSubstitutions ->
SPD[p] -> pp,
  FCReplaceD -> {D -> 4 - 2 Epsilon}, FeynmanIntegralPrefactor ->
"Textbook", "Euclidean" -> True]
```

$$\{(x(1)+x(2))^{2\varepsilon-2}(\mathrm{pp}x(1)x(2))^{-\varepsilon}, 2^{2\varepsilon-4}\pi^{\varepsilon-2}\Gamma(\varepsilon), \{x(1),x(2)\}\}$$

A tensor integral

```
FCFeynmanParametrize[FAD[{q, m}] FAD[{q - p, m2}] FVD[q, mu] FVD[q, nu],
{q},
  Names -> x, FCE -> True]
```

$$\Big\{(x(1)+x(2))^{-D}\left(m^2x(1)^2+m^2x(1)x(2)+\mathrm{m2}^2x(2)^2\right.$$
$$+\mathrm{m2}^2x(1)x(2)-p^2x(1)x(2)\Big)^{\frac{D}{2}-2}\left(x(2)^2\Gamma\left(2-\frac{D}{2}\right)p^{\mathrm{mu}}p^{\mathrm{nu}}\right.$$
$$\left.\left.-\frac{1}{2}\Gamma\left(1-\frac{D}{2}\right)g^{\mathrm{mu\,nu}}\left(m^2x(1)^2+m^2x(1)x(2)+\mathrm{m2}^2x(2)^2+\mathrm{m2}^2x(1)x(2)-p^2x(1)x(2)\right)\right),\right.$$
$$\left.1,\{x(1),x(2)\}\right\}$$

1-loop master formulas for Minkowski integrals (cf. Eq. 9.49b in Sterman's An introduction to QFT)

```
SFAD[{{k, 2 p . k}, M^2, s}]

FCFeynmanParametrize[%, {k}, Names -> x, FCE -> True,
FeynmanIntegralPrefactor -> 1,
  FCReplaceD -> {D -> n}]
```

$$(k^2+2(k\cdot p)-M^2+i\eta)^{-s}$$

$$\left\{1, \frac{i\pi^{n/2}(-1)^s\Gamma\left(s-\frac{n}{2}\right)\left(M^2+p^2\right)^{\frac{n}{2}-s}}{\Gamma(s)}, \{\}\right\}$$

```
FVD[k, \[Mu]] SFAD[{{k, 2 p . k}, M^2, s}]

FCFeynmanParametrize[%, {k}, Names -> x, FCE -> True,
FeynmanIntegralPrefactor -> 1,
  FCReplaceD -> {D -> n}]
```

$$k^\mu(k^2+2(k\cdot p)-M^2+i\eta)^{-s}$$

$$\left\{1, -\frac{i\pi^{n/2}(-1)^s p^\mu\Gamma\left(s-\frac{n}{2}\right)\left(M^2+p^2\right)^{\frac{n}{2}-s}}{\Gamma(s)}, \{\}\right\}$$

```
FVD[k, \[Mu]] FVD[k, \[Nu]] SFAD[{{k, 2 p . k}, M^2, s}]

FCFeynmanParametrize[%, {k}, Names -> x, FCE -> True,
FeynmanIntegralPrefactor -> 1,
  FCReplaceD -> {D -> n}]
```

$$k^\mu k^\nu (k^2 + 2(k \cdot p) - M^2 + i\eta)^{-s}$$

$$\left\{1, \frac{i\pi^{n/2}(-1)^s \left(M^2 + p^2\right)^{\frac{n}{2}-s} \left(p^\mu p^\nu \Gamma\left(s - \frac{n}{2}\right) - \frac{1}{2}\left(M^2 + p^2\right) g^{\mu\nu} \Gamma\left(-\frac{n}{2} + s - 1\right)\right)}{\Gamma(s)}, \{\}\right\}$$

1-loop master formulas for Euclidean integrals (cf. Eq. 9.49a in Sterman's An introduction to QFT)

```
SFAD[{{k, 2 p . k}, -M^2, s}]

FCFeynmanParametrize[%, {k}, Names -> x, FCE -> True, "Euclidean" -> True,
  FeynmanIntegralPrefactor -> I]
```

$$(k^2 + 2(k \cdot p) + M^2 + i\eta)^{-s}$$

$$\left\{1, \frac{i\pi^{D/2}\Gamma\left(s - \frac{D}{2}\right)\left(M^2 - p^2\right)^{\frac{D}{2}-s}}{\Gamma(s)}, \{\}\right\}$$

```
FVD[k, \[Mu]] SFAD[{{k, 2 p . k}, -M^2, s}]

FCFeynmanParametrize[%, {k}, Names -> x, FCE -> True,
FeynmanIntegralPrefactor -> I,
  FCReplaceD -> {D -> n}, "Euclidean" -> True]
```

$$k^\mu (k^2 + 2(k \cdot p) + M^2 + i\eta)^{-s}$$

$$\left\{1, -\frac{i\pi^{n/2} p^\mu \Gamma\left(s - \frac{n}{2}\right)\left(M^2 - p^2\right)^{\frac{n}{2}-s}}{\Gamma(s)}, \{\}\right\}$$

```
FVD[k, \[Mu]] FVD[k, \[Nu]] SFAD[{{k, 2 p . k}, -M^2, s}]

FCFeynmanParametrize[%, {k}, Names -> x, FCE -> True,
FeynmanIntegralPrefactor -> I,
  FCReplaceD -> {D -> n}, "Euclidean" -> True]
```

$$k^\mu k^\nu (k^2 + 2(k\cdot p) + M^2 + i\eta)^{-s}$$

$$\left\{1, \frac{i\pi^{n/2}\left(M^2-p^2\right)^{\frac{n}{2}-s}\left(\frac{1}{2}\left(M^2-p^2\right)g^{\mu\nu}\Gamma\left(-\frac{n}{2}+s-1\right)+p^\mu p^\nu\Gamma\left(s-\frac{n}{2}\right)\right)}{\Gamma(s)}, \{\}\right\}$$

1-loop massless box

```
FAD[p, p + q1, p + q1 + q2, p + q1 + q2 + q3]

FCFeynmanParametrize[%, {p}, Names -> x, FCReplaceD -> {D -> 4 - 2
Epsilon}]
```

$$\frac{1}{p^2.(p+\text{q1})^2.(p+\text{q1}+\text{q2})^2.(p+\text{q1}+\text{q2}+\text{q3})^2}$$

$$\Big\{(x(1)+x(2)+x(3)+x(4))^{2\varepsilon}\left(-2x(1)x(3)(\text{q1}\cdot\ \text{q2})-2x(1)x(4)(\text{q1}\cdot\ \text{q2})-2x(1)x(4)(\text{q1}\cdot\ \text{q3})\right.$$
$$-\text{q1}^2x(1)x(2)-\text{q1}^2x(1)x(3)-\text{q1}^2x(1)x(4)-2x(4)x(2)(\text{q2}\cdot\ \text{q3})-2x(1)x(4)(\text{q2}\cdot\ \text{q3})-\text{q2}^2x(3)x(2)$$
$$\left.-\text{q2}^2x(4)x(2)-\text{q2}^2x(1)x(3)-\text{q2}^2x(1)x(4)-\text{q3}^2x(4)x(2)-\text{q3}^2x(1)x(4)-\text{q3}^2x(3)x(4)\right)^{-\varepsilon-2},$$
$$\Gamma(\varepsilon+2), \{x(1),x(2),x(3),x(4)\}\Big\}$$

3-loop self-energy with two massive lines

```
SFAD[{{p1, 0}, {m^2, 1}, 1}, {{p2, 0}, {0, 1}, 1}, {{p3, 0}, {0, 1}, 1},
   {{p2 + p3, 0}, {0, 1}, 1}, {{p1 - Q, 0}, {m^2, 1}, 1}, {{p2 - Q, 0}, {0,
   1}, 1},
   {{p2 + p3 - Q, 0}, {0, 1}, 1}, {{p1 + p2 + p3 - Q, 0}, {0, 1}, 1}]

FCFeynmanParametrize[%, {p1, p2, p3}, Names -> x, FCReplaceD -> {D -> 4 - 2
Epsilon}]
```

$$\frac{1}{(\text{p1}^2 - m^2 + i\eta).(\text{p2}^2 + i\eta).(\text{p3}^2 + i\eta).((\text{p2}+\text{p3})^2 + i\eta).((\text{p1} - Q)^2 - m^2 + i\eta).((\text{p2} - Q)^2 + i\eta).((\text{p2}+\text{p3} - Q)^2 + i\eta)}$$

$$\Big\{(x(1)x(2)x(3) + x(1)x(4)x(3) + x(2)x(5)x(3) + x(4)x(5)x(3) + x(1)x(6)x(3) + x(5)x(6)x(3) + x(1)x(7)x(3)$$
$$+ x(5)x(7)x(3) + x(1)x(8)x(3) + x(2)x(8)x(3) + x(4)x(8)x(3) + x(5)x(8)x(3) + x(6)x(8)x(3) + x(7)x(8)x(3)$$
$$+ x(1)x(2)x(4) + x(2)x(4)x(5) + x(1)x(4)x(6) + x(4)x(5)x(6) + x(1)x(2)x(7) + x(2)x(5)x(7) + x(1)x(6)x(7)$$
$$+ x(5)x(6)x(7) + x(1)x(2)x(8) + x(2)x(4)x(8) + x(2)x(5)x(8) + x(1)x(6)x(8) + x(4)x(6)x(8) + x(5)x(6)x(8)$$
$$+ x(2)x(7)x(8) + x(6)x(7)x(8))^{4\varepsilon}\,(x(2)x(3)x(5)^2m^2 + x(2)x(4)x(5)^2m^2 + x(3)x(4)x(5)^2m^2 + x(1)^2x(2)x(3)m^2$$
$$+ x(1)^2x(2)x(4)m^2 + x(1)^2x(3)x(4)m^2 + 2x(1)x(2)x(3)x(5)m^2 + 2x(1)x(2)x(4)x(5)m^2 + 2x(1)x(3)x(4)x(5)m^2$$
$$+ x(3)x(5)^2x(6)m^2 + x(4)x(5)^2x(6)m^2 + x(1)^2x(3)x(6)m^2 + x(1)^2x(4)x(6)m^2 + 2x(1)x(3)x(5)x(6)m^2$$
$$+ 2x(1)x(4)x(5)x(6)m^2 + x(2)x(5)^2x(7)m^2 + x(3)x(5)^2x(7)m^2 + x(1)^2x(2)x(7)m^2 + x(1)^2x(3)x(7)m^2$$
$$+ 2x(1)x(2)x(5)x(7)m^2 + 2x(1)x(3)x(5)x(7)m^2 + x(1)^2x(6)x(7)m^2 + x(5)^2x(6)x(7)m^2 + 2x(1)x(5)x(6)x(7)m^2$$
$$+ x(2)x(5)^2x(8)m^2 + x(3)x(5)^2x(8)m^2 + x(1)^2x(2)x(8)m^2 + x(1)^2x(3)x(8)m^2 + x(1)x(2)x(3)x(8)m^2$$
$$+ x(1)x(2)x(4)x(8)m^2 + x(1)x(3)x(4)x(8)m^2 + 2x(1)x(2)x(5)x(8)m^2 + 2x(1)x(3)x(5)x(8)m^2$$
$$+x(2)x(3)x(5)x(8)m^2+x(2)x(4)x(5)x(8)m^2+x(3)x(4)x(5)x(8)m^2+x(1)^2x(6)x(8)m^2+x(5)^2x(6)x(8)m^2+x(1)x(3)x(6)x(8$$
$$+x(1)x(4)x(6)x(8)m^2+2x(1)x(5)x(6)x(8)m^2+x(3)x(5)x(6)x(8)m^2+x(4)x(5)x(6)x(8)m^2+x(1)x(2)x(7)x(8)m^2+x(1)x(3)$$
$$+x(2)x(5)x(7)x(8)m^2+x(3)x(5)x(7)x(8)m^2+x(1)x(6)x(7)x(8)m^2+x(5)x(6)x(7)x(8)m^2-Q^2x(1)x(2)x(3)x(5)-Q^2x(1)x($$
$$-Q^2x(1)x(3)x(4)x(5)-Q^2x(1)x(2)x(3)x(6)-Q^2x(1)x(2)x(4)x(6)-Q^2x(1)x(3)x(4)x(6)-Q^2x(1)x(3)x(5)x(6)-Q^2x(2)x(3$$
$$-Q^2x(1)x(4)x(5)x(6)-Q^2x(2)x(4)x(5)x(6)-Q^2x(3)x(4)x(5)x(6)-Q^2x(1)x(2)x(3)x(7)-Q^2x(1)x(2)x(4)x(7)-Q^2x(1)x(3$$
$$-Q^2x(1)x(2)x(5)x(7)-Q^2x(1)x(3)x(5)x(7)-Q^2x(2)x(3)x(5)x(7)-Q^2x(2)x(4)x(5)x(7)-Q^2x(3)x(4)x(5)x(7)-Q^2x(1)x(2$$
$$-Q^2x(1)x(4)x(6)x(7)-Q^2x(1)x(5)x(6)x(7)-Q^2x(2)x(5)x(6)x(7)-Q^2x(4)x(5)x(6)x(7)-Q^2x(1)x(2)x(3)x(8)-Q^2x(1)x(2$$
$$-Q^2x(1)x(3)x(4)x(8)-Q^2x(1)x(2)x(5)x(8)-Q^2x(1)x(3)x(5)x(8)-Q^2x(1)x(2)x(6)x(8)-Q^2x(2)x(3)x(6)x(8)-Q^2x(1)x(4$$
$$-Q^2x(2)x(4)x(6)x(8)-Q^2x(3)x(4)x(6)x(8)-Q^2x(1)x(5)x(6)x(8)-Q^2x(2)x(5)x(6)x(8)-Q^2x(3)x(5)x(6)x(8)-Q^2x(2)x(3$$
$$-Q^2x(2)x(4)x(7)x(8)-Q^2x(3)x(4)x(7)x(8)-Q^2x(2)x(5)x(7)x(8)-Q^2x(3)x(5)x(7)x(8)-Q^2x(2)x(6)x(7)x(8)-Q^2x(4)x(6$$
$$-Q^2x(5)x(6)x(7)x(8))^{-3\varepsilon-2}, \Gamma(3\varepsilon+2), \{x(1),x(2),x(3),x(4),x(5),x(6),x(7),x(8)\}\Big\}$$

An example of using **FCFeynmanParametrize** together with **FCFeynmanParameterJoin**

```
props = {SFAD[{p1, m^2}], SFAD[{p3, m^2}], SFAD[{{0, 2 p1 . n}}],
   SFAD[{{0, 2 (p1 + p3) . n}}]}
```

$$\left\{\frac{1}{(\text{p1}^2 - m^2 + i\eta)}, \frac{1}{(\text{p3}^2 - m^2 + i\eta)}, \frac{1}{(2(n\cdot\text{p1}) + i\eta)}, \frac{1}{(2(n\cdot(\text{p1}+\text{p3})) + i\eta)}\right\}$$

```
intT = FCFeynmanParameterJoin[{{props[[1]] props[[2]], 1, x},
    props[[3]] props[[4]], y}, {p1, p3}]
```

$$\left\{\frac{1}{\left(\left(-x(1)m^2 - x(2)m^2 + \text{p1}^2x(1) + \text{p3}^2x(2)\right)y(1) + 2(n\cdot\text{p1})y(2) + (2(n\cdot\text{p1}) + 2(n\cdot\text{p3}))y(3) + i\eta\right)^4},\right.$$
$$\left. 6y(1), \{x(1),x(2),y(1),y(2),y(3)\}\right\}$$

Here the Feynman parameter variables x_i and y_i are independent from each other, i.e. we have $\delta(1-x_1-x_2-x_3)\times\delta(1-y_1-y_2-y_3)$. This gives us much more freedom when exploiting the Cheng-Wu theorem.

```
FCFeynmanParametrize[intT[[1]], intT[[2]], {p1, p3}, Indexed -> True,
  FCReplaceD -> {D -> 4 - 2 ep}, FinalSubstitutions -> {SPD[n] -> 1, m ->
1}, Variables -> intT[[3]]]
```

$$\left\{y(1)\left(x(1)x(2)y(1)^2\right)^{3\,\text{ep}-2}\left(y(1)\left(x(1)x(2)^2y(1)^2+x(1)^2x(2)y(1)^2+x(2)y(2)^2+x(1)y(3)^2+x(2)y(3)^2+2x(2)y(2)y(3)\right)\right)^{-\Gamma(2\,\text{ep})},\{x(1),x(2),y(1),y(2),y(3)\}\right\}$$

In the case that we need **U** and **F** polynomials in addition to the normal output (e.g. for HyperInt)

```
(SFAD[{{0, 2*k1 . n}}]*SFAD[{{0, 2*k2 . n}}]*SFAD[{k1, m^2}]*
    SFAD[{k2, m^2}]*SFAD[{k1 - k2, m^2}])

out = FCFeynmanParametrize[%, {k1, k2}, Names -> x, FCReplaceD -> {D -> 4 -
2 Epsilon},
   FCFeynmanPrepare -> True]
```

$$\frac{1}{(\text{k1}^2-m^2+i\eta)(\text{k2}^2-m^2+i\eta)((\text{k1}-\text{k2})^2-m^2+i\eta)(2(\text{k1}\cdot n)+i\eta)(2(\text{k2}\cdot n)+i\eta)}$$

$$\Bigg\{ (x(3)x(4)+x(5)x(4)+x(3)x(5))^{3\varepsilon-1}\left(m^2x(3)x(4)^2+m^2x(3)x(5)^2+m^2x(4)x(5)^2\right.$$
$$+m^2x(3)^2x(4)+m^2x(3)^2x(5)+m^2x(4)^2x(5)+3m^2x(3)x(4)x(5)+n^2x(2)^2x(3)$$
$$\left.+n^2x(1)^2x(4)+n^2x(2)^2x(4)+2n^2x(1)x(2)x(4)+n^2x(1)^2x(5)\right)^{-2\varepsilon-1},$$
$$-\Gamma(2\varepsilon+1),\{x(1),x(2),x(3),x(4),x(5)\},\Bigg\{ x(3)x(4)+x(5)x(4)+x(3)x(5),$$
$$m^2x(3)x(4)^2+m^2x(3)x(5)^2+m^2x(4)x(5)^2+m^2x(3)^2x(4)+m^2x(3)^2x(5)+m^2x(4)^2x(5)$$
$$+3m^2x(3)x(4)x(5)+n^2x(2)^2x(3)+n^2x(1)^2x(4)+n^2x(2)^2x(4)+2n^2x(1)x(2)x(4)+n^2x(1)^2x(5),$$
$$\begin{pmatrix} x(1) & \frac{1}{(2(\mathrm{k1}\cdot n)+i\eta)} & 1\\ x(2) & \frac{1}{(2(\mathrm{k2}\cdot n)+i\eta)} & 1\\ x(3) & \frac{1}{(\mathrm{k1}^2-m^2+i\eta)} & 1\\ x(4) & \frac{1}{((\mathrm{k1}-\mathrm{k2})^2-m^2+i\eta)} & 1\\ x(5) & \frac{1}{(\mathrm{k2}^2-m^2+i\eta)} & 1 \end{pmatrix},\begin{pmatrix} x(3)+x(4) & -x(4)\\ -x(4) & x(4)+x(5) \end{pmatrix},$$
$$\left\{x(1)\left(-n^{\mathrm{FCGV(mu)}}\right),x(2)\left(-n^{\mathrm{FCGV(mu)}}\right)\right\},-m^2(x(3)+x(4)+x(5)),1,0\Bigg\}\Bigg\}$$

From this output we can easily extract the integrand, its x_i-independent prefactor and the two Symanzik polynomials

```
{integrand, pref} = out[[1 ;; 2]]

{uPoly, fPoly} = out[[4]][[1 ;; 2]]
```

$$\Big\{(x(3)x(4)+x(5)x(4)+x(3)x(5))^{3\varepsilon-1}\left(m^2x(3)x(4)^2+m^2x(3)x(5)^2+m^2x(4)x(5)^2\right.$$
$$+m^2x(3)^2x(4)+m^2x(3)^2x(5)+m^2x(4)^2x(5)+3m^2x(3)x(4)x(5)+n^2x(2)^2x(3)$$
$$\left.+n^2x(1)^2x(4)+n^2x(2)^2x(4)+2n^2x(1)x(2)x(4)+n^2x(1)^2x(5)\right)^{-2\varepsilon-1},-\Gamma(2\varepsilon+1)\Big\}$$

$$\{x(3)x(4)+x(5)x(4)+x(3)x(5),$$
$$m^2x(3)x(4)^2+m^2x(3)x(5)^2+m^2x(4)x(5)^2+m^2x(3)^2x(4)+m^2x(3)^2x(5)+m^2x(4)^2x(5)$$
$$+3m^2x(3)x(4)x(5)+n^2x(2)^2x(3)+n^2x(1)^2x(4)+n^2x(2)^2x(4)+2n^2x(1)x(2)x(4)+n^2x(1)^2x(5)\}$$

Symbolic propagator powers are fully supported

```
SFAD[{I k, 0, -1/2 + ep}, {I (k + p), 0, 1}, EtaSign -> -1]

v1 = FCFeynmanParametrize[%, {k}, Names -> x, FCReplaceD -> {D -> 4 - 2
ep},
   FinalSubstitutions -> {SPD[p] -> 1}]
```

$$\frac{1}{(-k^2 - i\eta)^{\text{ep}-\frac{1}{2}}.(-(k+p)^2 - i\eta)}$$

$$\left\{(-x(1) - x(2))^{3\,\text{ep}-\frac{7}{2}} x(2)^{\text{ep}-\frac{3}{2}} (-x(1)x(2))^{\frac{3}{2}-2\,\text{ep}}, \frac{(-1)^{\text{ep}+\frac{1}{2}}\Gamma\left(2\,\text{ep}-\frac{3}{2}\right)}{\Gamma\left(\text{ep}-\frac{1}{2}\right)}, \{x(1), x(2)\}\right\}$$

An alternative representation for symbolic powers can be obtained using the option **SplitSymbolicPowers**

```
SFAD[{I k, 0, -1/2 + ep}, {I (k + p), 0, 1}, EtaSign -> -1]

v2 = FCFeynmanParametrize[%, {k}, Names -> x, FCReplaceD -> {D -> 4 - 2
ep},
   FinalSubstitutions -> {SPD[p] -> 1}, SplitSymbolicPowers -> True]
```

$$\frac{1}{(-k^2 - i\eta)^{\text{ep}-\frac{1}{2}}.(-(k+p)^2 - i\eta)}$$

$$\left\{x(2)^{\text{ep}-\frac{1}{2}} \left(\left(\frac{1}{2}(1-2\,\text{ep}) + \frac{1}{2}(4-2\,\text{ep}) - 1\right) x(1)(-x(1)-x(2))^{3\,\text{ep}-\frac{7}{2}} (-x(1)x(2))^{\frac{1}{2}-2\,\text{ep}} \right.\right.$$
$$\left.\left. + \left(2\,\text{ep} + \frac{1}{2}(2\,\text{ep}-1) - 3\right)(-x(1)-x(2))^{3\,\text{ep}-\frac{9}{2}} (-x(1)x(2))^{\frac{3}{2}-2\,\text{ep}}\right),$$
$$\left.\frac{(-1)^{\text{ep}+\frac{1}{2}}\Gamma\left(2\,\text{ep}-\frac{3}{2}\right)}{\Gamma\left(\text{ep}+\frac{1}{2}\right)}, \{x(1), x(2)\}\right\}$$

Even though the parametric integrals evaluate to different values, the product of the integral and its prefactor remains the same

```
Integrate[Normal[Series[v1[[1]] /. x[1] -> 1, {ep, 0, 0}]] /. x[1] -> 1,
{x[2], 0, Infinity}]

Normal@Series[v1[[2]] %, {ep, 0, 0}]
```

$$\frac{2}{5}$$

$$-\frac{4i}{15}$$

```
Integrate[Normal[Series[v2[[1]] /. x[1] -> 1, {ep, 0, 0}]] /. x[1] -> 1,
{x[2], 0, Infinity}]

Normal@Series[v2[[2]] %, {ep, 0, 0}]
```

$$-\frac{1}{5}$$

$$-\frac{4i}{15}$$

Calculate the simplest divergent triangle integral from QCDLoop

```
FCClearScalarProducts[];
SPD[r] = 0;
SPD[s] = 0;
SPD[r, s] = -1/2;
int = FAD[{q, 0}, {q - r, 0}, {q - s, 0}]
```

$$\frac{1}{q^2.(q-r)^2.(q-s)^2}$$

```
ToPaVe[int, q]
```

$$i\pi^2\, \mathrm{C}_0(0,0,1,0,0,0)$$

```
fp = FCFeynmanParametrize[int, {q}, Names -> x, FCReplaceD -> {D -> 4 - 2
ep}, FeynmanIntegralPrefactor -> "LoopTools"]
```

$$\left\{(-x(2)x(3))^{-\mathrm{ep}-1}(x(1)+x(2)+x(3))^{2\,\mathrm{ep}-1}, -\frac{\Gamma(1-2\,\mathrm{ep})}{\Gamma(1-\mathrm{ep})^2}, \{x(1),x(2),x(3)\}\right\}$$

```
intRaw = Integrate[fp[[1]] /. x[2] -> 1, {x[1], 0, Infinity}, Assumptions
-> {ep < 0, x[3] >= 0}]
```

$$-\frac{(-x(3))^{-\text{ep}-1}(x(3)+1)^{2\,\text{ep}}}{2\,\text{ep}}$$

Reintroduce the correct $i\eta$-prescription to get the imaginary part right

```
intRes = Integrate[intRaw, {x[3], 0, Infinity}, Assumptions -> {ep < 0}] /.
(-1)^(-ep) -> (-1 - I eta)^(-ep)
```

$$\frac{(-1-i\,\text{eta})^{-\text{ep}}\Gamma(-\text{ep})^2}{2\,\text{ep}\Gamma(-2\,\text{ep})}$$

```
res = (Series[fp[[2]] intRes, {ep, 0, 0}] // Normal) /. Log[-1 - I eta] ->
Log[1] - I Pi
```

$$\frac{1}{\text{ep}^2}+\frac{i\pi}{\text{ep}}-\frac{\pi^2}{2}$$

Compare to the known result

```
resLit = Series[ScaleMu^(2 ep)/ep^2 1/pp^2 (-pp - I eta)^(-ep), {ep, 0, 0}]
/. Log[-pp - I eta] -> Log[pp] - I Pi // Normal
```

$$\frac{1}{\text{ep}^2\,\text{pp}^2}+\frac{2\log(\mu)-\log(\text{pp})+i\pi}{\text{ep}\,\text{pp}^2}+\frac{4\log^2(\mu)-4\log(\mu)(\log(\text{pp})-i\pi)+(\log(\text{pp})-i\pi)^2}{2\,\text{pp}^2}$$

```
(res - resLit) /. pp | ScaleMu -> 1
```

$$0$$

Lee-Pomeransky representation

1-loop tadpole

```
FCFeynmanParametrize[FAD[{q, m}], {q}, Names -> x, Method ->
"Lee-Pomeransky"]
```

$$\left\{\left(m^2x(1)^2+x(1)\right)^{-D/2},-\frac{\Gamma\left(\frac{D}{2}\right)}{\Gamma(D-1)},\{x(1)\}\right\}$$

Massless 1-loop 2-point function

```
FCFeynmanParametrize[FAD[q, q - p], {q}, Names -> x, Method ->
"Lee-Pomeransky"]
```

$$\left\{\left(-p^2x(2)x(1)+x(1)+x(2)\right)^{-D/2},\frac{\Gamma\left(\frac{D}{2}\right)}{\Gamma(D-2)},\{x(1),x(2)\}\right\}$$

2-loop self-energy with 3 massive lines and two eikonal propagators

```
FCFeynmanParametrize[{SFAD[{ p1, m^2}], SFAD[{ p3, m^2}],
   SFAD[{(p3 - p1), m^2}], SFAD[{{0, 2 p1 . n}}], SFAD[{{0, 2 p3 . n}}]},
{p1, p3},
  Names -> x, Method -> "Lee-Pomeransky", FCReplaceD -> {D -> 4 - 2 ep},
  FinalSubstitutions -> {SPD[n] -> 1, m -> 1}]
```

$$\left\{\left(x(4)x(1)^2+x(5)x(1)^2+2x(2)x(5)x(1)+x(3)x(4)^2+x(3)x(5)^2+x(4)x(5)^2+x(2)^2x(3)+x(3)^2x(4)\right.\right.$$
$$\left.+x(3)x(4)+x(2)^2x(5)+x(3)^2x(5)+x(4)^2x(5)+x(3)x(5)+3x(3)x(4)x(5)+x(4)x(5)\right)^{\text{ep}-2},$$
$$\left.-\frac{\Gamma(2-\text{ep})}{\Gamma(1-3\,\text{ep})},\{x(1),x(2),x(3),x(4),x(5)\}\right\}$$

9.27 FCFeynmanPrepare

`FCFeynmanPrepare[int, {q1, q2, ...}]` is an auxiliary function that returns all necessary building for writing down a Feynman parametrization of the given tensor or scalar multi-loop integral. The integral int can be Lorentzian or Cartesian.

The output of the function is a list given by **`{U, F, pows, M, Q, J, N, r}`**, where **`U`** and **`F`** are the Symanzik polynomials, with $U = detM$, while **`pows`** contains the powers of the occurring propagators. The vector **`Q`** and the function **`J`** are the usual quantities appearing in the definition of the F" polynomial.

If the integral has free indices, then **`N`** encodes its tensor structure, while **`r`** gives its tensor rank. For scalar integrals **`N`** is always **`1`** and r is **`0`**. In **`N`** the **`F`**-polynomial is not substituted but left as **`FCGV["F"]`**.

To ensure a certain correspondence between propagators and Feynman parameters, it is also possible to enter the integral as a list of propagators, e.g. **FCFeynmanPrepare[{FAD[{q, m1}], FAD[{q-p, m2}], SPD[p, q]}, {q}]**. In this case the tensor part of the integral should be the very last element of the list.

It is also possible to invoke the function as **FCFeynmanPrepare[GLI[...], FCTopology[...]]** or **FCFeynmanPrepare[FCTopology[...]]**. Notice that in this case the value of the option **FinalSubstitutions** is ignored, as replacement rules will be extracted directly from the definition of the topology.

The definitions of **M**, **Q**, **J** and **N** follow from Eq. 4.17 in the PhD Thesis of Stefan Jahn and arXiv:1010.1667.The algorithm for deriving the UF-parametrization of a loop integral was adopted from the UF generator available in multiple codes of Alexander Smirnov, such as FIESTA (arXiv:1511.03614) and FIRE (arXiv:1901.07808). The code UF.m is also mentioned in the book "Analytic Tools for Feynman Integrals" by Vladimir Smirnov, Chapter 2.3.

9.27.1 See also

Overview, FCFeynmanParametrize, FCFeynmanProjectivize, FCLoopValidTopologyQ.

9.27.2 Examples

One of the simplest examples is the 1-loop tadpole

```
FCFeynmanPrepare[FAD[{q, m1}], {q}]
```

$$\left\{\text{FCGV(x)(1)}, \text{m1}^2 (\text{FCGV(x)(1)})^2, \left(\begin{array}{ccc} \text{FCGV(x)(1)} & \frac{1}{q^2-\text{m1}^2} & 1 \end{array} \right), \left(\begin{array}{c} \text{FCGV(x)(1)} \end{array} \right), \{0\}, -\text{m1}^2\ \text{FCGV(x)(1)}, 1, 0\right\}$$

Use the option **Names** to have specific symbols denoting Feynman parameters

```
FCFeynmanPrepare[FAD[{q, m1}], {q}, Names -> x]
```

$$\left\{x(1), \text{m1}^2 x(1)^2, \left(\begin{array}{ccc} x(1) & \frac{1}{q^2-\text{m1}^2} & 1 \end{array} \right), \left(\begin{array}{c} x(1) \end{array} \right), \{0\}, -\text{m1}^2 x(1), 1, 0\right\}$$

It is also possible to obtain e.g. **x1, x2, x3, ...** instead of **x[1], x[2], x[3], ...**

```
FCFeynmanPrepare[FAD[{q, m1}], {q}, Names -> x, Indexed -> False]
```

$$\left\{\text{x1}, \text{m1}^2\ \text{x1}^2, \left(\begin{array}{ccc} \text{x1} & \frac{1}{q^2-\text{m1}^2} & 1 \end{array} \right), \left(\begin{array}{c} \text{x1} \end{array} \right), \{0\}, -\text{m1}^2\ \text{x1}, 1, 0\right\}$$

To fix the correspondence between Feynman parameters and propagators, the latter should be entered as a list

```
FCFeynmanPrepare[{FAD[{q, m}], FAD[{q - p, m2}], FVD[q, \[Mu]] FVD[q,
\[Nu]] FVD[q, \[Rho]]}, {q}, Names -> x]
```

$$\left\{x(1)+x(2), m^2x(1)^2+m^2x(1)x(2)+\text{m2}^2x(2)^2+\text{m2}^2x(1)x(2)-p^2x(1)x(2),\right.$$
$$\begin{pmatrix} x(1) & \frac{1}{q^2-m^2} & 1 \\ x(2) & \frac{1}{(p-q)^2-\text{m2}^2} & 1 \end{pmatrix}, \begin{pmatrix} x(1)+x(2) \end{pmatrix}, \left\{x(2)p^{\text{FCGV(mu)}}\right\}, m^2(-x(1))-\text{m2}^2x(2)+p^2x(2),$$
$$-\frac{1}{2}x(2)\Gamma\left(1-\frac{D}{2}\right)\text{FCGV(F)}p^\mu g^{\nu\rho}-\frac{1}{2}x(2)\Gamma\left(1-\frac{D}{2}\right)\text{FCGV(F)}p^\nu g^{\mu\rho}$$
$$\left.-\frac{1}{2}x(2)\Gamma\left(1-\frac{D}{2}\right)\text{FCGV(F)}p^\rho g^{\mu\nu}+x(2)^3\Gamma\left(2-\frac{D}{2}\right)p^\mu p^\nu p^\rho, 3\right\}$$

Massless 2-loop self-energy

```
FCFeynmanPrepare[FAD[p1, p2, Q - p1 - p2, Q - p1, Q - p2], {p1, p2}, Names
-> x]
```

$$\left\{x(1)x(2)+x(3)x(2)+x(5)x(2)+x(1)x(4)+x(3)x(4)+x(1)x(5)+x(3)x(5)+x(4)x(5),\right.$$
$$-Q^2(x(1)x(2)x(3)+x(1)x(4)x(3)+x(2)x(4)x(3)+x(1)x(5)x(3)$$
$$+x(4)x(5)x(3)+x(1)x(2)x(4)+x(1)x(2)x(5)+x(2)x(4)x(5)),$$
$$\begin{pmatrix} x(1) & \frac{1}{\text{p1}^2} & 1 \\ x(2) & \frac{1}{\text{p2}^2} & 1 \\ x(3) & \frac{1}{(\text{p1}-Q)^2} & 1 \\ x(4) & \frac{1}{(\text{p2}-Q)^2} & 1 \\ x(5) & \frac{1}{(\text{p1}+\text{p2}-Q)^2} & 1 \end{pmatrix}, \begin{pmatrix} x(1)+x(3)+x(5) & x(5) \\ x(5) & x(2)+x(4)+x(5) \end{pmatrix},$$
$$\left.\left\{(x(3)+x(5))Q^{\text{FCGV(mu)}}, (x(4)+x(5))Q^{\text{FCGV(mu)}}\right\}, Q^2(x(3)+x(4)+x(5)), 1, 0\right\}$$

Factorizing integrals also work

```
FCFeynmanPrepare[FAD[{p1, m1}, {p2, m2}, Q - p1, Q - p2], {p1, p2}, Names
-> x]
```

$$\left\{(x(1)+x(3))(x(2)+x(4)), \mathrm{m1}^2 x(1)^2 x(2)+\mathrm{m1}^2 x(1)x(2)x(3)+\mathrm{m1}^2 x(1)^2 x(4)\right.$$
$$+\mathrm{m1}^2 x(1)x(3)x(4)+\mathrm{m2}^2 x(1)x(2)^2+\mathrm{m2}^2 x(2)^2 x(3)+\mathrm{m2}^2 x(1)x(2)x(4)+\mathrm{m2}^2 x(2)x(3)x(4)$$
$$-Q^2 x(1)x(2)x(3)-Q^2 x(1)x(2)x(4)-Q^2 x(1)x(3)x(4)-Q^2 x(2)x(3)x(4),$$
$$\begin{pmatrix} x(1) & \frac{1}{\mathrm{p1}^2-\mathrm{m1}^2} & 1 \\ x(2) & \frac{1}{\mathrm{p2}^2-\mathrm{m2}^2} & 1 \\ x(3) & \frac{1}{(\mathrm{p1}-Q)^2} & 1 \\ x(4) & \frac{1}{(\mathrm{p2}-Q)^2} & 1 \end{pmatrix}, \begin{pmatrix} x(1)+x(3) & 0 \\ 0 & x(2)+x(4) \end{pmatrix}, \left\{x(3)Q^{\mathrm{FCGV(mu)}},\right.$$
$$\left.\left.x(4)Q^{\mathrm{FCGV(mu)}}\right\}, \mathrm{m1}^2(-x(1))-\mathrm{m2}^2 x(2)+Q^2 x(3)+Q^2 x(4), 1, 0\right\}$$

Cartesian propagators are equally supported

```
FCFeynmanPrepare[CSPD[q, p] CFAD[{q, m}, {q - p, m2}], {q}, Names -> x]
```

$$\left\{x(1)+x(2),\right.$$
$$\frac{1}{4}\left(4mx(1)^2+4mx(2)x(1)+4\,\mathrm{m2}x(2)x(1)+4\,\mathrm{m2}x(2)^2+4p^2x(2)x(1)-p^2x(3)^2+4p^2x(2)x(3)\right),$$
$$\begin{pmatrix} x(1) & \frac{1}{(q^2+m-i\eta)} & 1 \\ x(2) & \frac{1}{((p-q)^2+\mathrm{m2}-i\eta)} & 1 \\ x(3) & p\cdot q & -1 \end{pmatrix}, \begin{pmatrix} x(1)+x(2) \end{pmatrix},$$
$$\left.\left\{\frac{1}{2}(2x(2)-x(3))p^{\mathrm{FCGV(i)}}\right\}, mx(1)+\mathrm{m2}x(2)+p^2x(2), 1, 0\right\}$$

FCFeynmanPrepare also works with **FCTopology** and **GLI** objects

```
topo1 = FCTopology["prop2Lv1", {SFAD[{p1, m1^2}], SFAD[{p2, m2^2}],
     SFAD[p1 - q], SFAD[p2 - q], SFAD[{p1 - p2, m3^2}]}, {p1, p2}, {Q}, {},
{}]

topo2 = FCTopology["prop2Lv2", {SFAD[{p1, m1^2}], SFAD[{p2, m2^2}],
    SFAD[{p1 - q, M^2}], SFAD[{p2 - q, M^2}], SFAD[p1 - p2]}, {p1, p2},
{Q}, {}, {}]
```

$$\text{FCTopology}\left(\text{prop2Lv1}, \left\{\frac{1}{(\text{p1}^2 - \text{m1}^2 + i\eta)}, \frac{1}{(\text{p2}^2 - \text{m2}^2 + i\eta)}, \frac{1}{((\text{p1} - q)^2 + i\eta)}, \frac{1}{((\text{p2} - q)^2 + i\eta)}, \frac{1}{((\text{p1} - \text{p2})^2 - \text{m3}^2 + i\eta)}\right\}, \{\text{p1}, \text{p2}\}, \{Q\}, \{\}, \{\}\right)$$

$$\text{FCTopology}\left(\text{prop2Lv2}, \left\{\frac{1}{(\text{p1}^2 - \text{m1}^2 + i\eta)}, \frac{1}{(\text{p2}^2 - \text{m2}^2 + i\eta)}, \frac{1}{((\text{p1} - q)^2 - M^2 + i\eta)}, \frac{1}{((\text{p2} - q)^2 - M^2 + i\eta)}, \frac{1}{((\text{p1} - \text{p2})^2 + i\eta)}\right\}, \{\text{p1}, \text{p2}\}, \{Q\}, \{\}, \{\}\right)$$

```
FCFeynmanPrepare[topo1, Names -> x]
```

$$\left\{\begin{array}{l} x(1)x(2)+x(3)x(2)+x(5)x(2)+x(1)x(4)+x(3)x(4)+x(1)x(5)+x(3)x(5)+x(4)x(5), \\ \\ \mathrm{m1}^2x(1)^2x(2)+\mathrm{m1}^2x(1)x(2)x(3)+\mathrm{m1}^2x(1)^2x(4)+\mathrm{m1}^2x(1)x(3)x(4)+\mathrm{m1}^2x(1)^2x(5) \\ +\mathrm{m1}^2x(1)x(2)x(5)+\mathrm{m1}^2x(1)x(3)x(5)+\mathrm{m1}^2x(1)x(4)x(5)+\mathrm{m2}^2x(1)x(2)^2 \\ +\mathrm{m2}^2x(2)^2x(3)+\mathrm{m2}^2x(1)x(2)x(4)+\mathrm{m2}^2x(2)x(3)x(4)+\mathrm{m2}^2x(2)^2x(5)+\mathrm{m2}^2x(1)x(2)x(5) \\ +\mathrm{m2}^2x(2)x(3)x(5)+\mathrm{m2}^2x(2)x(4)x(5)+\mathrm{m3}^2x(1)x(5)^2+\mathrm{m3}^2x(2)x(5)^2+\mathrm{m3}^2x(3)x(5)^2 \\ +\mathrm{m3}^2x(4)x(5)^2+\mathrm{m3}^2x(1)x(2)x(5)+\mathrm{m3}^2x(2)x(3)x(5)+\mathrm{m3}^2x(1)x(4)x(5) \\ +\mathrm{m3}^2x(3)x(4)x(5)-q^2x(1)x(2)x(3)-q^2x(1)x(2)x(4)-q^2x(1)x(3)x(4) \\ -q^2x(2)x(3)x(4)-q^2x(1)x(3)x(5)-q^2x(2)x(3)x(5)-q^2x(1)x(4)x(5)-q^2x(2)x(4)x(5), \\ \begin{pmatrix} x(1) & \frac{1}{(\mathrm{p1}^2-\mathrm{m1}^2+i\eta)} & 1 \\ x(2) & \frac{1}{(\mathrm{p2}^2-\mathrm{m2}^2+i\eta)} & 1 \\ x(3) & \frac{1}{((\mathrm{p1}-q)^2+i\eta)} & 1 \\ x(4) & \frac{1}{((\mathrm{p2}-q)^2+i\eta)} & 1 \\ x(5) & \frac{1}{((\mathrm{p1}-\mathrm{p2})^2-\mathrm{m3}^2+i\eta)} & 1 \end{pmatrix}, \begin{pmatrix} x(1)+x(3)+x(5) & -x(5) \\ -x(5) & x(2)+x(4)+x(5) \end{pmatrix}, \\ \\ \left\{x(3)q^{\mathrm{FCGV(mu)}}, x(4)q^{\mathrm{FCGV(mu)}}\right\}, \mathrm{m1}^2(-x(1))-\mathrm{m2}^2x(2)-\mathrm{m3}^2x(5)+q^2x(3)+q^2x(4), 1, 0 \end{array}\right\}$$

```
FCFeynmanPrepare[{topo1, topo2}, Names -> x]
```

$$\left(\begin{array}{ll} x(1)x(2)+x(3)x(2)+x(5)x(2)+x(1)x(4)+x(3)x(4)+x(1)x(5)+x(3)x(5)+x(4)x(5) & \\ \\ x(1)x(2)+x(3)x(2)+x(5)x(2)+x(1)x(4)+x(3)x(4)+x(1)x(5)+x(3)x(5)+x(4)x(5) & M^2x(2)x(3)^2+M^2x(1)x(4) \\ \\ \end{array}\right.$$

```
FCFeynmanPrepare[{GLI["prop2Lv1", {1, 1, 1, 1, 0}], GLI["prop2Lv2", {1, 1,
0, 0, 1}]},
  {topo1, topo2}, Names -> x]
```

$$\left(\begin{array}{ll} (x(1)+x(3))(x(2)+x(4)) & \mathrm{m1}^2x(1)^2x(2)+\mathrm{m1}^2x(1)x(2)x(3)+\mathrm{m1}^2x(1)^2x(4)+\mathrm{m1}^2x(1)x(3)x(4)+\mathrm{m2}^2x(1)x(\\ x(1)x(2)+x(3)x(2)+x(1)x(3) & (x(1 \end{array}\right)$$

FCFeynmanPrepare can also handle products of **GLI**s. In this case it will automatically introduce dummy names for the loop momenta (the name generation is controlled by the **LoopMomentum** option).

```
topo = FCTopology[
   prop2Ltopo13311, {SFAD[{{I*p1, 0}, {-m1^2, -1}, 1}], SFAD[{{I*(p1 + q1),
0}, {-
        m3^2, -1}, 1}], SFAD[{{I*p3, 0}, {-m3^2, -1}, 1}], SFAD[{{I*(p3 +
q1), 0}, {-m1^2,
       -1}, 1}], SFAD[{{I*(p1 - p3), 0}, {-m1^2, -1}, 1}]}, {p1, p3}, {q1},
       {SPD[q1, q1] -> m1^2}, {}]
```

$$\mathrm{FCTopology}\left(\mathrm{prop2Ltopo13311}, \left\{\frac{1}{(-\mathrm{p1}^2+\mathrm{m1}^2-i\eta)}, \frac{1}{(-(\mathrm{p1}+\mathrm{q1})^2+\mathrm{m3}^2-i\eta)}, \frac{1}{(-\mathrm{p3}^2+\mathrm{m3}^2-i\eta)}, \frac{1}{(-(\mathrm{p3}+\mathrm{q1})^2+\mathrm{m1}^2-i\eta)}, \frac{1}{(-(\mathrm{p1}-\mathrm{p3})^2+\mathrm{m1}^2-i\eta)}\right\}, \{\mathrm{p1}, \mathrm{p3}\}, \{\mathrm{q1}\}, \left\{\mathrm{q1}^2 \to \mathrm{m1}^2\right\}, \{\}\right)$$

```
FCFeynmanPrepare[GLI[prop2Ltopo13311, {1, 0, 0, 0, 0}]^2, topo, Names -> x,
FCE -> True,
  LoopMomenta -> Function[{x, y}, lmom[x, y]]]
```

$$\left\{x(1)x(2), -\mathrm{m1}^2x(1)x(2)(x(1)+x(2)), \begin{pmatrix} x(1) & \frac{1}{(-\mathrm{lmom}(1,1)^2+\mathrm{m1}^2-i\eta)} & 1 \\ x(2) & \frac{1}{(-\mathrm{lmom}(2,1)^2+\mathrm{m1}^2-i\eta)} & 1 \end{pmatrix}, \begin{pmatrix} -x(1) & 0 \\ 0 & -x(2) \end{pmatrix}, \{0,0\}, \mathrm{m1}^2(x(1)+x(2)), 1, 0\right\}$$

9.28 FCFeynmanProjectiveQ

FCFeynmanProjectiveQ[int, x] checks if the given Feynman parameter integral (without prefactors) depending on x[1], x[2], ... is a projective form.

It is similar to **FCFeynmanProjectivize** but unlike the former it simply returns **True** or **False** depending on whether the integral is projective or not.

9.28.1 See also

Overview, FCFeynmanParametrize, FCFeynmanPrepare, FCFeynmanProjectivize.

9.28.2 Examples

```
int = SFAD[{p3, mg^2}] SFAD[{p3 - p1, mg^2}] SFAD[{{0, -2 p1 . q}}]
```

$$\frac{1}{(\text{p3}^2 - \text{mg}^2 + i\eta)((\text{p3} - \text{p1})^2 - \text{mg}^2 + i\eta)(-2(\text{p1} \cdot q) + i\eta)}$$

```
fp = FCFeynmanParametrize[int, {p1, p3}, Names -> x, Indexed -> True,
FCReplaceD -> {D -> 4 - 2 ep},
   Simplify -> True, Assumptions -> {mg > 0, ep > 0}, FinalSubstitutions ->
   {SPD[q] -> qq, mg^2 -> mg2}]
```

$$\left\{(x(2)x(3))^{3\,\text{ep}-3}\left((x(2)+x(3))\left(\text{mg2}x(2)x(3)+\text{qq}x(1)^2\right)\right)^{1-2\,\text{ep}}, -\Gamma(2\,\text{ep}-1), \{x(1), x(2), x(3)\}\right\}$$

```
FCFeynmanProjectiveQ[fp[[1]], x]
```

True

```
FCFeynmanProjectiveQ[(x[1] + x[2])^(-2 + 2*ep)/(mb2*(x[1]^2 + x[1]*x[2] +
x[2]^2))^ep, x]
```

True

Feynman parametrization derived from propagator representation should be projective in most cases. However, arbitrary Feynman parameter integral do not necessarily have this property.

```
FCFeynmanProjectiveQ[x[1]^(x - 1) (x[2])^(y - 1), x]
```

False

9.29 FCFeynmanProjectivize

FCFeynmanProjectivize[int, x] checks if the given Feynman parameter integral (without prefactors) depending on x[1], x[2], ... is a projective form. If this is not the case, the integral will be projectivized.

Projectivity is a necessary condition for computing the integral with the aid of the Cheng-Wu theorem

9.29.1 See also

Overview, FCFeynmanParametrize, FCFeynmanPrepare, FCFeynmanProjectiveQ.

9.29.2 Examples

```
int = SFAD[{p3, mg^2}] SFAD[{p3 - p1, mg^2}] SFAD[{{0, -2 p1 . q}}]
```

$$\frac{1}{(\text{p3}^2 - \text{mg}^2 + i\eta)((\text{p3} - \text{p1})^2 - \text{mg}^2 + i\eta)(-2(\text{p1} \cdot q) + i\eta)}$$

```
fp = FCFeynmanParametrize[int, {p1, p3}, Names -> x, Indexed -> True,
FCReplaceD -> {D -> 4 - 2 ep},
   Simplify -> True, Assumptions -> {mg > 0, ep > 0}, FinalSubstitutions ->
   {SPD[q] -> qq, mg^2 -> mg2}]
```

$$\left\{(x(2)x(3))^{3\,\text{ep}-3}\left((x(2)+x(3))\left(\text{mg2}x(2)x(3)+\text{qq}x(1)^2\right)\right)^{1-2\,\text{ep}}, -\Gamma(2\,\text{ep}-1), \{x(1), x(2), x(3)\}\right\}$$

```
FCFeynmanProjectivize[fp[[1]], x]
```

FCFeynmanProjectivize: The integral is already projective, no further transformations are required.

$$(x(2)x(3))^{3\,\text{ep}-3}\left((x(2)+x(3))\left(\text{mg2}x(2)x(3)+\text{qq}x(1)^2\right)\right)^{1-2\,\text{ep}}$$

```
FCFeynmanProjectivize[(x[1] + x[2])^(-2 + 2*ep)/(mb2*(x[1]^2 + x[1]*x[2] +
        x[2]^2))^ep, x]
```

FCFeynmanProjectivize: The integral is already projective, no further transformations are required.

$$(x(1)+x(2))^{2\,\mathrm{ep}-2}\left(\mathrm{mb2}\left(x(1)^2+x(2)x(1)+x(2)^2\right)\right)^{-\mathrm{ep}}$$

Feynman parametrizations derived from propagator representations should be projective in most cases. However, arbitrary Feynman parameter integrals do not necessarily have this property.

```
FCFeynmanProjectivize[x[1]^(x - 1) (x[2])^(y - 1), x]
```

FCFeynmanProjectivize: The integral is not projective, trying to projectivize.

FCFeynmanProjectivize: Projective transformation successful, the integral is now projective.

$$\frac{\left(\frac{x(1)}{x(1)+x(2)}\right)^{x-1}\left(\frac{x(2)}{x(1)+x(2)}\right)^{y-1}}{(x(1)+x(2))^2}$$

9.30 FCLoopAddEdgeTags

`FCLoopAddEdgeTags[edges_List, labels_List]` adds user-defined styles and labels to the given edges using the provided list of labels. Styles and labels are attached using the replacement rules provided via the **`Style`** and **`Labeled`** options.

9.30.1 See also

Overview, FCLoopGraphPlot.

9.30.2 Examples

If you use **`FCLoopIntegralToGraph`** for visualizing loop integrals, the first two entries of its output can be used as the input for FCLoopAddEdgeTags, e.g.

```
FCLoopAddEdgeTags[FCLoopIntegralToGraph[FAD[p, p - k], {p}][[1 ;; 2]]]

GraphPlot[%]
```

$$\{-3 \leftrightarrow 2, -1 \leftrightarrow 1, 1 \leftrightarrow 2, 1 \leftrightarrow 2\}$$

If you just want to plot the obtained graph, it is easier to process the output of **FCLoopIntegralToGraph** directly with **FCLoopGraphPlot**, which internally uses **FCLoopAddEdgeTags**.

```
FCLoopIntegralToGraph[FAD[p, p - k], {p}]

FCLoopGraphPlot[%]
```

$$\left\{\{-3\to 2,-1\to 1,1\to 2,1\to 2\},\{k,k,\{p,1,0\},\{p-k,1,0\}\},\left\{0,0,\frac{1}{(p^2+i\eta)},\frac{1}{((p-k)^2+i\eta)}\right\},1\right\}$$

9.31 FCLoopGraphPlot

FCLoopGraphPlot[{edges, labels}] visualizes the graph of the given loop integral using the provided list of edges, styles and labels using the built-in function **Graph**. The Option **Graph** can be used to pass options to the **Graph** objects.

By default, **FCLoopGraphPlot** returns a **Graph**. When using Mathematica 12.2 or newer, it is also possible to return a **Graphics** object created by **GraphPlot**. For this the option **GraphPlot** must be set to a list of options that will be passed to **GraphPlot**. An empty list is also admissible. For example, **FCLoopGraphPlot[int, GraphPlot -> {MultiedgeStyle -> 0.35, Frame -> True}]**.

Given a list of **Graph** or **Graphics** objects created by **FCLoopGraphPlot**, a nice way to get a better overview is to employ **Magnify[Grid[(Partition[out, UpTo[4]])], 0.9]**.

Notice that older Mathematica versions have numerous shortcomings in the graph drawing capabilities that cannot be reliably worked around. This why to use **FCLoopGraphPlot** you need to have at least Mathematica 11.0 or newer. For best results we recommend using Mathematica 12.2 or newer.

9.31.1 See also

Overview, FCLoopIntegralToGraph.

9.31.2 Examples

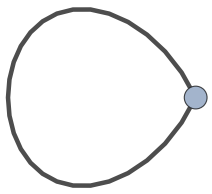
Showcases

1-loop tadpole

```
FCLoopIntegralToGraph[FAD[{p, m}], {p}]

FCLoopGraphPlot[%]
```

$$\left\{\{1 \to 1\}, \begin{pmatrix} p & 1 & -m^2 \end{pmatrix}, \left\{\frac{1}{(p^2 - m^2 + i\eta)}\right\}, 1\right\}$$



1-loop massless bubble

```
FCLoopIntegralToGraph[FAD[p, p - q], {p}]

FCLoopGraphPlot[%]
```

$$\left\{\{-3 \to 2, -1 \to 1, 1 \to 2, 1 \to 2\}, \{q, q, \{p, 1, 0\}, \{p - q, 1, 0\}\}, \left\{0, 0, \frac{1}{(p^2 + i\eta)}, \frac{1}{((p - q)^2 + i\eta)}\right\}, 1\right\}$$

1-loop massive bubble

```
FCLoopIntegralToGraph[FAD[{p, m1}, {p - q, m2}], {p}]

FCLoopGraphPlot[%]
```

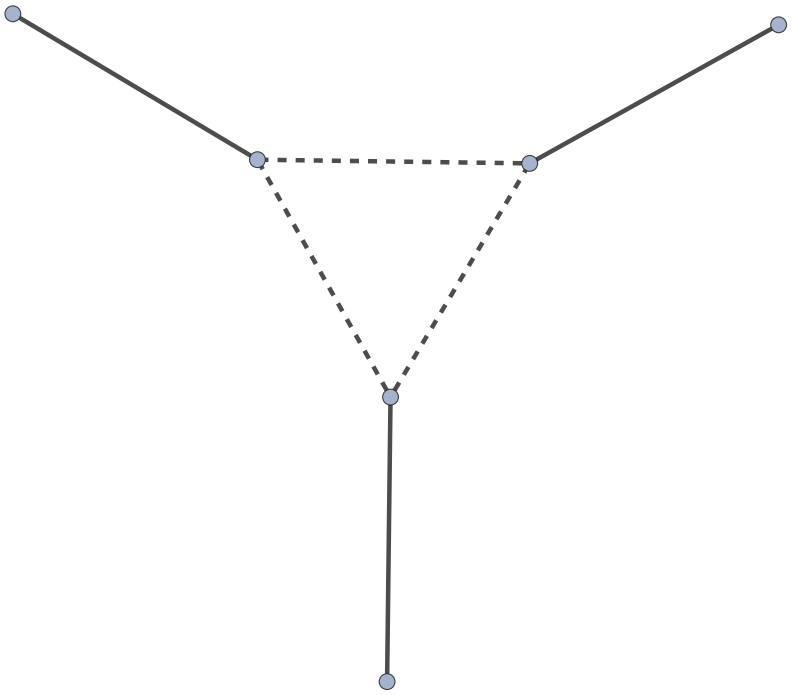
$$\left\{\{-3\to 2,-1\to 1,1\to 2,1\to 2\},\{q,q,\{p,1,-\text{m1}^2\},\{p-q,1,\right.$$
$$\left.-\text{m2}^2\}\},\left\{0,0,\frac{1}{(p^2-\text{m1}^2+i\eta)},\frac{1}{((p-q)^2-\text{m2}^2+i\eta)}\right\},1\right\}$$

1-loop massless triangle

```
FCLoopIntegralToGraph[FAD[p, p + q1, p + q1 + q2], {p}]

FCLoopGraphPlot[%]
```

$$\left\{\{-3\to 3,-2\to 1,-1\to 2,1\to 2,1\to 3,2\to 3\},\{\text{q1}-\text{q2},\text{q1},\text{q2},\{p+\text{q1},1,0\},\right.$$
$$\left.\{p+\text{q1}+\text{q2},1,0\},\{p,1,0\}\},\left\{0,0,0,\frac{1}{(p^2+i\eta)},\frac{1}{((p+\text{q1})^2+i\eta)},\frac{1}{((p+\text{q1}+\text{q2})^2+i\eta)}\right\},1\right\}$$

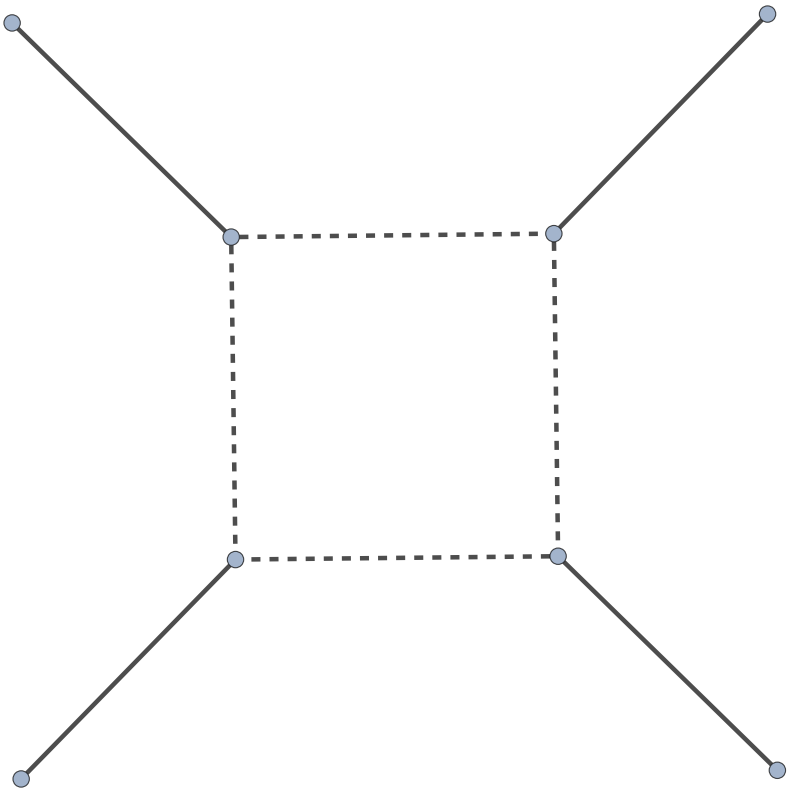


1-loop massless box

```
FCLoopIntegralToGraph[FAD[p, p + q1, p + q1 + q2, p + q1 + q2 + q3], {p}]

FCLoopGraphPlot[%]
```

$$\left\{\{-4\to 4,-3\to 1,-2\to 2,-1\to 3,1\to 2,1\to 4,2\to 3,3\to 4\},\{\text{q1}-\text{q2}-\text{q3},\text{q1},\text{q2},\right.$$
$$\text{q3},\{p+\text{q1}+\text{q2},1,0\},\{p+\text{q1}+\text{q2}+\text{q3},1,0\},\{p+\text{q1},1,0\},\{p,1,0\}\},\left\{0,0,0,0,\right.$$
$$\left.\left.\frac{1}{(p^2+i\eta)},\frac{1}{((p+\text{q1})^2+i\eta)},\frac{1}{((p+\text{q1}+\text{q2})^2+i\eta)},\frac{1}{((p+\text{q1}+\text{q2}+\text{q3})^2+i\eta)}\right\},1\right\}$$

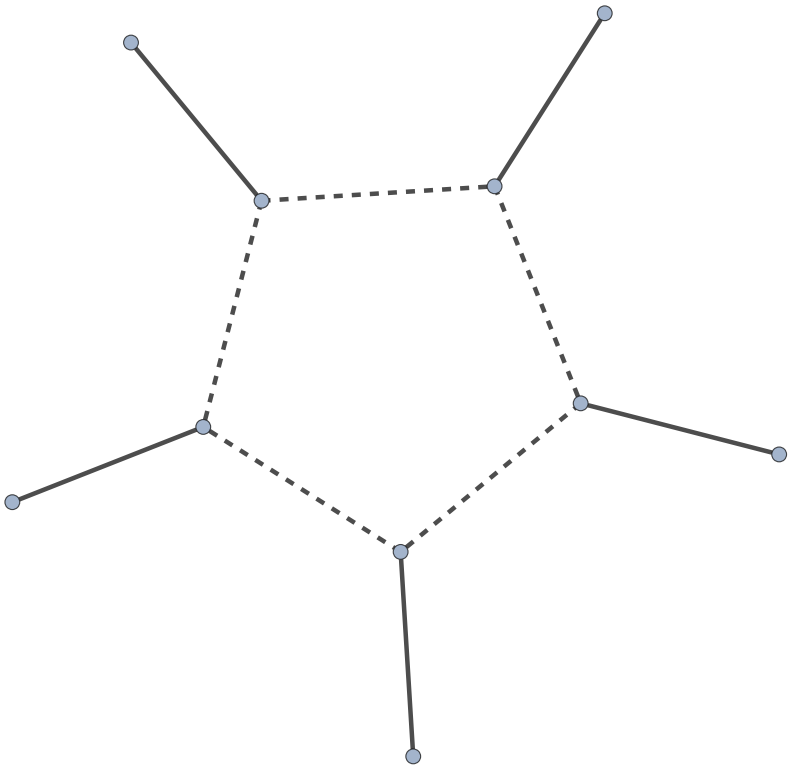


1-loop massless pentagon

```
FCLoopIntegralToGraph[FAD[p, p + q1, p + q1 + q2, p + q1 + q2 + q3, p + q1
+ q2 + q3 + q4], {p}]

FCLoopGraphPlot[%]
```

$$\left\{\{-5\to 5,-4\to 1,-3\to 2,-2\to 3,-1\to 4,1\to 2,1\to 5,2\to 3,3\to 4,4\to 5\},\right.$$
$$\{\text{q1}-\text{q2}-\text{q3}-\text{q4},\text{q1},\text{q2},\text{q3},\text{q4},\{p+\text{q1}+\text{q2}+\text{q3},1,0\},\{p+\text{q1}+\text{q2}+\text{q3}+\text{q4},1,$$
$$0\},\{p+\text{q1}+\text{q2},1,0\},\{p+\text{q1},1,0\},\{p,1,0\}\},\left\{0,0,0,0,0,\frac{1}{(p^2+i\eta)},\frac{1}{((p+\text{q1})^2+i\eta)},\right.$$
$$\left.\left.\frac{1}{((p+\text{q1}+\text{q2})^2+i\eta)},\frac{1}{((p+\text{q1}+\text{q2}+\text{q3})^2+i\eta)},\frac{1}{((p+\text{q1}+\text{q2}+\text{q3}+\text{q4})^2+i\eta)}\right\},1\right\}$$

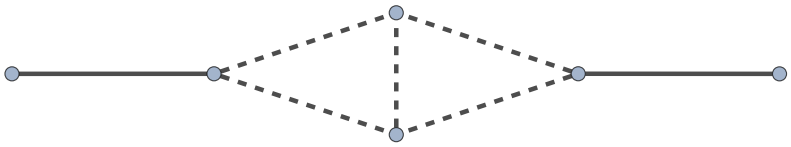


2-loop massless self-energy

```
FCLoopIntegralToGraph[FAD[p1, p2, Q - p1 - p2, Q - p1, Q - p2], {p1, p2}]

FCLoopGraphPlot[%]
```

$$\left\{\{-3\to 2,-1\to 1,1\to 3,1\to 4,2\to 3,2\to 4,3\to 4\},\{Q,Q,\{\text{p2},1,0\},\right.$$
$$\{Q-\text{p2},1,0\},\{Q-\text{p1},1,0\},\{\text{p1},1,0\},\{-\text{p1}-\text{p2}+Q,1,0\}\},\left\{0,0,\frac{1}{(\text{p2}^2+i\eta)},\right.$$
$$\left.\left.\frac{1}{(\text{p1}^2+i\eta)},\frac{1}{((Q-\text{p2})^2+i\eta)},\frac{1}{((Q-\text{p1})^2+i\eta)},\frac{1}{((-\text{p1}-\text{p2}+Q)^2+i\eta)}\right\},1\right\}$$

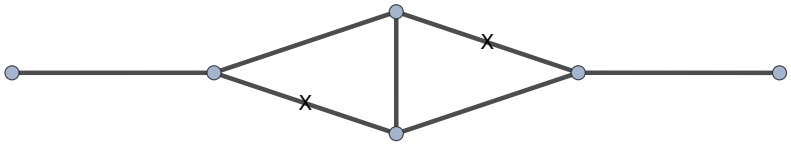


Same topology as before but now fully massive and with some dots

```
FCLoopIntegralToGraph[FAD[{p1, m}, {p2, m2}, {Q - p1 - p2, m}, {Q - p1, m,
2}, {Q - p2, m,2}], {p1, p2}]

FCLoopGraphPlot[%]
```

$$\left\{\{-3\to 2,-1\to 1,1\to 3,1\to 4,2\to 3,2\to 4,3\to 4\},\{Q,Q,\{\text{p2},1,-\text{m2}^2\},\{Q-\text{p2},2,-m^2\},\{Q-\text{p1},2,-m^2\},\{\text{p1},1,-m^2\},\{-\text{p1}-\text{p2}+Q,1,-m^2\}\},\left\{0,0,\frac{1}{(\text{p2}^2-\text{m2}^2+i\eta)},\frac{1}{(\text{p1}^2-m^2+i\eta)},\frac{1}{((Q-\text{p2})^2-m^2+i\eta)},\frac{1}{((Q-\text{p1})^2-m^2+i\eta)},\frac{1}{((-\text{p1}-\text{p2}+Q)^2-m^2+i\eta)}\right\},1\right\}$$

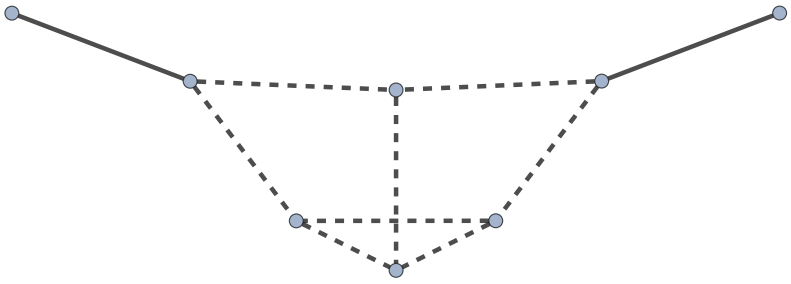


3-loop massless self-energy

```
FCLoopIntegralToGraph[FAD[p1, p2, p3, Q - p1 - p2 - p3, Q - p1 - p2, Q - p1, Q - p2, p1 + p3], {p1, p2, p3}]

FCLoopGraphPlot[%]
```

$$\left\{\{-3\to 2,-1\to 1,1\to 5,1\to 6,2\to 3,2\to 5,3\to 4,3\to 6,4\to 5,4\to 6\},\{Q,Q,\{\text{p2},1,0\},\{Q-\text{p2},1,0\},\{\text{p1},1,0\},\{Q-\text{p1},1,0\},\{\text{p3},1,0\},\{\text{p1}+\text{p3},1,0\},\{-\text{p1}-\text{p2}+Q,1,0\},\{-\text{p1}-\text{p2}-\text{p3}+Q,1,0\}\},\left\{0,0,\frac{1}{(\text{p3}^2+i\eta)},\frac{1}{(\text{p2}^2+i\eta)},\frac{1}{(\text{p1}^2+i\eta)},\frac{1}{((\text{p1}+\text{p3})^2+i\eta)},\frac{1}{((Q-\text{p2})^2+i\eta)},\frac{1}{((Q-\text{p1})^2+i\eta)},\frac{1}{((-\text{p1}-\text{p2}+Q)^2+i\eta)},\frac{1}{((-\text{p1}-\text{p2}-\text{p3}+Q)^2+i\eta)}\right\},1\right\}$$



3-loop self-energy with two massive lines

```
FCLoopIntegralToGraph[Times @@ {SFAD[{{p1, 0}, {m^2, 1}, 1}], SFAD[{{p2, 0}, {0, 1}, 1}],
     SFAD[{{p3, 0}, {0, 1}, 1}], SFAD[{{p1 + p2 + p3 - Q, 0}, {0, 1}, 1}], SFAD[{{p2 + p3, 0}, {0, 1}, 1}],
```

```
    SFAD[{{p2 - Q, 0}, {0, 1}, 1}], SFAD[{{p1 - Q, 0}, {m^2, 1},
1}],SFAD[{{p2 + p3 - Q, 0}, {0, 1}, 1}]},
   {p1, p2, p3}]

FCLoopGraphPlot[%]
```

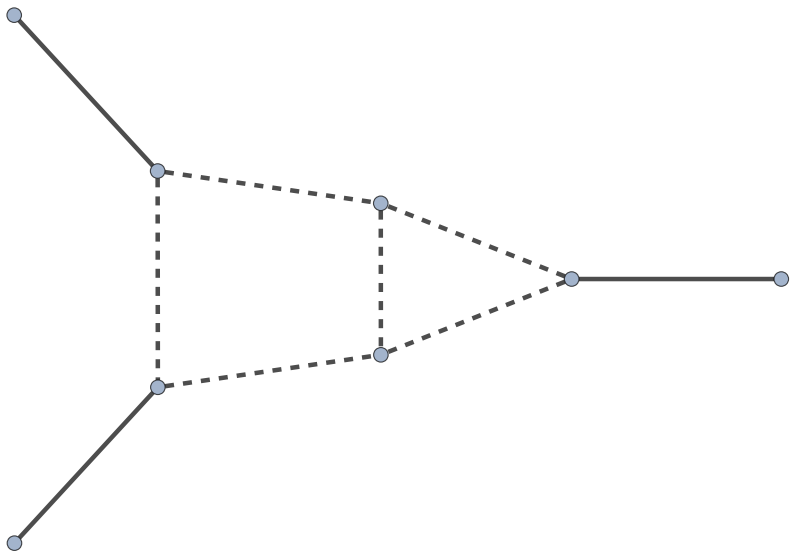
$$\left\{\{-3\to 2,-1\to 1,1\to 3,1\to 4,2\to 5,2\to 6,3\to 4,3\to 5,4\to 6,5\to 6\},\left\{Q,Q,\{\text{p2},1,0\},\right.\right.$$
$$\{\text{p2}-Q,1,0\},\left\{\text{p1}-Q,1,-m^2\right\},\left\{\text{p1},1,-m^2\right\},\{\text{p3},1,0\},\{\text{p2}+\text{p3},1,0\},\{\text{p2}+\text{p3}-Q,1,0\},$$
$$\left.\{\text{p1}+\text{p2}+\text{p3}-Q,1,0\}\right\},\left\{0,0,\frac{1}{(\text{p3}^2+i\eta)},\frac{1}{(\text{p2}^2+i\eta)},\frac{1}{((\text{p2}+\text{p3})^2+i\eta)},\frac{1}{((\text{p2}-Q)^2+i\eta)},\right.$$
$$\left.\left.\frac{1}{(\text{p1}^2-m^2+i\eta)},\frac{1}{((\text{p2}+\text{p3}-Q)^2+i\eta)},\frac{1}{((\text{p1}+\text{p2}+\text{p3}-Q)^2+i\eta)},\frac{1}{((\text{p1}-Q)^2-m^2+i\eta)}\right\},1\right\}$$

2-loop triangle

```
FCLoopIntegralToGraph[FAD[p1, p2, Q1 + p1, Q2 - p1, Q1 + p1 + p2, Q2 - p1 -
p2], {p1, p2}]

FCLoopGraphPlot[%]
```

$$\left\{\{-3\to 3,-2\to 1,-1\to 2,1\to 2,1\to 5,2\to 4,3\to 4,3\to 5,4\to 5\},\{\text{Q1}-\text{Q2},\right.$$
$$\text{Q1},\text{Q2},\{\text{p1},1,0\},\{\text{Q2}-\text{p1},1,0\},\{\text{p1}+\text{Q1},1,0\},\{\text{p1}+\text{p2}+\text{Q1},1,0\},$$
$$\{-\text{p1}-\text{p2}+\text{Q2},1,0\},\{\text{p2},1,0\}\},\left\{0,0,0,\frac{1}{(\text{p2}^2+i\eta)},\frac{1}{(\text{p1}^2+i\eta)},\frac{1}{((\text{p1}+\text{Q1})^2+i\eta)},\right.$$
$$\left.\left.\frac{1}{((\text{p1}+\text{p2}+\text{Q1})^2+i\eta)},\frac{1}{((\text{Q2}-\text{p1})^2+i\eta)},\frac{1}{((-\text{p1}-\text{p2}+\text{Q2})^2+i\eta)}\right\},1\right\}$$



Special cases

Not all loop integrals admit a graph representation. Furthermore, an integral may have a weird momentum routing that cannot be automatically recognized by the employed algorithm. Consider e.g.

```
topo = FCTopology[TRIX1, {SFAD[{{p2, 0}, {0, 1}, 1}], SFAD[{{p1 + Q1, 0},
{0, 1}, 1}],
    SFAD[{{p1 + p2 + Q1, 0}, {0, 1}, 1}], SFAD[{{-p1 + Q2, 0}, {0, 1}, 1}],
    SFAD[{{-p1 - p2 + Q2, 0}, {0, 1}, 1}]}, {p1, p2}, {Q1, Q2}, {}, {}]
```

$$\text{FCTopology}\left(\text{TRIX1}, \left\{\frac{1}{(\text{p2}^2+i\eta)}, \frac{1}{((\text{p1}+\text{Q1})^2+i\eta)}, \frac{1}{((\text{p1}+\text{p2}+\text{Q1})^2+i\eta)}, \frac{1}{((\text{Q2}-\text{p1})^2+i\eta)}, \frac{1}{((-\text{p1}-\text{p2}+\text{Q2})^2+i\eta)}\right\}, \{\text{p1},\text{p2}\}, \{\text{Q1},\text{Q2}\}, \{\}, \{\}\right)$$

Here **FCLoopIntegralToGraph** has no way to know that the actual momentum is Q1+Q2, i.e. it is a 2- and not 3-point function

```
FCLoopIntegralToGraph[topo]
```

FCLoopIntegralToGraph: Error! FCLoopIntegralToGraph encountered a fatal problem and must abort the computation. The problem reads: Failed to reconstruct the graph of the given loop integral. If the integral factorizes, try increasing VertexDegree

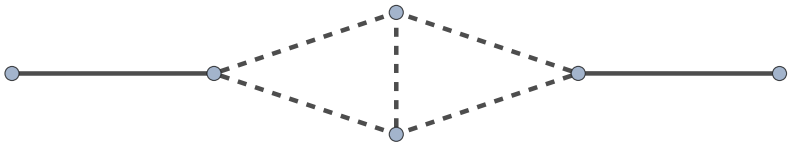
False

However, if we explicitly provide this information, in many cases the function can still perform the proper reconstruction

```
FCLoopIntegralToGraph[topo, Momentum -> {Q1 + Q2}]

FCLoopGraphPlot[%]
```

$$\left\{\{-3\to 2,-1\to 1,1\to 3,1\to 4,2\to 3,2\to 4,3\to 4\},\{\text{Q1}+\text{Q2},\text{Q1}+\text{Q2},\{\text{p1}+\text{Q1},1,0\},\right.$$
$$\{\text{Q2}-\text{p1},1,0\},\{\text{p1}+\text{p2}+\text{Q1},1,0\},\{-\text{p1}-\text{p2}+\text{Q2},1,0\},\{\text{p2},1,0\}\},\left\{0,0,\frac{1}{(\text{p2}^2+i\eta)},\right.$$
$$\left.\left.\frac{1}{((\text{p1}+\text{Q1})^2+i\eta)},\frac{1}{((\text{p1}+\text{p2}+\text{Q1})^2+i\eta)},\frac{1}{((\text{Q2}-\text{p1})^2+i\eta)},\frac{1}{((-\text{p1}-\text{p2}+\text{Q2})^2+i\eta)}\right\},1\right\}$$



And here is another example. This NRQCD integral from arXiv:1907.08227 looks like as if it has only one external momentum flowing in

```
FCLoopIntegralToGraph[FAD[{k, m}, l + p, l - p, k + l], {k, l}]
```

FCLoopIntegralToGraph: Error! FCLoopIntegralToGraph encountered a fatal problem and must abort the computation. The problem reads: Failed to reconstruct the graph of the given loop integral. If the integral factorizes, try increasing VertexDegree

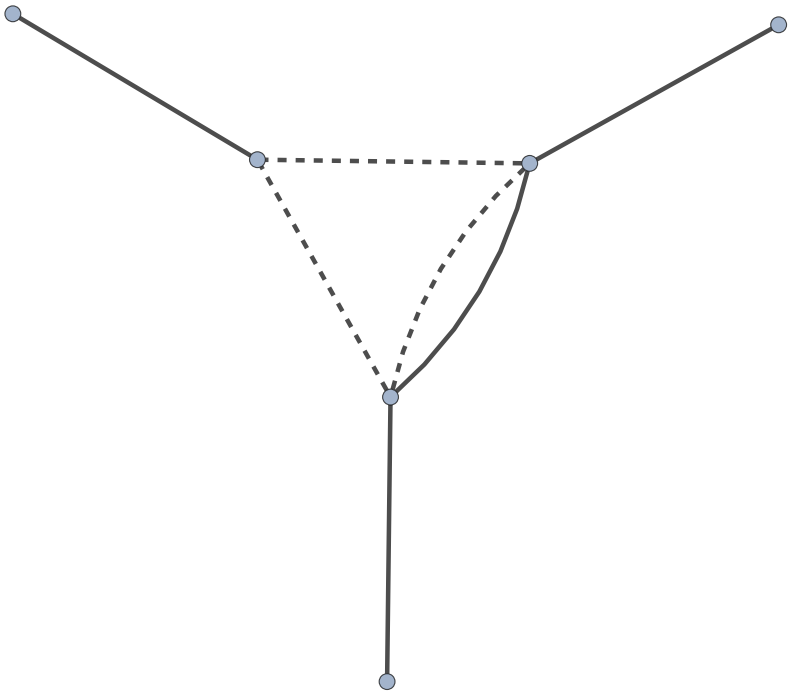
False

while in reality there are two of them: **p** and **2p**

```
FCLoopIntegralToGraph[FAD[{k, m}, l + p, l - p, k + l], {k, l},
   Momentum -> {2 p, p}, FCE -> True]

FCLoopGraphPlot[%]
```

$$\left\{\{-4\to 3,-2\to 2,-1\to 1,1\to 2,1\to 3,2\to 3,2\to 3\},\{3p,2p,p,\{l+p,1,0\},\{l-p,1,0\},\{k+l,\right.$$
$$\left.1,0\},\{k,1,-m^2\}\},\left\{0,0,0,\frac{1}{((l+p)^2+i\eta)},\frac{1}{((k+l)^2+i\eta)},\frac{1}{((l-p)^2+i\eta)},\frac{1}{(k^2-m^2+i\eta)}\right\},1\right\}$$



In this case the correct form of the external momentum can be deduced upon performing some elementary shifts. The direct application of the function fails

```
ex = FCTopology[topo1X12679, {SFAD[{{p1, 0}, {0, 1}, 1}], SFAD[{{p2 + p3,
0}, {0, 1}, 1}],
    SFAD[{{p2 - Q, 0}, {0, 1}, 1}], SFAD[{{p1 + p3 - Q, 0}, {0, 1}, 1}]},
{p1, p2, p3}, {Q}, {}, {}]
```

$$\text{FCTopology}\left(\text{topo1X12679}, \left\{\frac{1}{(\text{p1}^2 + i\eta)}, \frac{1}{((\text{p2}+\text{p3})^2 + i\eta)}, \frac{1}{((\text{p2} - Q)^2 + i\eta)}, \frac{1}{((\text{p1}+\text{p3} - Q)^2 + i\eta)}\right\}, \{\text{p1}, \text{p2}, \text{p3}\}, \{Q\}, \{\}, \{\}\right)$$

```
FCLoopIntegralToGraph[ex]
```

FCLoopIntegralToGraph: Error! FCLoopIntegralToGraph encountered a fatal problem and must abort the computation. The problem reads: Failed to reconstruct the graph of the given loop integral. If the integral factorizes, try increasing VertexDegree

False

Yet let us consider

```
exShifted = FCReplaceMomenta[ex, {p2 -> p2 - p3 + p1 - Q, p3 -> p3 - p1 +
Q}]
```

$$\text{FCTopology}\left(\text{topo1X12679}, \left\{\frac{1}{(\text{p1}^2+i\eta)}, \frac{1}{(\text{p2}^2+i\eta)}, \frac{1}{((\text{p1}+\text{p2}-\text{p3}-2Q)^2+i\eta)}, \frac{1}{(\text{p3}^2+i\eta)}\right\}, \{\text{p1}, \text{p2}, \text{p3}\}, \{Q\}, \{\}, \{\}\right)$$

Now we immediately see that the proper external momentum to consider is **2Q** instead of just **Q**

```
FCLoopIntegralToGraph[exShifted, Momentum -> {2 Q}]

FCLoopGraphPlot[%]
```

$$\left\{\{-3\to 2,-1\to 1,1\to 2,1\to 2,1\to 2,1\to 2\}, \{2Q, 2Q, \{\text{p3}, 1, 0\}, \{\text{p2}, 1, 0\}, \{\text{p1}, 1, 0\}, \{\text{p1}+\text{p2}-\text{p3}-2Q, 1, 0\}\}, \left\{0, 0, \frac{1}{(\text{p3}^2+i\eta)}, \frac{1}{(\text{p2}^2+i\eta)}, \frac{1}{(\text{p1}^2+i\eta)}, \frac{1}{((\text{p1}+\text{p2}-\text{p3}-2Q)^2+i\eta)}\right\}, 1\right\}$$

When dealing with products of tadpole integrals, the function may not always recognize that the appearing external momenta are spurious. For example, here there is no **q** momentum flowing through any of the lines

```
int = SFAD[{{ p1, 0}, {mg^2, 1}, 1}] SFAD[{{ p3, -2 p3 . q}, {0, 1}, 1}]

FCLoopIntegralToGraph[int, {p1, p3}]
```

$$\frac{1}{(\text{p1}^2-\text{mg}^2+i\eta)(\text{p3}^2-2(\text{p3}\cdot q)+i\eta)}$$

FCLoopIntegralToGraph: Error! FCLoopIntegralToGraph encountered a fatal problem and must abort the computation. The problem reads: Failed to reconstruct the graph of the given loop integral. If the integral factorizes, try increasing VertexDegree

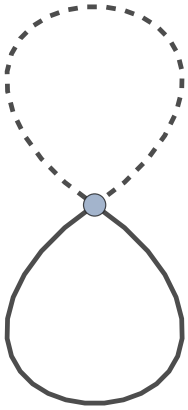
False

In this case we may explicitly tell the function that this integral doesn't depend on any external momenta

```
FCLoopIntegralToGraph[int, {p1, p3}, Momentum -> {}]

FCLoopGraphPlot[%]
```

$$\left\{\{1 \to 1, 1 \to 1\}, \begin{pmatrix} \text{p1} & 1 & -\text{mg}^2 \\ \text{p3} - q & 1 & 0 \end{pmatrix}, \left\{\frac{1}{(\text{p1}^2 - \text{mg}^2 + i\eta)}, \frac{1}{(\text{p3}^2 - 2(\text{p3} \cdot q) + i\eta)}\right\}, 1\right\}$$



Eye candy

The **`Style`** option can be used to label lines carrying different masses in a particular way

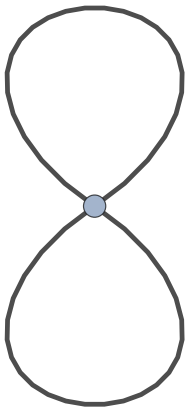
```
OptionValue[FCLoopGraphPlot, Style]
```

{{InternalLine, _, _, 0} :→ {Dashed, Thick, Black}, {InternalLine, _, _, FeynCalc`FCLoopGraphPlot`Private`mm_ /; FeynCalc`FCLoopGraphPlot`Private`mm =!=0} :→ {Thick, Black}, {ExternalLine, _} :→ {Thick, Black}}

When dealing with factorizing integral it might be necessary to increase **VertexDegree** to **7** or **8** (or even a higher value, depending on the integrals)

```
FCLoopIntegralToGraph[FAD[{p1, m1}] FAD[{p2, m2}], {p1, p2}]
FCLoopGraphPlot[%]
```

$$\left\{\{1 \to 1, 1 \to 1\}, \begin{pmatrix} \text{p2} & 1 & -\text{m2}^2 \\ \text{p1} & 1 & -\text{m1}^2 \end{pmatrix}, \left\{\frac{1}{(\text{p2}^2 - \text{m2}^2 + i\eta)}, \frac{1}{(\text{p1}^2 - \text{m1}^2 + i\eta)}\right\}, 1\right\}$$



```
FCLoopIntegralToGraph[FAD[{p1, m1}] FAD[{p2, m2}] FAD[p3, p3 + q], {p1, p2,
p3},
   VertexDegree -> 7]

FCLoopGraphPlot[%]
```

$$\left\{\{-3\to 2,-1\to 1,1\to 2,1\to 2,2\to 2,2\to 2\},\{q,q,\{\text{p3},1,0\},\{\text{p3}+q,1,0\},\{\text{p2},1,-\text{m2}^2\},\right.$$
$$\left.\{\text{p1},1,-\text{m1}^2\}\right\},\left\{0,0,\frac{1}{(\text{p3}^2+i\eta)},\frac{1}{((\text{p3}+q)^2+i\eta)},\frac{1}{(\text{p2}^2-\text{m2}^2+i\eta)},\frac{1}{(\text{p1}^2-\text{m1}^2+i\eta)}\right\},1\Bigg\}$$

```
FCLoopIntegralToGraph[FAD[{p1, m1}] FAD[{p2, m2}] FAD[p3, p3 + q] FAD[{p4,
m4}],
   {p1, p2, p3, p4}, VertexDegree -> 9]

FCLoopGraphPlot[%]
```

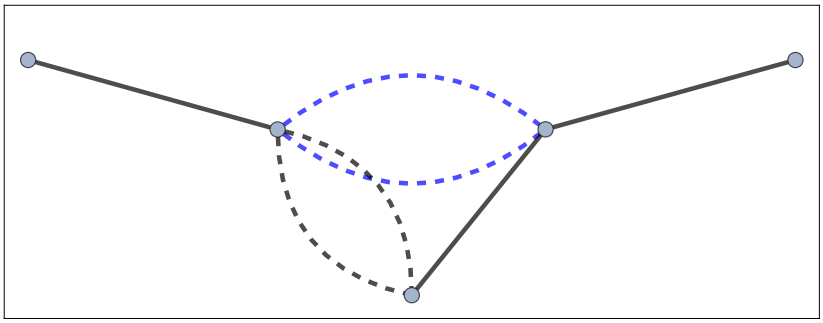
$$\left\{\{-3\to 2,-1\to 1,1\to 2,1\to 2,2\to 2,2\to 2\},\{q,q,\{\text{p3},1,0\},\{\text{p3}+q,\right.$$
$$1,0\},\{\text{p4},1,-\text{m4}^2\},\{\text{p2},1,-\text{m2}^2\},\{\text{p1},1,-\text{m1}^2\}\},\left\{0,0,\frac{1}{(\text{p3}^2+i\eta)},\right.$$
$$\left.\left.\frac{1}{((\text{p3}+q)^2+i\eta)},\frac{1}{(\text{p4}^2-\text{m4}^2+i\eta)},\frac{1}{(\text{p2}^2-\text{m2}^2+i\eta)},\frac{1}{(\text{p1}^2-\text{m1}^2+i\eta)}\right\},1\right\}$$

Here we choose to use thick dashed blue and red lines for massive lines containing **mc** and **mg** respectively. The massless lines are black an dashed.

```
FCLoopIntegralToGraph[ FAD[{k2, mb}, {k3}, {k1 - q, mc}, {k1 - k2, mc}, {k2
- k3}], {k1, k2, k3}]

Magnify[FCLoopGraphPlot[%, GraphPlot -> {MultiedgeStyle -> 0.35, Frame ->
True},
   Style -> {{"InternalLine", _, _, mm_ /; ! FreeQ[mm, mg]} -> {Red, Thick,
   Dashed},
     {"InternalLine", _, _, mm_ /; ! FreeQ[mm, mc]} -> {Blue, Thick,
     Dashed}}], 1.5]
```

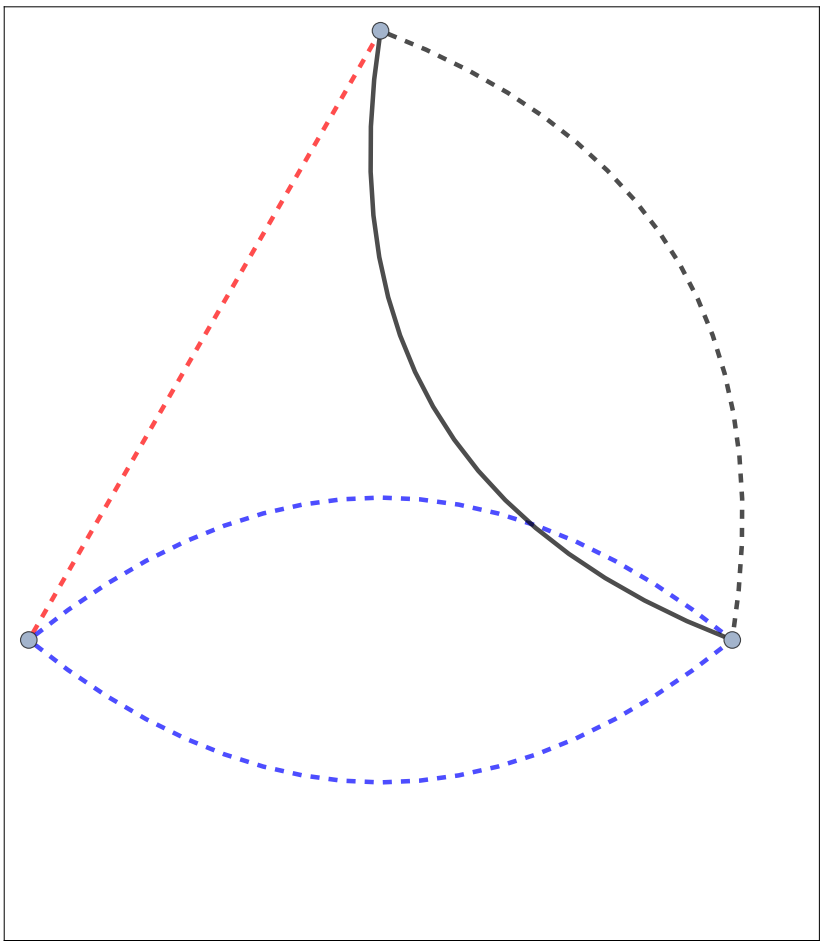
$$\left\{\{-3\to 2,-1\to 1,1\to 2,1\to 2,1\to 3,2\to 3,2\to 3\},\{q,q,\{\text{k1}-q,1,-\text{mc}^2\}\right.,$$
$$\{\text{k1}-\text{k2},1,-\text{mc}^2\},\{\text{k2},1,-\text{mb}^2\},\{\text{k3},1,0\},\{\text{k2}-\text{k3},1,0\}\},\left\{0,0,\frac{1}{(\text{k3}^2+i\eta)},\right.$$
$$\left.\left.\frac{1}{((\text{k2}-\text{k3})^2+i\eta)},\frac{1}{(\text{k2}^2-\text{mb}^2+i\eta)},\frac{1}{((\text{k1}-q)^2-\text{mc}^2+i\eta)},\frac{1}{((\text{k1}-\text{k2})^2-\text{mc}^2+i\eta)}\right\},1\right\}$$



```
FCLoopIntegralToGraph[ FAD[{k2, mg}, {k3, mc}, {k1, q}, {k1 - k2}, {k2 -
k3, mc}], {k1, k2,k3}]

Magnify[FCLoopGraphPlot[%, GraphPlot -> {MultiedgeStyle -> 0.35, Frame ->
True},
   Style -> {{"InternalLine", _, _, mm_ /; ! FreeQ[mm, mg]} -> {Red, Thick,
   Dashed},
     {"InternalLine", _, _, mm_ /; ! FreeQ[mm, mc]} -> {Blue, Thick,
     Dashed}}], 1.5]
```

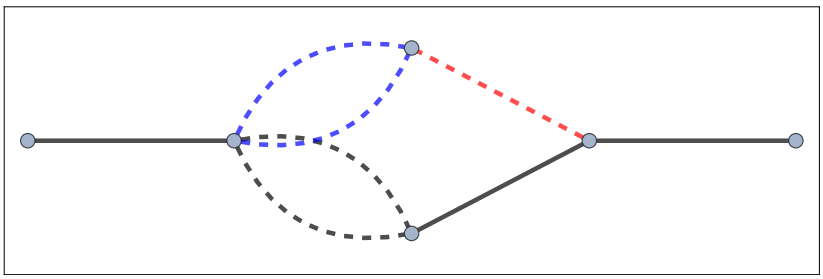
$$\left\{\{1\to 2,1\to 3,1\to 3,2\to 3,2\to 3\},\begin{pmatrix}\text{k2} & 1 & -\text{mg}^2\\ \text{k3} & 1 & -\text{mc}^2\\ \text{k2}-\text{k3} & 1 & -\text{mc}^2\\ \text{k1}-\text{k2} & 1 & 0\\ \text{k1} & 1 & -q^2\end{pmatrix},\left\{\frac{1}{(\text{k3}^2-\text{mc}^2+i\eta)},\right.\right.$$
$$\left.\left.\frac{1}{(\text{k2}^2-\text{mg}^2+i\eta)},\frac{1}{((\text{k1}-\text{k2})^2+i\eta)},\frac{1}{(\text{k1}^2-q^2+i\eta)},\frac{1}{((\text{k2}-\text{k3})^2-\text{mc}^2+i\eta)}\right\},1\right\}$$



```
FCLoopIntegralToGraph[ FAD[{k2, mg}, {k3, mc}, {k1 - q}, {k2 - q, mb}, {k1
- k2}, {k2 - k3,mc}],
   {k1, k2, k3}]

Magnify[FCLoopGraphPlot[%, GraphPlot -> {MultiedgeStyle -> 0.35, Frame ->
True},
   Style -> {{"InternalLine", _, _, mm_ /; ! FreeQ[mm, mg]} -> {Red, Thick,
   Dashed},
     {"InternalLine", _, _, mm_ /; ! FreeQ[mm, mc]} -> {Blue, Thick,
     Dashed}}], 1.5]
```

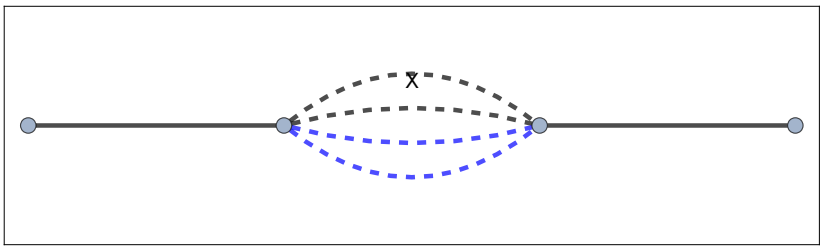
$$\left\{\{-3\to 2,-1\to 1,1\to 3,1\to 4,2\to 3,2\to 3,2\to 4,2\to 4\},\left\{q,q,\left\{\text{k2}-q,1,\right.\right.\right.$$
$$\left.\left.-\text{mb}^2\right\},\left\{\text{k2},1,-\text{mg}^2\right\},\{\text{k1}-q,1,0\},\{\text{k1}-\text{k2},1,0\},\left\{\text{k3},1,-\text{mc}^2\right\},\left\{\text{k2}-\text{k3},\right.\right.$$
$$\left.\left.1,-\text{mc}^2\right\}\right\},\left\{0,0,\frac{1}{((\text{k1}-q)^2+i\eta)},\frac{1}{(\text{k3}^2-\text{mc}^2+i\eta)},\frac{1}{(\text{k2}^2-\text{mg}^2+i\eta)},\right.$$
$$\left.\left.\frac{1}{((\text{k1}-\text{k2})^2+i\eta)},\frac{1}{((\text{k2}-q)^2-\text{mb}^2+i\eta)},\frac{1}{((\text{k2}-\text{k3})^2-\text{mc}^2+i\eta)}\right\},1\right\}$$



```
FCLoopIntegralToGraph[ FAD[{k2, 0, 2}, {k1 - q}, {k1 - k3, mc}, {k2 - k3,
mc}], {k1, k2, k3}]

Magnify[FCLoopGraphPlot[%, GraphPlot -> {MultiedgeStyle -> 0.35, Frame ->
True},
   Style -> {{"InternalLine", _, _, mm_ /; ! FreeQ[mm, mg]} -> {Red, Thick,
   Dashed},
     {"InternalLine", _, _, mm_ /; ! FreeQ[mm, mc]} -> {Blue, Thick,
     Dashed}}], 1.5]
```

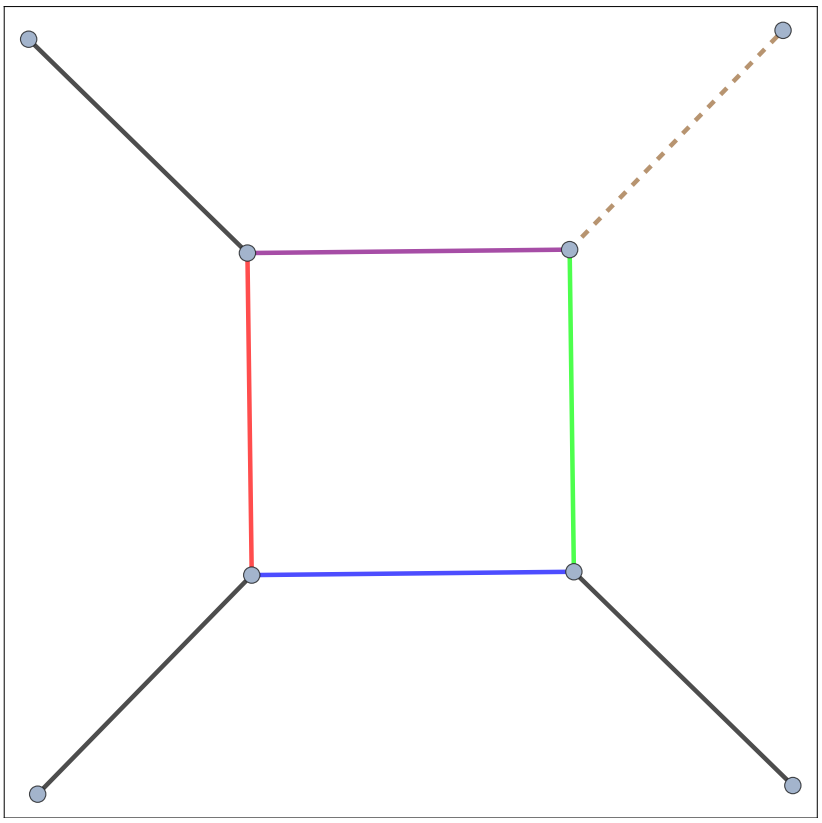
$$\left\{\{-3\to 2,-1\to 1,1\to 2,1\to 2,1\to 2,1\to 2\},\{q,q,\{\text{k2},2,0\},\{\text{k1}-q,\right.$$
$$1,0\},\{\text{k2}-\text{k3},1,-\text{mc}^2\},\{\text{k1}-\text{k3},1,-\text{mc}^2\}\},\left\{0,0,\frac{1}{(\text{k2}^2+i\eta)},\right.$$
$$\left.\left.\frac{1}{((\text{k1}-q)^2+i\eta)},\frac{1}{((\text{k2}-\text{k3})^2-\text{mc}^2+i\eta)},\frac{1}{((\text{k1}-\text{k3})^2-\text{mc}^2+i\eta)}\right\},1\right\}$$



We can style a fully massive 1-loop box in a very creative way

```
FCLoopIntegralToGraph[FAD[{p, m1}, {p + q1, m2}, {p + q1 + q2, m3}, {p + q1
+ q2 + q3, m4}], {p}]
FCLoopGraphPlot[%, GraphPlot -> {MultiedgeStyle -> 0.35, Frame -> True},
Style -> {
    {"InternalLine", _, _, mm_ /; ! FreeQ[mm, m1]} -> {Red, Thick},
    {"InternalLine", _, _, mm_ /; ! FreeQ[mm, m2]} -> {Blue, Thick},
    {"InternalLine", _, _, mm_ /; ! FreeQ[mm, m3]} -> {Green, Thick},
    {"InternalLine", _, _, mm_ /; ! FreeQ[mm, m4]} -> {Purple, Thick},
    {"ExternalLine", q1} -> {Brown, Thick, Dashed}
   }]
```

$$\left\{\{-4\to 4,-3\to 1,-2\to 2,-1\to 3,1\to 2,1\to 4,2\to 3,3\to 4\},\{\text{q1}-\text{q2}-\text{q3},\text{q1},\text{q2},\text{q3},\{p+\text{q1}+\text{q2},1,-\text{m3}^2\},\{p+\text{q1}+\text{q2}+\text{q3},1,-\text{m4}^2\},\{p+\text{q1},1,-\text{m2}^2\},\{p,1,-\text{m1}^2\}\},\left\{0,0,0,0,\frac{1}{(p^2-\text{m1}^2+i\eta)},\frac{1}{((p+\text{q1})^2-\text{m2}^2+i\eta)},\frac{1}{((p+\text{q1}+\text{q2})^2-\text{m3}^2+i\eta)},\frac{1}{((p+\text{q1}+\text{q2}+\text{q3})^2-\text{m4}^2+i\eta)}\right\},1\right\}$$

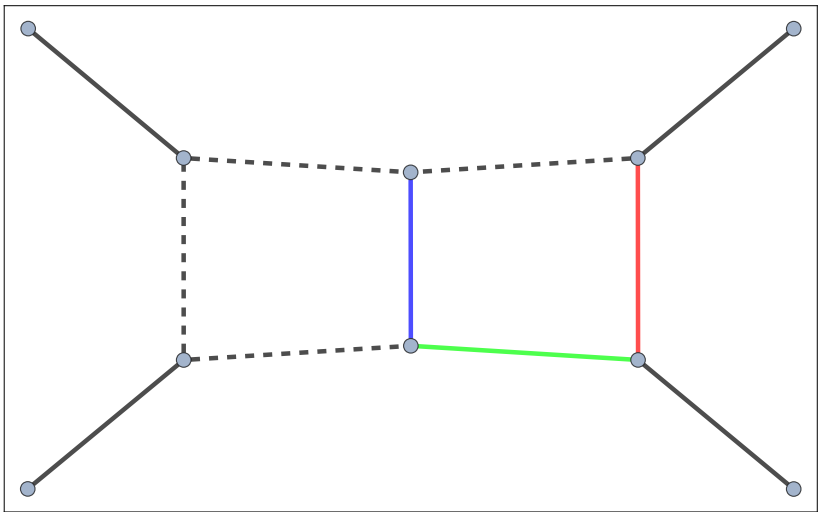


The same goes for a 2-loop box with 3 massive lines

```
FCLoopIntegralToGraph[FAD[{p1, m1}, {p2, m2}, {Q1 + p1, m3}, Q2 - p1, Q1 +
p1 + p2, Q2 - p1 - p2,
    Q2 + Q3 - p1 - p2], {p1, p2}]

FCLoopGraphPlot[%, GraphPlot -> {MultiedgeStyle -> 0.35, Frame -> True},
Style -> {
    {"InternalLine", _, _, mm_ /; ! FreeQ[mm, m1]} -> {Red, Thick},
    {"InternalLine", _, _, mm_ /; ! FreeQ[mm, m2]} -> {Blue, Thick},
    {"InternalLine", _, _, mm_ /; ! FreeQ[mm, m3]} -> {Green, Thick},
    {"InternalLine", _, _, mm_ /; ! FreeQ[mm, m4]} -> {Purple, Thick},
    {"ExternalLine", q1} -> {Brown, Thick, Dashed}
   }]
```

$$\Big\{\{-4\to 4,-3\to 1,-2\to 2,-1\to 3,1\to 4,1\to 6,2\to 3,2\to 6,3\to 5,4\to 5,5\to 6\},$$
$$\{\text{Q1}-\text{Q2}-\text{Q3},\text{Q1},\text{Q2},\text{Q3},\{-\text{p1}-\text{p2}+\text{Q2}+\text{Q3},1,0\},\{-\text{p1}-\text{p2}+\text{Q2},1,0\},\{\text{p1},1,$$
$$-\text{m1}^2\},\{\text{Q2}-\text{p1},1,0\},\{\text{p1}+\text{Q1},1,-\text{m3}^2\},\{\text{p1}+\text{p2}+\text{Q1},1,0\},\{\text{p2},1,-\text{m2}^2\}\},\Big\{0,$$
$$0,0,0,\frac{1}{((\text{p1}+\text{p2}+\text{Q1})^2+i\eta)},\frac{1}{((\text{Q2}-\text{p1})^2+i\eta)},\frac{1}{(\text{p2}^2-\text{m2}^2+i\eta)},\frac{1}{(\text{p1}^2-\text{m1}^2+i\eta)},$$
$$\frac{1}{((\text{p1}+\text{Q1})^2-\text{m3}^2+i\eta)},\frac{1}{((-\text{p1}-\text{p2}+\text{Q2})^2+i\eta)},\frac{1}{((-\text{p1}-\text{p2}+\text{Q2}+\text{Q3})^2+i\eta)}\Big\},1\Big\}$$

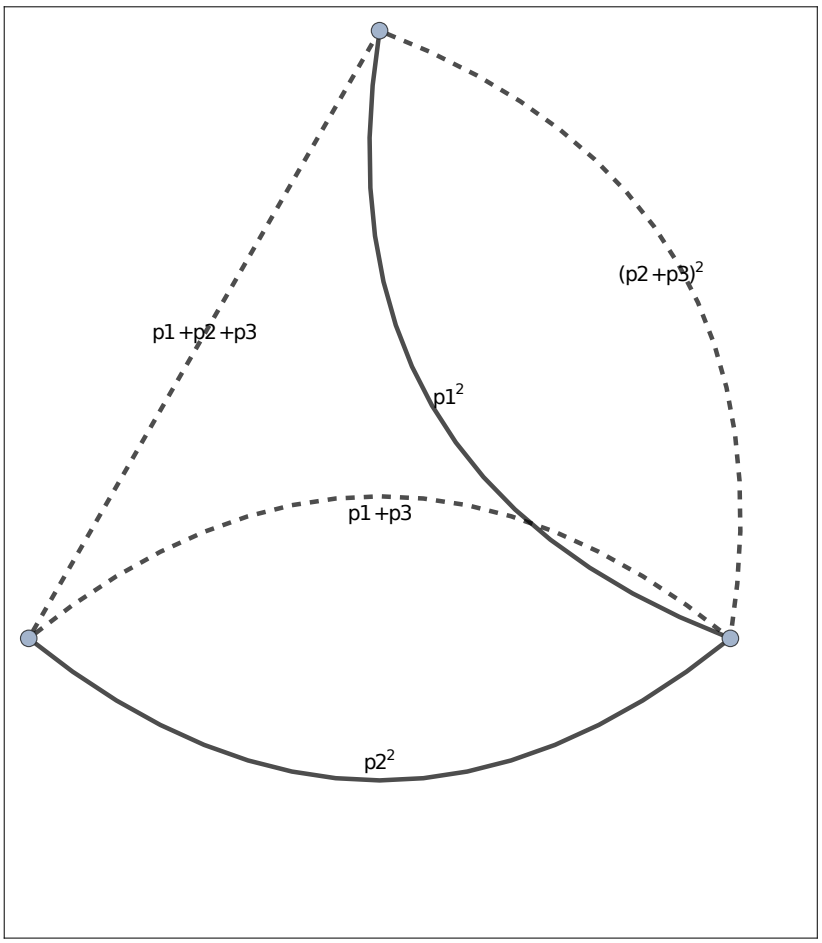


One can also (sort of) visualize the momentum flow, where we use powers to denote the dots

```
FCLoopIntegralToGraph[FCTopology[topo1X1, {SFAD[{{p2, 0}, {m1^2, 1}, 2}],
     SFAD[{{p1, 0}, {m1^2, 1}, 2}],
     SFAD[{{p2 + p3, 0}, {0, 1}, 1}], SFAD[{{p2 + p3, 0}, {0, 1}, 1}],
     SFAD[{{p1 + p3, 0}, {0, 1}, 1}], SFAD[{{p1 + p2 + p3, 0}, {0, 1},
1}]}, {p1, p2, p3}, {}, {}, {}]]

FCLoopGraphPlot[%, GraphPlot -> {MultiedgeStyle -> 0.35, Frame -> True},
Labeled -> {
    {"InternalLine", x_, pow_, _} :> x^pow,
    {"ExternalLine", _} :> {}}]
```

$$\Bigg\{\{1\to 2,1\to 3,1\to 3,2\to 3,2\to 3\},\begin{pmatrix}\text{p1}+\text{p2}+\text{p3} & 1 & 0\\ \text{p1}+\text{p3} & 1 & 0\\ \text{p2} & 2 & -\text{m1}^2\\ \text{p2}+\text{p3} & 2 & 0\\ \text{p1} & 2 & -\text{m1}^2\end{pmatrix},\Big\{\frac{1}{((\text{p1}+\text{p3})^2+i\eta)},$$
$$\frac{1}{((\text{p2}+\text{p3})^2+i\eta)},\frac{1}{((\text{p1}+\text{p2}+\text{p3})^2+i\eta)},\frac{1}{(\text{p1}^2-\text{m1}^2+i\eta)},\frac{1}{(\text{p2}^2-\text{m1}^2+i\eta)}\Big\},1\Bigg\}$$



9.32 FCLoopIntegralToGraph

FCLoopIntegralToGraph[int, {q1, q2, ...}] constructs a graph representation of the loop integral **int** that depends on the loop momenta **q1, q2, ...**. The function returns a list of the form **{ edges, labels, props, pref}**, where **edges** is a list of edge rules representing the loop integral **int**, **labels** is a list of lists containing the line momentum, multiplicity and the mass term of each propagator, **props** is a list with the original propagators and **pref** is the piece of the integral that was ignored when constructing the graph representation (e.g. scalar products or vectors in the numerator) .

Use **FCLoopGraphPlot** to visualize the output of **FCLoopIntegralToGraph**.

A quick and simple way to plot the graph is to evaluate **GraphPlot[List @@@ Transpose[output[[1 ;; 2]]]]** or **GraphPlot[Labeled @@@ Transpose[output[[1 ;; 2]]]]**. The visual quality will not be that great, though. To obtain a nicer plot one might use **GraphPlot** with a custom **EdgeTaggedGraph** or export the output to a file and visualize it with an external tool such as dot/neato from graphviz.

It is also possible to invoke the function as **FCLoopIntegralToGraph[GLI[...], FCTopology[...]]** or **FCLoopIntegralToGraph[FCTopology[...]]**.

9.32.1 See also

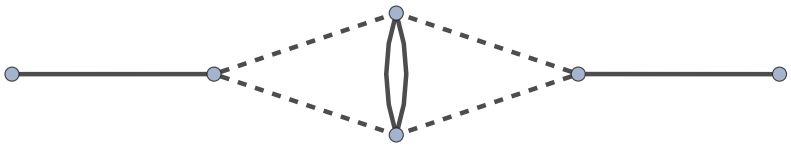
Overview, FCLoopGraphPlot.

9.32.2 Examples

```
out = FCLoopIntegralToGraph[FAD[{q - k1}, k1, q - k2, k2, {k2 - k3, mb},
{k1 - k3, mb}], {k1, k2, k3}]
```

$$\left\{\{-3\to 2,-1\to 1,1\to 3,1\to 4,2\to 3,2\to 4,3\to 4,3\to 4\},\{q,q,\{\text{k2},1,0\},\{q-\text{k2},1,0\},\right.$$
$$\{\text{k1},1,0\},\{q-\text{k1},1,0\},\left\{\text{k2}-\text{k3},1,-\text{mb}^2\right\},\left\{\text{k1}-\text{k3},1,-\text{mb}^2\right\}\},\left\{0,0,\frac{1}{(\text{k2}^2+i\eta)},\frac{1}{(\text{k1}^2+i\eta)},\right.$$
$$\left.\left.\frac{1}{((q-\text{k2})^2+i\eta)},\frac{1}{((q-\text{k1})^2+i\eta)},\frac{1}{((\text{k2}-\text{k3})^2-\text{mb}^2+i\eta)},\frac{1}{((\text{k1}-\text{k3})^2-\text{mb}^2+i\eta)}\right\},1\right\}$$

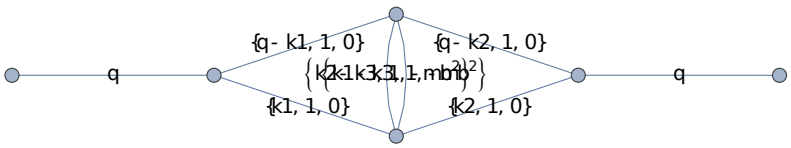
```
FCLoopGraphPlot[out]
```



```
Labeled @@@ Transpose[out[[1 ;; 2]]]
```

$$\left\{\begin{matrix}-3\to 2, & -1\to 1, & 1\to 3, & 1\to 4, \\ q & q & \{\text{k2},1,0\} & \{q-\text{k2},1,0\}\end{matrix}\right.$$
$$\left.\begin{matrix}2\to 3, & 2\to 4, & 3\to 4, & 3\to 4 \\ \{\text{k1},1,0\} & \{q-\text{k1},1,0\} & \left\{\text{k2}-\text{k3},1,-\text{mb}^2\right\} & \left\{\text{k1}-\text{k3},1,-\text{mb}^2\right\}\end{matrix}\right\}$$

```
GraphPlot[List @@@ Transpose[out[[1 ;; 2]]]]
```




```
FCLoopIntegralToGraph[FAD[{q - k1}, k1, q - k2, k2, {k2 - k3, mb}, {k1 -
k3, mb}], {k1, k2,k3}]
```

$$\left\{\{-3\to 2,-1\to 1,1\to 3,1\to 4,2\to 3,2\to 4,3\to 4,3\to 4\},\{q,q,\{\text{k2},1,0\},\{q-\text{k2},1,0\},\right.$$
$$\{\text{k1},1,0\},\{q-\text{k1},1,0\},\left\{\text{k2}-\text{k3},1,-\text{mb}^2\right\},\left\{\text{k1}-\text{k3},1,-\text{mb}^2\right\}\},\left\{0,0,\frac{1}{(\text{k2}^2+i\eta)},\frac{1}{(\text{k1}^2+i\eta)},\right.$$
$$\left.\left.\frac{1}{((q-\text{k2})^2+i\eta)},\frac{1}{((q-\text{k1})^2+i\eta)},\frac{1}{((\text{k2}-\text{k3})^2-\text{mb}^2+i\eta)},\frac{1}{((\text{k1}-\text{k3})^2-\text{mb}^2+i\eta)}\right\},1\right\}$$

```
FAD[q - k1, k1, q - k2, k2, {k2 - k3, mb}, {k1 - k3, mb}]
```

$$\frac{1}{(q-\text{k1})^2.\text{k1}^2.(q-\text{k2})^2.\text{k2}^2.\left((\text{k2}-\text{k3})^2-\text{mb}^2\right).\left((\text{k1}-\text{k3})^2-\text{mb}^2\right)}$$

```
FCLoopIntegralToGraph[FCTopology[topo1, {FAD[q - k1], FAD[k1], FAD[q - k2],
FAD[k2],
    FAD[{k2 - k3, mb}], FAD[{k1 - k3, mb}]}, {k1, k2, k3}, {q}, {}, {}]]
```

$$\left\{\{-3\to 2,-1\to 1,1\to 3,1\to 4,2\to 3,2\to 4,3\to 4,3\to 4\},\{q,q,\{\text{k2},1,0\},\{q-\text{k2},1,0\},\right.$$
$$\{\text{k1},1,0\},\{q-\text{k1},1,0\},\left\{\text{k1}-\text{k3},1,-\text{mb}^2\right\},\left\{\text{k2}-\text{k3},1,-\text{mb}^2\right\}\},\left\{0,0,\frac{1}{(\text{k2}^2+i\eta)},\frac{1}{(\text{k1}^2+i\eta)},\right.$$
$$\left.\left.\frac{1}{((q-\text{k2})^2+i\eta)},\frac{1}{((q-\text{k1})^2+i\eta)},\frac{1}{((\text{k1}-\text{k3})^2-\text{mb}^2+i\eta)},\frac{1}{((\text{k2}-\text{k3})^2-\text{mb}^2+i\eta)}\right\},1\right\}$$

```
FCLoopIntegralToGraph[GLI[topo1, {1, 1, 1, 1, 1, 1}],
  FCTopology[topo1, {FAD[q - k1], FAD[k1], FAD[q - k2], FAD[k2],
    FAD[{k2 - k3, mb}], FAD[{k1 - k3, mb}]}, {k1, k2, k3}, {q}, {}, {}]]
```

$$\left\{\{-3\to 2,-1\to 1,1\to 3,1\to 4,2\to 3,2\to 4,3\to 4,3\to 4\},\{q,q,\{\text{k2},1,0\},\{q-\text{k2},1,0\},\right.$$
$$\{\text{k1},1,0\},\{q-\text{k1},1,0\},\left\{\text{k2}-\text{k3},1,-\text{mb}^2\right\},\left\{\text{k1}-\text{k3},1,-\text{mb}^2\right\}\},\left\{0,0,\frac{1}{(\text{k2}^2+i\eta)},\frac{1}{(\text{k1}^2+i\eta)},\right.$$
$$\left.\left.\frac{1}{((q-\text{k2})^2+i\eta)},\frac{1}{((q-\text{k1})^2+i\eta)},\frac{1}{((\text{k2}-\text{k3})^2-\text{mb}^2+i\eta)},\frac{1}{((\text{k1}-\text{k3})^2-\text{mb}^2+i\eta)}\right\},1\right\}$$

If the second argument contains multiple topologies, the function will automatically select the relevant ones.

```
FCLoopIntegralToGraph[GLI[topo1, {1, 1, 1, 0, 0, 0}],
  {FCTopology[topo1, {FAD[q - k1], FAD[k1], FAD[q - k2], FAD[k2],
     FAD[{k2 - k3, mb}], FAD[{k1 - k3, mb}]}, {k1, k2, k3}, {q}, {}, {}],
   FCTopology[topo2, {FAD[q - k1], FAD[k1], FAD[q - k2], FAD[k2],
     FAD[{k2 - k3, mg}], FAD[{k1 - k3, mg}]}, {k1, k2, k3}, {q}, {}, {}]
  }]
```

$$\left\{\{-3\to 2,-1\to 1,1\to 2,1\to 2,2\to 2\},\{q,q,\{\text{k1},1,0\},\{q-\text{k1},1,0\},\right.$$
$$\left.\{q-\text{k2},1,0\}\},\left\{0,0,\frac{1}{(\text{k1}^2+i\eta)},\frac{1}{((q-\text{k2})^2+i\eta)},\frac{1}{((q-\text{k1})^2+i\eta)}\right\},1\right\}$$

9.33 FCLoopPropagatorsToLineMomenta

`FCLoopPropagatorsToLineMomenta[{prop1, prop2, ...}]` is an auxiliary function that extracts line momenta flowing through the given list of propagators.

9.33.1 See also

Overview, FCLoopIntegralToGraph, AuxiliaryMomenta.

9.33.2 Examples

```
FCLoopPropagatorsToLineMomenta[{SFAD[{q + l, m^2}], SFAD[{p, -m^2}]}, FCE
-> True]
```

$$\begin{pmatrix} l+q & p \\ -m^2 & m^2 \\ \frac{1}{((l+q)^2-m^2+i\eta)} & \frac{1}{(p^2+m^2+i\eta)} \end{pmatrix}$$

```
FCLoopPropagatorsToLineMomenta[{CFAD[{{0, 2 v . (q + r)}, m^2}]}, FCE ->
True,
  AuxiliaryMomenta -> {v}]
```

$$\begin{pmatrix} q+r \\ m^2 \\ \frac{1}{(2((q+r)\cdot v)+m^2-i\eta)} \end{pmatrix}$$

Reversed signs are also supported

```
{SFAD[{I (q + l), -m^2}], SFAD[{I p, -m^2}]}

FCLoopPropagatorsToLineMomenta[%, FCE -> True]
```

$$\left\{\frac{1}{(-(l+q)^2+m^2+i\eta)},\frac{1}{(-p^2+m^2+i\eta)}\right\}$$

$$\begin{pmatrix} l+q & p \\ m^2 & m^2 \\ \frac{1}{(-(l+q)^2+m^2+i\eta)} & \frac{1}{(-p^2+m^2+i\eta)} \end{pmatrix}$$

```
FCLoopPropagatorsToLineMomenta[{SFAD[{I (q + l), -m^2}], SFAD[{I p,
-m^2}]},
   FCE -> True] // InputForm
```

```
{{l + q, p}, {m^2, m^2}, {SFAD[{{I*(l + q), 0}, {-m^2, 1}, 1}],
  SFAD[{{I*p, 0}, {-m^2, 1}, 1}]}}
```

```
FCLoopPropagatorsToLineMomenta[{SFAD[{{I p1, -2 p1 . q}, {0, 1}, 1}]},FCE
-> True]
```

$$\begin{pmatrix} \text{p1}+q \\ 0 \\ \frac{1}{(-\text{p1}^2-2(\text{p1}\cdot q)+i\eta)} \end{pmatrix}$$

9.34 FCLoopApplyTopologyMappings

FCLoopApplyTopologyMappings[expr, {mappings, topos}] applies mappings between topologies obtained using **FCLoopFindTopologyMappings** to the output of **FCLoopFindTopologies** denoted as **expr**. The argument **topos** denotes the final set of topologies present in the expression.

Instead of **{mappings, topos}** one can directly use the output **FCLoopFindTopologyMappings**.

By default the function will attempt to rewrite all the occurring loop integrals as **GLI**s. If you just want to apply the mappings without touching the remaining scalar products, set the option **FCLoopCreateRulesToGLI** to **False**. Even when all scalar products depending on loop momenta are rewritten as **GLI**s, you can still suppress the step of multiplying out products of **GLI**s by setting the option **GLIMultiply** to **False**.

9.34.1 See also

Overview, FCTopology, GLI, FCLoopFindTopologies, FCLoopFindTopologyMappings.

9.34.2 Examples

This is a trial expression representing some loop amplitude that has already been processed using **FCFind-Topologies**

```
ex = gliProduct[cc6*SPD[p1, p1], GLI[fctopology1, {1, 1, 2, 1, 1, 1, 1, 1,
1}]] +
   gliProduct[cc2*SPD[p1, p2], GLI[fctopology2, {1, 1, 1, 1, 1, 1, 1, 1,
1}]] +
   gliProduct[cc4*SPD[p1, p2], GLI[fctopology4, {1, 1, 1, 1, 1, 1, 1, 1,
1}]] +
   gliProduct[cc1*SPD[p1, Q], GLI[fctopology1, {1, 1, 1, 1, 1, 1, 1, 1,
1}]] +
   gliProduct[cc3*SPD[p2, p2], GLI[fctopology3, {1, 1, 1, 1, 1, 1, 1, 1,
1}]] +
   gliProduct[cc5*SPD[p2, Q], GLI[fctopology5, {1, 1, 1, 1, 1, 1, 1, 1,
1}]]
```

$$\text{gliProduct}\left(\text{cc1}(\text{p1}\cdot Q), G^{\text{fctopology1}}(1,1,1,1,1,1,1,1,1)\right) + \text{gliProduct}\left(\text{cc2}(\text{p1}\cdot\ \text{p2}), G^{\text{fctopology2}}(1,1,1,1,1,1,1,1,1)\right) + \text{gliProduct}\left(\text{cc3}\ \text{p2}^2, G^{\text{fctopology3}}(1,1,1,1,1,1,1,1,1)\right) + \text{gliProduct}\left(\text{cc4}(\text{p1}\cdot\ \text{p2}), G^{\text{fctopology4}}(1,1,1,1,1,1,1,1,1)\right) + \text{gliProduct}\left(\text{cc5}(\text{p2}\cdot Q), G^{\text{fctopology5}}(1,1,1,1,1,1,1,1,1)\right) + \text{gliProduct}\left(\text{cc6}\ \text{p1}^2, G^{\text{fctopology1}}(1,1,2,1,1,1,1,1,1)\right)$$

These mapping rules describe how the 3 topologies “fctopology3”, “fctopology4” and “fctopology5” are mapped to the topologies “fctopology1” and “fctopology2”

```
mappings = {
   {FCTopology[fctopology3, {SFAD[{{p3, 0}, {0, 1}, 1}], SFAD[{{p2, 0}, {0,
   1}, 1}],
      SFAD[{{p1, 0}, {0, 1}, 1}], SFAD[{{p2 + p3, 0}, {0, 1}, 1}],
SFAD[{{p1 + p3, 0}, {0, 1}, 1}],
      SFAD[{{p2 - Q, 0}, {0, 1}, 1}], SFAD[{{p2 + p3 - Q, 0}, {0, 1}, 1}],
SFAD[{{p1 + p3 - Q, 0}, {0, 1}, 1}],
      SFAD[{{p1 + p2 + p3 - Q, 0}, {0, 1}, 1}]}, {p1, p2, p3}, {Q}, {},
{}], {p1 -> -p1 - p3 + Q, p2 -> -p2 - p3 + Q, p3 -> p3},
    GLI[fctopology3, {n1_, n7_, n8_, n5_, n6_, n4_, n2_, n3_, n9_}] :>
     GLI[fctopology1, {n1, n2, n3, n4, n5, n6, n7, n8, n9}]},

   {FCTopology[fctopology4, {SFAD[{{p3, 0}, {0, 1}, 1}], SFAD[{{p2, 0}, {0,
   1}, 1}], SFAD[{{p1, 0}, {0, 1}, 1}],
      SFAD[{{p2 + p3, 0}, {0, 1}, 1}], SFAD[{{p1 + p3, 0}, {0, 1}, 1}],
SFAD[{{p2 - Q, 0}, {0, 1}, 1}], SFAD[{{p1 - Q, 0}, {0, 1}, 1}],
      SFAD[{{p1 + p3 - Q, 0}, {0, 1}, 1}], SFAD[{{p1 + p2 + p3 - Q, 0}, {0,
1}, 1}]}, {p1, p2, p3}, {Q}, {}, {}], {p1 -> -p2 + Q, p2 -> -p1 + Q, p3 ->
-p3},
    GLI[fctopology4, {n1_, n6_, n5_, n8_, n7_, n3_, n2_, n4_, n9_}] :>
     GLI[fctopology1, {n1, n2, n3, n4, n5, n6, n7, n8, n9}]},

   {FCTopology[fctopology5, {SFAD[{{p3, 0}, {0, 1}, 1}], SFAD[{{p2, 0}, {0,
   1}, 1}], SFAD[{{p1, 0}, {0, 1}, 1}],
```

```
      SFAD[{{p1 + p3, 0}, {0, 1}, 1}], SFAD[{{p2 - Q, 0}, {0, 1},
1}],SFAD[{{p1 - Q, 0}, {0, 1}, 1}], SFAD[{{p1 + p3 - Q, 0}, {0, 1}, 1}],
      SFAD[{{p1 + p2 - Q, 0}, {0, 1}, 1}], SFAD[{{p1 + p2 + p3 - Q, 0}, {0,
1}, 1}]}, {p1, p2, p3}, {Q}, {}, {}], {p1 -> p2, p2 -> p1, p3 -> p3},
    GLI[fctopology5, {n1_, n3_, n2_, n4_, n6_, n5_, n7_, n8_, n9_}] :>
     GLI[fctopology2, {n1, n2, n3, n4, n5, n6, n7, n8, n9}]}}
```

$$\left(\begin{array}{l}\text{FCTopology}\left(\text{fctopology3}, \left\{\frac{1}{(\text{p3}^2+i\eta)}, \frac{1}{(\text{p2}^2+i\eta)}, \frac{1}{(\text{p1}^2+i\eta)}, \frac{1}{((\text{p2}+\text{p3})^2+i\eta)}, \frac{1}{((\text{p1}+\text{p3})^2+i\eta)}, \frac{1}{((\text{p2}-Q)^2+i\eta)}, \frac{1}{((\text{p2}+\text{p3}-Q)}\right.\right. \\ \text{FCTopology}\left(\text{fctopology4}, \left\{\frac{1}{(\text{p3}^2+i\eta)}, \frac{1}{(\text{p2}^2+i\eta)}, \frac{1}{(\text{p1}^2+i\eta)}, \frac{1}{((\text{p2}+\text{p3})^2+i\eta)}, \frac{1}{((\text{p1}+\text{p3})^2+i\eta)}, \frac{1}{((\text{p2}-Q)^2+i\eta)}, \frac{1}{((\text{p1}-Q)^2+}\right.\right. \\ \text{FCTopology}\left(\text{fctopology5}, \left\{\frac{1}{(\text{p3}^2+i\eta)}, \frac{1}{(\text{p2}^2+i\eta)}, \frac{1}{(\text{p1}^2+i\eta)}, \frac{1}{((\text{p1}+\text{p3})^2+i\eta)}, \frac{1}{((\text{p2}-Q)^2+i\eta)}, \frac{1}{((\text{p1}-Q)^2+i\eta)}, \frac{1}{((\text{p1}+\text{p3}-Q)^2}\right.\right.\end{array}\right.$$

These are the two topologies onto which everything is mapped

```
finalTopos = {
   FCTopology[fctopology1, {SFAD[{{p3, 0}, {0, 1}, 1}], SFAD[{{p2, 0}, {0,
1}, 1}], SFAD[{{p1, 0}, {0, 1}, 1}], SFAD[{{p2 + p3, 0}, {0, 1}, 1}],
SFAD[{{p2 - Q, 0}, {0, 1}, 1}],
     SFAD[{{p1 - Q, 0}, {0, 1}, 1}], SFAD[{{p2 + p3 - Q, 0}, {0, 1}, 1}],
SFAD[{{p1 + p3 - Q, 0}, {0, 1}, 1}], SFAD[{{p1 + p2 + p3 - Q, 0}, {0, 1},
1}]}, {p1, p2, p3}, {Q}, {}, {}],
   FCTopology[fctopology2, {SFAD[{{p3, 0}, {0, 1}, 1}], SFAD[{{p2, 0}, {0,
1}, 1}], SFAD[{{p1, 0}, {0, 1}, 1}], SFAD[{{p2 + p3, 0}, {0, 1}, 1}],
SFAD[{{p2 - Q, 0}, {0, 1}, 1}],
     SFAD[{{p1 - Q, 0}, {0, 1}, 1}], SFAD[{{p2 + p3 - Q, 0}, {0, 1}, 1}],
SFAD[{{p1 + p2 - Q, 0}, {0, 1}, 1}], SFAD[{{p1 + p2 + p3 - Q, 0}, {0, 1},
1}]}, {p1, p2, p3}, {Q}, {}, {}]}
```

$$\left\{\text{FCTopology}\left(\text{fctopology1}, \left\{\frac{1}{(\text{p3}^2+i\eta)}, \frac{1}{(\text{p2}^2+i\eta)}, \frac{1}{(\text{p1}^2+i\eta)}, \right.\right.\right.$$
$$\left.\frac{1}{((\text{p2}+\text{p3})^2+i\eta)}, \frac{1}{((\text{p2}-Q)^2+i\eta)}, \frac{1}{((\text{p1}-Q)^2+i\eta)}, \frac{1}{((\text{p2}+\text{p3}-Q)^2+i\eta)}, \right.$$
$$\left.\left.\frac{1}{((\text{p1}+\text{p3}-Q)^2+i\eta)}, \frac{1}{((\text{p1}+\text{p2}+\text{p3}-Q)^2+i\eta)}\right\}, \{\text{p1}, \text{p2}, \text{p3}\}, \{Q\}, \right.$$
$$\left.\{\}, \{\}\right), \text{FCTopology}\left(\text{fctopology2}, \left\{\frac{1}{(\text{p3}^2+i\eta)}, \frac{1}{(\text{p2}^2+i\eta)}, \frac{1}{(\text{p1}^2+i\eta)}, \right.\right.$$
$$\left.\frac{1}{((\text{p2}+\text{p3})^2+i\eta)}, \frac{1}{((\text{p2}-Q)^2+i\eta)}, \frac{1}{((\text{p1}-Q)^2+i\eta)}, \frac{1}{((\text{p2}+\text{p3}-Q)^2+i\eta)}, \right.$$
$$\left.\left.\left.\frac{1}{((\text{p1}+\text{p2}-Q)^2+i\eta)}, \frac{1}{((\text{p1}+\text{p2}+\text{p3}-Q)^2+i\eta)}\right\}, \{\text{p1}, \text{p2}, \text{p3}\}, \{Q\}, \{\}, \{\}\right)\right\}$$

`FCLoopApplyTopologyMappings` applies the given mappings to the expression creating an output that is ready to be processed further

```
FCLoopApplyTopologyMappings[ex, {mappings, finalTopos}, Head -> gliProduct,
FCVerbose -> 0]
```

$$\frac{1}{2}G^{\text{fctopology1}}(1,1,1,1,1,1,1,1,1)\left(\text{cc1}Q^2+\text{cc4}Q^2+2\,\text{cc6}\right)+\frac{1}{2}(\text{cc1}-\text{cc4})G^{\text{fctopology1}}(1,1,0,1,1,1,1,1,1)-\frac{1}{2}(\text{cc1}-\text{cc4})G^{\text{fctopology1}}(1,1,1,1,1,0,1,1,1)+\frac{1}{2}Q^2(\text{cc2}+\text{cc5})G^{\text{fctopology2}}(1,1,1,1,1,1,1,1,1)-\frac{1}{2}(\text{cc2}+\text{cc5})G^{\text{fctopology2}}(1,1,1,1,0,1,1,1,1)-\frac{1}{2}\,\text{cc2}G^{\text{fctopology2}}(1,1,1,1,0,1,1,1,1)+\frac{1}{2}\,\text{cc2}G^{\text{fctopology2}}(1,1,1,1,1,1,0,1,1)+\text{cc3}G^{\text{fctopology1}}(1,1,1,1,1,1,0,1,1)+\frac{1}{2}\,\text{cc4}G^{\text{fctopology1}}(0,1,1,1,1,1,1,1,1)-\frac{1}{2}\,\text{cc4}G^{\text{fctopology1}}(1,1,1,0,1,1,1,1,1)-\frac{1}{2}\,\text{cc4}G^{\text{fctopology1}}(1,1,1,1,1,1,1,0,1)+\frac{1}{2}\,\text{cc4}G^{\text{fctopology1}}(1,1,1,1,1,1,1,1,0)+\frac{1}{2}\,\text{cc5}G^{\text{fctopology2}}(1,1,0,1,1,1,1,1,1)$$

This just applies the mappings without any further simplifications

```
FCLoopApplyTopologyMappings[ex, {mappings, finalTopos}, Head -> gliProduct,
FCLoopCreateRulesToGLI -> False]
```

$$\text{gliProduct}\left(\text{cc3}(-\text{p2}-\text{p3}+Q)^2,G^{\text{fctopology1}}(1,1,1,1,1,1,1,1,1)\right)+\text{gliProduct}\left(\text{cc4}((Q-\text{p1})\cdot(Q-\text{p2})),G^{\text{fctopology1}}(1,1,1,1,1,1,1,1,1)\right)+\text{gliProduct}\left(\text{cc1}(\text{p1}\cdot Q),G^{\text{fctopology1}}(1,1,1,1,1,1,1,1,1)\right)+\text{gliProduct}\left(\text{cc2}(\text{p1}\cdot\ \text{p2}),G^{\text{fctopology2}}(1,1,1,1,1,1,1,1,1)\right)+\text{gliProduct}\left(\text{cc5}(\text{p1}\cdot Q),G^{\text{fctopology2}}(1,1,1,1,1,1,1,1,1)\right)+\text{gliProduct}\left(\text{cc6}\ \text{p1}^2,G^{\text{fctopology1}}(1,1,2,1,1,1,1,1,1)\right)$$

This applies the mappings and eliminates the numerators but still keeps products of **GLI**s in the expression

```
FCLoopApplyTopologyMappings[ex, {mappings, finalTopos}, Head -> gliProduct,
FCLoopCreateRulesToGLI -> True, GLIMultiply -> False]
```

$$\text{gliProduct}\left(\frac{1}{2}\,\text{cc1}\left(G^{\text{fctopology1}}(0,0,-1,0,0,0,0,0,0)-G^{\text{fctopology1}}(0,0,0,0,0,-1,0,0,0)+Q^2\right),G^{\text{fctopology1}}(1,1,1,1,1,1,1,1,1)\right)+\text{gliProduct}\left(\frac{1}{2}\,\text{cc2}\left(-G^{\text{fctopology2}}(0,0,0,0,-1,0,0,0,0)-G^{\text{fctopology2}}(0,0,0,0,0,0,0,-1,0)+Q^2\right),G^{\text{fctopology2}}(1,1,1,1,1,1,1,1,1)\right)+\text{gliProduct}\left(\text{cc3}G^{\text{fctopology1}}(0,0,0,0,0,0,-1,0,0),G^{\text{fctopology1}}(1,1,1,1,1,1,1,1,1)\right)+\text{gliProduct}\left(\text{cc4}\left(\frac{1}{2}\left(-G^{\text{fctopology1}}(0,-1,0,0,0,0,0,0,0)+G^{\text{fctopology1}}(0,0,0,0,-1,0,0,0,0)-Q^2\right)+\frac{1}{2}\left(-G^{\text{fctopology1}}(0,0,-1,0,0,0,0,0,0)+G^{\text{fctopology1}}(0,0,0,0,0,-1,0,0,0)-Q^2\right)+\frac{1}{2}\left(G^{\text{fctopology1}}(-1,0,0,0,0,0,0,0,0)+G^{\text{fctopology1}}(0,-1,0,0,0,0,0,0,0)-G^{\text{fctopology1}}(0,0,0,-1,0,0,0,0,0)-G^{\text{fctopology1}}(0,0,0,0,-1,0,0,0,0)-G^{\text{fctopology1}}(0,0,0,0,0,0,-1,0,0)+G^{\text{fctopology1}}(0,0,0,0,0,0,0,0,-1)+Q^2\right)+Q^2\right),G^{\text{fctopology1}}(1,1,1,1,1,1,1,1,1)\right)+\text{gliProduct}\left(\frac{1}{2}\,\text{cc5}\left(G^{\text{fctopology2}}(0,0,-1,0,0,0,0,0,0)-G^{\text{fctopology2}}(0,0,0,0,0,-1,0,0,0)+Q^2\right),G^{\text{fctopology2}}(1,1,1,1,1,1,1,1,1)\right)+\text{gliProduct}\left(\text{cc6}G^{\text{fctopology1}}(0,0,-1,0,0,0,0,0,0),G^{\text{fctopology1}}(1,1,2,1,1,1,1,1,1)\right)$$

9.35 FCLoopCreateRuleGLIToGLI

`FCLoopCreateRuleGLIToGLI[topology1, topology2]` creates a GLI replacement rule assuming that the **`topology2`** is a subtopology of **`topology1`**. Both topologies must be given as **`FCTopology`** objects.

It is also possible to use **`FCLoopCreateRuleGLIToGLI[topo1, {subtopo1, subtopo2, ...}]`** provided that **`{subtopo1, subtopo2, ...}`** are subtopologies of **`topo1`** that were obtained by removing some propagators from **`topo1`** and not performing any loop momentum shifts afterwards.

Furthermore, when working with lists of topologies one can write **`FCLoopCreateRuleGLIToGLI[{topo1, topo2, ...}, {{subtopo11, subtopo12, ...}, {subtopo21, subtopo22, ...}, ..}]`**.

9.35.1 See also

Overview, FCTopology, GLI, FCLoopFindTopologies, FCLoopFindTopologyMappings.

9.35.2 Examples

```
FCLoopCreateRuleGLIToGLI[FCTopology[topo1, {SFAD[p]}], FCTopology[topo2,
{SFAD[p]}]]
```

$$G^{\text{topo2}}(\text{n1\_}) : \\ \to G^{\text{topo1}}(\text{n1})$$

```
FCLoopCreateRuleGLIToGLI[FCTopology[topo1, {SFAD[p], SFAD[q]}],
  FCTopology[topo2, {SFAD[p]}]]
```

$$G^{\text{topo2}}(\text{n1\_}) : \\ \to G^{\text{topo1}}(\text{n1}, 0)$$

```
FCLoopCreateRuleGLIToGLI[FCTopology[topo1, {SFAD[p], SFAD[q]}],
  FCTopology[topo2, {SFAD[q], SFAD[p]}]]
```

$$G^{\text{topo2}}(\text{n2\_}, \text{n1\_}) : \\ \to G^{\text{topo1}}(\text{n1}, \text{n2})$$

```
FCLoopCreateRuleGLIToGLI[FCTopology[topo1, {SFAD[r], SFAD[p], SFAD[q]}],
  FCTopology[topo2, {SFAD[p]}]]
```

$$G^{\text{topo2}}(\text{n2\_}) : \\ \to G^{\text{topo1}}(0, \text{n2}, 0)$$

```
FCLoopCreateRuleGLIToGLI[FCTopology["tmpTopo4",
   {SFAD[{{0, (k1 + k2) . nb}, {0, 1}, 1}], SFAD[{{0, (k1 - k3) . n}, {0,
   1}, 1}],
    SFAD[{{0, n . (-k1 - k2 + q)}, {0, 1}, 1}], SFAD[{{0, nb . (-k1 + k3 +
q)}, {0, 1}, 1}],
    SFAD[{{-k1, 0}, {0, 1}, 1}], SFAD[{{k2, 0}, {0, 1}, 1}], SFAD[{{k1 +
k2, 0}, {0, 1}, 1}],
    SFAD[{{-k3, 0}, {0, 1}, 1}], SFAD[{{-k1 + k3, 0}, {0, 1}, 1}],
    SFAD[{{k1 - k3 - q, 0}, {0, 1}, 1}], SFAD[{{k1 + k2 - k3 - q, 0}, {0,
1}, 1}],
    SFAD[{{-k1 - k2 + q, 0}, {0, 1}, 1}]}],

  FCTopology["tmpTopo18", {SFAD[{{0, (k1 + k2) . nb}, {0, 1}, 1}],
    SFAD[{{0, n . (-k1 - k2 + q)}, {0, 1}, 1}], SFAD[{{0, nb . (-k1 + k3 +
q)}, {0, 1}, 1}],
    SFAD[{{-k1, 0}, {0, 1}, 1}], SFAD[{{k2, 0}, {0, 1}, 1}],
    SFAD[{{k1 + k2, 0}, {0, 1}, 1}], SFAD[{{-k3, 0}, {0, 1}, 1}],
    SFAD[{{-k1 + k3, 0}, {0, 1}, 1}], SFAD[{{k1 - k3 - q, 0}, {0, 1}, 1}],
    SFAD[{{k1 + k2 - k3 - q, 0}, {0, 1}, 1}], SFAD[{{-k1 - k2 + q, 0}, {0,
1}, 1}]}]]
```

$$G^{\text{tmpTopo18}}(\text{n1\_}, \text{n3\_}, \text{n4\_},\\ \text{n5\_}, \text{n6\_}, \text{n7\_}, \text{n8\_}, \text{n9\_}, \text{n10\_}, \text{n11\_}, \text{n12\_}) :\\ \to G^{\text{tmpTopo4}}(\text{n1}, 0, \text{n3}, \text{n4}, \text{n5}, \text{n6}, \text{n7}, \text{n8}, \text{n9}, \text{n10}, \text{n11}, \text{n12})$$

```
FCLoopIntegralToGraph[FCTopology["tad2l", {FAD[{p1, m1}], FAD[{p2, m2}],
FAD[{p1 - p2, m3}]},
   {p1, p2}, {}, {}, {}]]
```

$$\left\{\{1 \to 2, 1 \to 2, 1 \to 2\}, \begin{pmatrix} \text{p2} & 1 & -\text{m2}^2 \\ \text{p1} & 1 & -\text{m1}^2 \\ \text{p1} - \text{p2} & 1 & -\text{m3}^2 \end{pmatrix}, \left\{\frac{1}{(\text{p2}^2 - \text{m2}^2 + i\eta)}, \right.\right.\\ \left.\left.\frac{1}{(\text{p1}^2 - \text{m1}^2 + i\eta)}, \frac{1}{((\text{p1} - \text{p2})^2 - \text{m3}^2 + i\eta)}\right\}, 1\right\}$$

```
FCLoopCreateRuleGLIToGLI[
  {FCTopology["prop2l", {FAD[{p1, m1}], FAD[{p2, m2}], FAD[{p1 - q, m3}],
  FAD[{p1 - q, m4}],
     FAD[{p1 - p2, m5}]}, {p1, p2}, {q}, {}, {}],
   FCTopology["tad2l", {FAD[{p1, m1}], FAD[{p2, m2}], FAD[{p1 - p2, m3}]},
{p1, p2}, {}, {}, {}]}, {
   {
    FCTopology["prop2lX1", {FAD[{p2, m2}], FAD[{p1 - q, m3}], FAD[{p1 - q,
m4}], FAD[{p1 - p2, m5}]},
     {p1, p2}, {q}, {}, {}],
    FCTopology["prop2lX5", {FAD[{p1, m1}], FAD[{p2, m2}], FAD[{p1 - q,
m3}], FAD[{p1 - q, m4}]},
     {p1, p2}, {q}, {}, {}]
   },
   {
    FCTopology["tad2lX2", {FAD[{p1, m1}], FAD[{p1 - p2, m3}]}, {p1, p2},
{}, {}, {}],
    FCTopology["tad2lX3", {FAD[{p1, m1}], FAD[{p2, m2}]}, {p1, p2}, {}, {},
{}]
   }
  }]
```

$$\left\{\left\{G^{\text{prop2lX1}}(\text{n2\_}, \text{n3\_}, \text{n4\_}, \text{n5\_}) :\to G^{\text{prop2l}}(0, \text{n2}, \text{n3}, \text{n4}, \text{n5}), G^{\text{prop2lX5}}(\text{n1\_}, \text{n2\_}, \text{n3\_}, \text{n4\_}) :\to G^{\text{prop2l}}(\text{n1},\right.\right.\\ \left.\left.\text{n2}, \text{n3}, \text{n4}, 0)\right\}, \left\{G^{\text{tad2lX2}}(\text{n1\_}, \text{n3\_}) :\to G^{\text{tad2l}}(\text{n1}, 0, \text{n3}), G^{\text{tad2lX3}}(\text{n1\_}, \text{n2\_}) :\to G^{\text{tad2l}}(\text{n1}, \text{n2}, 0)\right\}\right\}$$

`FCLoopFindMomentumShifts[source, target, {p1, p2, ...}]` finds loop momentum shifts that bring loop integrals or topologies in the list **`source`** to the form specified in target. The integrals/topologies in **`intFrom`** and **`intTo`** are assumed to be equivalent and their denominators must be

properly ordered via **FCLoopToPakForm**. Here the loop momenta **p1, p2, ...** belong to the source topologies.

target must be provided as a list of **FeynAmpDenominator** objects, while **intFrom** is a list of such lists.

It is also possible to invoke the function as **FCLoopFindMomentumShifts[{FCTopology[...], FCTopology[...]}, FCTopology[...]]**.

For topologies involving kinematic constraints some mappings may require shifts not only in the loop but also in the external momenta. Such shifts are disabled by default but can be activated by setting the option **Momentum** to **All**.

Normally, **FCLoopFindMomentumShifts** will abort the evaluation if it fails to find any suitable shifts. Setting the option **Abort** to **False** will force the function to merely return an empty list in such situations.

9.35.3 See also

Overview, FCLoopToPakForm, FCLoopPakOrder.

9.35.4 Examples

```
source = {{FAD[p4], FAD[p1], FAD[p1 - p3 - p4],
    FAD[{p1 - p4, m1}], FAD[{p3, m1}], FAD[p3 + q1],
    FAD[p1 + q1]}}
```

$$\left(\frac{1}{p4^2} \quad \frac{1}{p1^2} \quad \frac{1}{(p1-p3-p4)^2} \quad \frac{1}{(p1-p4)^2-m1^2} \quad \frac{1}{p3^2-m1^2} \quad \frac{1}{(p3+q1)^2} \quad \frac{1}{(p1+q1)^2} \right)$$

```
target = {FAD[p4], FAD[p1 + p4 + q1], FAD[p1 - p3 + q1],
   FAD[{p1 + q1, m1}], FAD[{p3, m1}], FAD[p3 + q1],
   FAD[p1 + p4 + 2 q1]}
```

$$\left\{ \frac{1}{p4^2}, \frac{1}{(p1+p4+q1)^2}, \frac{1}{(p1-p3+q1)^2}, \frac{1}{(p1+q1)^2-m1^2}, \frac{1}{p3^2-m1^2}, \frac{1}{(p3+q1)^2}, \frac{1}{(p1+p4+2\,q1)^2} \right\}$$

```
FCLoopFindMomentumShifts[source, target, {p1, p3, p4}]
```

$$\{\{p1 \to p1+p4+q1, p3 \to p3, p4 \to p4\}\}$$

```
FCLoopFindMomentumShifts[{{FAD[r4], FAD[r1], FAD[r1 - p3 - r4],
    FAD[{r1 - r4, m1}], FAD[{p3, m1}], FAD[p3 + q1], FAD[r1 + q1]}},
  {FAD[p4], FAD[p1 + p4 + q1], FAD[p1 - p3 + q1], FAD[{p1 + q1, m1}],
   FAD[{p3, m1}], FAD[p3 + q1], FAD[p1 + p4 + 2 q1]}, {p1, p3, p4, r4,r1}]
```

$$\{\{p3 \to p3, r4 \to p4, r1 \to p1 + p4 + q1\}\}$$

```
source1 = {FCTopology[
    fctopology3, {SFAD[{{p1, 0}, {0, 1}, 1}],
        SFAD[{{p3, 0}, {0, 1}, 1}],
        SFAD[{{p1 + p2 + p3 - Q, 0}, {0, 1}, 1}],
        SFAD[{{p2 + p3, 0}, {0, 1}, 1}],
        SFAD[{{p2 + p3 - Q, 0}, {0, 1}, 1}],
        SFAD[{{p1 + p3 - Q, 0}, {0, 1}, 1}],
        SFAD[{{p1 + p3, 0}, {0, 1}, 1}], SFAD[{{p2 - Q, 0}, {0, 1}, 1}],
        SFAD[{{p2, 0}, {0, 1}, 1}]}, {p1, p2, p3}, {Q}, {}, {}],
   FCTopology[
    fctopology4, {SFAD[{{p2 + p3, 0}, {0, 1}, 1}],
        SFAD[{{p3, 0}, {0, 1}, 1}],
        SFAD[{{p1 + p2 + p3 - Q, 0}, {0, 1}, 1}],
        SFAD[{{p1, 0}, {0, 1}, 1}], SFAD[{{p1 - Q, 0}, {0, 1}, 1}],
        SFAD[{{p2 - Q, 0}, {0, 1}, 1}], SFAD[{{p2, 0}, {0, 1}, 1}],
        SFAD[{{p1 + p3 - Q, 0}, {0, 1}, 1}],
        SFAD[{{p1 + p3, 0}, {0, 1}, 1}]}, {p1, p2, p3}, {Q}, {}, {}]}
```

$$\left\{\text{FCTopology}\left(\text{fctopology3}, \left\{\frac{1}{(p1^2 + i\eta)}, \frac{1}{(p3^2 + i\eta)}, \frac{1}{((p1 + p2 + p3 - Q)^2 + i\eta)}, \frac{1}{((p2 + p3)^2 + i\eta)}, \frac{1}{((p2 + p3 - Q)^2 + i\eta)}, \frac{1}{((p1 + p3 - Q)^2 + i\eta)}, \frac{1}{((p1 + p3)^2 + i\eta)}, \frac{1}{((p2 - Q)^2 + i\eta)}, \frac{1}{(p2^2 + i\eta)}\right\}, \{p1, p2, p3\}, \{Q\}, \{\}, \{\}\right), \text{FCTopology}\left(\text{fctopology4}, \left\{\frac{1}{((p2 + p3)^2 + i\eta)}, \frac{1}{(p3^2 + i\eta)}, \frac{1}{((p1 + p2 + p3 - Q)^2 + i\eta)}, \frac{1}{(p1^2 + i\eta)}, \frac{1}{((p1 - Q)^2 + i\eta)}, \frac{1}{((p2 - Q)^2 + i\eta)}, \frac{1}{(p2^2 + i\eta)}, \frac{1}{((p1 + p3 - Q)^2 + i\eta)}, \frac{1}{((p1 + p3)^2 + i\eta)}\right\}, \{p1, p2, p3\}, \{Q\}, \{\}, \{\}\right)\right\}$$

```
target1 = FCTopology[
   fctopology1, {SFAD[{{p1 + p3 - Q, 0}, {0, 1}, 1}],
    SFAD[{{p3, 0}, {0, 1}, 1}],
    SFAD[{{p1 + p2 + p3 - Q, 0}, {0, 1}, 1}],
    SFAD[{{p2 - Q, 0}, {0, 1}, 1}], SFAD[{{p2, 0}, {0, 1}, 1}],
    SFAD[{{p1, 0}, {0, 1}, 1}], SFAD[{{p1 - Q, 0}, {0, 1}, 1}],
    SFAD[{{p2 + p3, 0}, {0, 1}, 1}],
    SFAD[{{p2 + p3 - Q, 0}, {0, 1}, 1}]}, {p1, p2, p3}, {Q}, {}, {}]
```

$$\text{FCTopology}\left(\text{fctopology1}, \left\{\frac{1}{((\text{p1}+\text{p3}-Q)^2+i\eta)}, \frac{1}{(\text{p3}^2+i\eta)}, \frac{1}{((\text{p1}+\text{p2}+\text{p3}-Q)^2+i\eta)}, \frac{1}{((\text{p2}-Q)^2+i\eta)}, \frac{1}{(\text{p2}^2+i\eta)}, \frac{1}{(\text{p1}^2+i\eta)}, \frac{1}{((\text{p1}-Q)^2+i\eta)}, \frac{1}{((\text{p2}+\text{p3})^2+i\eta)}, \frac{1}{((\text{p2}+\text{p3}-Q)^2+i\eta)}\right\}, \{\text{p1}, \text{p2}, \text{p3}\}, \{Q\}, \{\}, \{\}\right)$$

```
FCLoopFindMomentumShifts[source1, target1]
```

$$\{\{\text{p1} \to -\text{p1}-\text{p3}+Q, \text{p2} \to -\text{p2}-\text{p3}+Q, \text{p3} \to \text{p3}\}, \{\text{p1} \to Q-\text{p2}, \text{p2} \to Q-\text{p1}, \text{p3} \to -\text{p3}\}\}$$

```
source2 = {FCTopology[topo1, {
     SFAD[{{l1 + q1, 0}, {m^2, 1}, 1}],
     SFAD[{{l1 - l2, 0}, {0, 1}, 1}],
     SFAD[{{l2 + q1, 0}, {m^2, 1}, 1}],
     SFAD[{{l2 - q2, 0}, {m^2, 1}, 1}],
     SFAD[{{l2, 0}, {0, 1}, 1}]}, {l1, l2}, {q1, q2},
    {SPD[q1, q1] -> 0, SPD[q2, q2] -> 0, SPD[q1, q2] -> s/2}, {}]}
```

$$\left\{\text{FCTopology}\left(\text{topo1}, \left\{\frac{1}{((\text{l1}+\text{q1})^2-m^2+i\eta)}, \frac{1}{((\text{l1}-\text{l2})^2+i\eta)}, \frac{1}{((\text{l2}+\text{q1})^2-m^2+i\eta)}, \frac{1}{((\text{l2}-\text{q2})^2-m^2+i\eta)}, \frac{1}{(\text{l2}^2+i\eta)}\right\}, \{\text{l1}, \text{l2}\}, \{\text{q1}, \text{q2}\}, \left\{\text{q1}^2 \to 0, \text{q2}^2 \to 0, \text{q1}\cdot\text{q2} \to \frac{s}{2}\right\}, \{\}\right)\right\}$$

```
target2 = FCTopology[topo2, {
    SFAD[{{l1 - l2, 0}, {m^2, 1}, 1}],
    SFAD[{{l1 - q2, 0}, {0, 1}, 1}],
    SFAD[{{l2 - q2, 0}, {m^2, 1}, 1}],
    SFAD[{{l2 + q1, 0}, {m^2, 1}, 1}],
    SFAD[{{l2, 0}, {0, 1}, 1}]}, {l1, l2}, {q1, q2},
   {SPD[q1, q1] -> 0, SPD[q2, q2] -> 0, SPD[q1, q2] -> s/2}, {}]
```

$$\text{FCTopology}\left(\text{topo2}, \left\{\frac{1}{((\text{l1}-\text{l2})^2-m^2+i\eta)}, \frac{1}{((\text{l1}-\text{q2})^2+i\eta)}, \frac{1}{((\text{l2}-\text{q2})^2-m^2+i\eta)}, \frac{1}{((\text{l2}+\text{q1})^2-m^2+i\eta)}, \frac{1}{(\text{l2}^2+i\eta)}\right\}, \{\text{l1}, \text{l2}\}, \{\text{q1}, \text{q2}\}, \left\{\text{q1}^2 \to 0, \text{q2}^2 \to 0, \text{q1}\cdot\text{q2} \to \frac{s}{2}\right\}, \{\}\right)$$

Mapping these two topologies onto each other requires shifts in the external momenta

```
Quiet[FCLoopFindMomentumShifts[source2, target2, Abort -> False]]
```

$$\{\{\}\}$$

Once we allow such shifts, everything works as expected

```
FCLoopFindMomentumShifts[source2, target2, Momentum -> All]
```

$$\{\{l1 \to l1 - l2 - q2, l2 \to -l2, q1 \to q2, q2 \to q1\}\}$$

9.36 FCLoopFindIntegralMappings

`FCLoopFindIntegralMappings[{int1, int2, ...}, {p1, p2, ...}]` finds mappings between scalar multiloop integrals **`int1, int2, ...`** that depend on the loop momenta **`p1, p2, ...`** using the algorithm of Alexey Pak arXiv:1111.0868.

The current implementation is based on the **`FindEquivalents`** function from FIRE 6 arXiv:1901.07808

It is also possible to invoke the function as **`FCLoopFindIntegralMappings[{GLI[...], ...}, {FCTopology[...], ...}]` or `FCLoopFindIntegralMappings[{FCTopology[...], ...}]`**.

Notice that in this case the value of the option **`FinalSubstitutions`** is ignored, as replacement rules will be extracted directly from the definition of the topology.

The default output is a list of two lists. The first list contains mapping rules between different loop integrals, while the second list provides all unique master integrals extracted from the input.

An alternative output mode is activated when the option **`List`** is set to **`True`**. In this case the output is a list of lists, where each list contains master integrals that were identified to be identical.

The option **`PreferredIntegrals`** can be used to enforce the mapping onto a preferred set of master integral. Notice that the final result will only contain those preferred integrals, that are actually present in the input.

9.36.1 See also

Overview, FCTopology, GLI, FCLoopToPakForm, FCLoopPakOrder, FCLoopFindTopologyMappings

9.36.2 Examples

When given a list of **`FeynAmpDenominator`**-integrals, the function will merely group identical integrals into sublists

```
ints = {FAD[{p1, m1}], FAD[{p1 + q, m1}], FAD[{p1, m2}]}
```

$$\left\{\frac{1}{p1^2 - m1^2}, \frac{1}{(p1+q)^2 - m1^2}, \frac{1}{p1^2 - m2^2}\right\}$$

```
FCLoopFindIntegralMappings[ints, {p1}]
```

$$\left\{\left\{\frac{1}{\text{p1}^2-\text{m1}^2},\frac{1}{(\text{p1}+q)^2-\text{m1}^2}\right\},\left\{\frac{1}{\text{p1}^2-\text{m2}^2}\right\}\right\}$$

The following 3 integrals look rather different from each other, but are actually identical

```
ints = {FAD[p1] FAD[p1 - p3 - p4] FAD[p4] FAD[p3 + q1] FAD[{p3, m1}]
FAD[{p1 - p4, m1}]*
    FAD[{p1 + q1, 0}, {p1 + q1, 0}], FAD[p4] FAD[p1 - p3 + q1] FAD[p3 + q1]
FAD[p1 + p4 + q1]*
    FAD[{p3, m1}] FAD[{p1 + q1, m1}] FAD[{p1 + p4 + 2 q1, 0}, {p1 + p4 + 2
q1, 0}],
   FAD[p1] FAD[p4 - 2 q1] FAD[p3 + q1] FAD[p1 - p3 - p4 + 2 q1] FAD[{p3,
m1}]*
    FAD[{p1 - p4 + 2 q1, m1}] FAD[{p1 + q1, 0}, {p1 + q1, 0}]}
```

$$\left\{\frac{1}{\text{p1}^2\,\text{p4}^2\left(\text{p3}^2-\text{m1}^2\right)(\text{p1}+\text{q1})^4(\text{p3}+\text{q1})^2(\text{p1}-\text{p3}-\text{p4})^2\left((\text{p1}-\text{p4})^2-\text{m1}^2\right)},\right.$$
$$\frac{1}{\text{p4}^2\left(\text{p3}^2-\text{m1}^2\right)(\text{p3}+\text{q1})^2(\text{p1}-\text{p3}+\text{q1})^2(\text{p1}+\text{p4}+\text{q1})^2(\text{p1}+\text{p4}+2\,\text{q1})^4\left((\text{p1}+\text{q1})^2-\text{m1}^2\right)},$$
$$\left.\frac{1}{\text{p1}^2\left(\text{p3}^2-\text{m1}^2\right)(\text{p1}+\text{q1})^4(\text{p3}+\text{q1})^2(\text{p4}-2\,\text{q1})^2(\text{p1}-\text{p3}-\text{p4}+2\,\text{q1})^2\left((\text{p1}-\text{p4}+2\,\text{q1})^2-\text{m1}^2\right)}\right\}$$

```
FCLoopFindIntegralMappings[ints, {p1, p3, p4}]
```

$$\left(\frac{1}{\text{p1}^2\,\text{p4}^2\left(\text{p3}^2-\text{m1}^2\right)(\text{p1}+\text{q1})^4(\text{p3}+\text{q1})^2(\text{p1}-\text{p3}-\text{p4})^2\left((\text{p1}-\text{p4})^2-\text{m1}^2\right)}\quad\frac{1}{\text{p4}^2\left(\text{p3}^2-\text{m1}^2\right)(\text{p3}+\text{q1})^2(\text{p1}-\text{p3}+\text{q1})^2(\text{p1}+\text{p4}+\text{q1}}\right.$$

If the input is a list of **GLI**-integrals, **FCLoopFindIntegralMappings** will return a list containing two sublists. The former will be a list of replacement rules while the latter will contain all unique master integrals

```
ClearAll[topo1, topo2];

topos = {
   FCTopology[topo1, {SFAD[{p1, m^2}], SFAD[{p2, m^2}]}, {p1, p2}, {}, {},
{}],
   FCTopology[topo2, {SFAD[{p3, m^2}], SFAD[{p4, m^2}]}, {p3, p4}, {}, {},
{}]
  }
```

$$\left\{\text{FCTopology}\left(\text{topo1}, \left\{\frac{1}{(\text{p1}^2 - m^2 + i\eta)}, \frac{1}{(\text{p2}^2 - m^2 + i\eta)}\right\}, \{\text{p1}, \text{p2}\}, \{\}, \{\}, \{\}\right),\right.$$
$$\left.\text{FCTopology}\left(\text{topo2}, \left\{\frac{1}{(\text{p3}^2 - m^2 + i\eta)}, \frac{1}{(\text{p4}^2 - m^2 + i\eta)}\right\}, \{\text{p3}, \text{p4}\}, \{\}, \{\}, \{\}\right)\right\}$$

```
glis = {GLI[topo1, {1, 1}], GLI[topo1, {1, 2}], GLI[topo1, {2, 1}],
GLI[topo2, {1, 1}],
   GLI[topo2, {2, 2}]}
```

$$\left\{G^{\text{topo1}}(1,1), G^{\text{topo1}}(1,2), G^{\text{topo1}}(2,1), G^{\text{topo2}}(1,1), G^{\text{topo2}}(2,2)\right\}$$

```
FCLoopFindIntegralMappings[glis, topos]
```

$$\left\{\left\{G^{\text{topo2}}(1,1) \to G^{\text{topo1}}(1,1), G^{\text{topo1}}(2,1) \to G^{\text{topo1}}(1,2)\right\}, \left\{G^{\text{topo1}}(1,1), G^{\text{topo1}}(1,2), G^{\text{topo2}}(2,2)\right\}\right\}$$

This behavior can be turned off by setting the value of the option **`List`** to **`True`**

```
FCLoopFindIntegralMappings[glis, topos, List -> True]
```

$$\left\{\left\{G^{\text{topo1}}(1,1), G^{\text{topo2}}(1,1)\right\}, \left\{G^{\text{topo1}}(1,2), G^{\text{topo1}}(2,1)\right\}, \left\{G^{\text{topo2}}(2,2)\right\}\right\}$$

In practice, one usually has a list of preferred integrals onto which one would like to map the occurring master integrals. Such integrals can be specified via the **`PreferredIntegrals`** option

```
FCLoopFindIntegralMappings[glis, topos, PreferredIntegrals -> {GLI[topo2,
{1, 1}],
    GLI[topo2, {2, 1}]}]
```

$$\left(\begin{array}{ccc} G^{\text{topo1}}(1,1) \to G^{\text{topo2}}(1,1) & G^{\text{topo1}}(1,2) \to G^{\text{topo2}}(2,1) & G^{\text{topo1}}(2,1) \to G^{\text{topo2}}(2,1) \\ G^{\text{topo2}}(1,1) & G^{\text{topo2}}(2,1) & G^{\text{topo2}}(2,2) \end{array}\right)$$

The indices of **`GLI`**s do not have to be integers

```
topos = {
   FCTopology[prop2LtopoG20, {SFAD[{{p1, 0}, {0, 1}, 1}],
     SFAD[{{p1 + q1, 0}, {m3^2, 1}, 1}], SFAD[{{p3, 0}, {0, 1}, 1}],
     SFAD[{{p3 + q1, 0}, {0, 1}, 1}], SFAD[{{p1 - p3, 0}, {0, 1}, 1}]},
    {p1, p3}, {q1}, {}, {}],
   FCTopology[prop2LtopoG21, {SFAD[{{p1, 0}, {m1^2, 1}, 1}],
     SFAD[{{p1 + q1, 0}, {m3^2, 1}, 1}],
     SFAD[{{p3, 0}, {0, 1}, 1}], SFAD[{{p3 + q1, 0}, {0, 1}, 1}],
     SFAD[{{p1 - p3, 0}, {0, 1}, 1}]}, {p1, p3}, {q1}, {}, {}]
  }
```

$$\left\{\text{FCTopology}\left(\text{prop2LtopoG20}, \left\{\frac{1}{(\text{p1}^2+i\eta)}, \frac{1}{((\text{p1}+\text{q1})^2-\text{m3}^2+i\eta)}, \frac{1}{(\text{p3}^2+i\eta)}, \frac{1}{((\text{p3}+\text{q1})^2+i\eta)}, \frac{1}{((\text{p1}-\text{p3})^2+i\eta)}\right\}, \{\text{p1},\text{p3}\}, \{\text{q1}\}, \{\}, \{\}\right), \text{FCTopology}\left(\text{prop2LtopoG21}, \left\{\frac{1}{(\text{p1}^2-\text{m1}^2+i\eta)}, \frac{1}{((\text{p1}+\text{q1})^2-\text{m3}^2+i\eta)}, \frac{1}{(\text{p3}^2+i\eta)}, \frac{1}{((\text{p3}+\text{q1})^2+i\eta)}, \frac{1}{((\text{p1}-\text{p3})^2+i\eta)}\right\}, \{\text{p1},\text{p3}\}, \{\text{q1}\}, \{\}, \{\}\right)\right\}$$

```
FCLoopFindIntegralMappings[{GLI[prop2LtopoG21, {0, n1, n2, n3, n4}],
   GLI[prop2LtopoG20, {0, n1, n2, n3, n4}]}, topos]
```

$$\left(\begin{array}{c} G^{\text{prop2LtopoG21}}(0,\text{n1},\text{n2},\text{n3},\text{n4}) \to G^{\text{prop2LtopoG20}}(0,\text{n1},\text{n2},\text{n3},\text{n4}) \\ G^{\text{prop2LtopoG20}}(0,\text{n1},\text{n2},\text{n3},\text{n4}) \end{array}\right)$$

It is also possible to find mappings for factorizing integrals, provided that suitable products of integrals are given as preferred integrals

```
topos = {FCTopology[prop2Ltopo31313, {SFAD[{{
        I p1, 0}, {-m3^2, -1}, 1}], SFAD[{{I (p1 + q1), 0}, {-m1^2, -1},
        1}], SFAD[{{
        I p3, 0}, {-m3^2, -1}, 1}], SFAD[{{I
         (p3 + q1), 0}, {-m1^2, -1}, 1}], SFAD[{{
        I (p1 - p3), 0}, {-m3^2, -1}, 1}]}, {p1, p3}, {q1}, {SPD[q1, q1] ->
        m1^2}, {}],
   FCTopology[tad1Ltopo2, {SFAD[{{I p1, 0}, {-m3^2, -1}, 1}]}, {p1}, {},
{SPD[q1,q1] -> m1^2}, {}]}
```

$$\left\{\text{FCTopology}\left(\text{prop2Ltopo31313},\left\{\frac{1}{(-\text{p1}^2+\text{m3}^2-i\eta)},\frac{1}{(-(\text{p1}+\text{q1})^2+\text{m1}^2-i\eta)},\right.\right.\right.$$
$$\left.\frac{1}{(-\text{p3}^2+\text{m3}^2-i\eta)},\frac{1}{(-(\text{p3}+\text{q1})^2+\text{m1}^2-i\eta)},\frac{1}{(-(\text{p1}-\text{p3})^2+\text{m3}^2-i\eta)}\right\},$$
$$\{\text{p1},\text{p3}\},\{\text{q1}\},\{\text{q1}^2\to\ \text{m1}^2\},\{\}\Big),\text{FCTopology}\Big(\text{tad1Ltopo2},$$
$$\left.\left\{\frac{1}{(-\text{p1}^2+\text{m3}^2-i\eta)}\right\},\{\text{p1}\},\{\},\{\text{q1}^2\to\ \text{m1}^2\},\{\}\Big)\right\}$$

Here we ask the function to map all products of two 1-loop tadpoles to **`GLI[tad1Ltopo2, {1}]^2`**

```
FCLoopFindIntegralMappings[{GLI[tad1Ltopo2, {1}]^2,
   GLI[prop2Ltopo31313, {0, 0, 1, 0, 1}]}, topos, PreferredIntegrals ->
{GLI[tad1Ltopo2, {1}]^2}]
```

$$\left(\begin{array}{c} G^{\text{prop2Ltopo31313}}(0,0,1,0,1)\to G^{\text{tad1Ltopo2}}(1)^2 \\ G^{\text{tad1Ltopo2}}(1)^2 \end{array}\right)$$

`FCLoopFindSubtopologies[topo]` finds all scalefull subtopologies of the FCTopology **`topo`**.

Each subtopology receives a marker that specifies the topology from which it was derived. The symbol denoting the marker is specified via the option **`SubtopologyMarker`**. Setting it to **`False`** will disable the inclusion of the markers

9.36.3 See also

Overview, FCTopology, FCLoopFindTopologies, FCLoopFindTopologyMappings, SubtopologyMarker.

9.36.4 Examples

```
res = FCLoopFindSubtopologies[FCTopology[TRI, {SFAD[{{p1, 0}, {0, 1}, 1}],
      SFAD[{{p2, 0}, {0, 1}, 1}], SFAD[{{p1 + Q1, 0}, {0, 1}, 1}],
SFAD[{{p1 + p2 + Q1, 0},
        {0, 1}, 1}], SFAD[{{-p1 + Q2, 0}, {0, 1}, 1}], SFAD[{{-p1 - p2 +
        Q2, 0}, {0, 1}, 1}]},
     {p1, p2}, {Q1, Q2}, {}, {}]];
```

```
res // Length
```

19

Show the first three subtopologies of this 2-loop self-energy topology

```
res[[1 ;; 3]]
```

$$\left\{\text{FCTopology}\left(\text{TRI},\left\{\frac{1}{(\text{p1}^2+i\eta)},\frac{1}{(\text{p2}^2+i\eta)},\frac{1}{((\text{p1}+\text{Q1})^2+i\eta)},\frac{1}{((\text{p1}+\text{p2}+\text{Q1})^2+i\eta)},\frac{1}{((\text{Q2}-\text{p1})^2+i\eta)},\frac{1}{((-\text{p1}-\text{p2}+\text{Q2})^2+i\eta)}\right\},\{\text{p1},\text{p2}\},\{\text{Q1},\text{Q2}\},\{\},\{\}\right),\text{FCTopology}\left(\text{TRIR1},\left\{\frac{1}{(\text{p2}^2+i\eta)},\frac{1}{((\text{p1}+\text{Q1})^2+i\eta)},\frac{1}{((\text{p1}+\text{p2}+\text{Q1})^2+i\eta)},\frac{1}{((\text{Q2}-\text{p1})^2+i\eta)},\frac{1}{((-\text{p1}-\text{p2}+\text{Q2})^2+i\eta)}\right\},\{\text{p1},\text{p2}\},\{\text{Q1},\text{Q2}\},\{\},\{\text{FCGV(SubtopologyOf)}\to\text{TRI}\}\right),\text{FCTopology}\left(\text{TRIR2},\left\{\frac{1}{(\text{p1}^2+i\eta)},\frac{1}{((\text{p1}+\text{Q1})^2+i\eta)},\frac{1}{((\text{p1}+\text{p2}+\text{Q1})^2+i\eta)},\frac{1}{((\text{Q2}-\text{p1})^2+i\eta)},\frac{1}{((-\text{p1}-\text{p2}+\text{Q2})^2+i\eta)}\right\},\{\text{p1},\text{p2}\},\{\text{Q1},\text{Q2}\},\{\},\{\text{FCGV(SubtopologyOf)}\to\text{TRI}\}\right)\right\}$$

```
res = FCLoopFindSubtopologies[FCTopology[topo1, {SFAD[{{p3, 0}, {0, 1},
1}],
      SFAD[{{p2, 0}, {0, 1}, 1}], SFAD[{{p1, 0}, {0, 1}, 1}],
      SFAD[{{p2 + p3, 0}, {0, 1}, 1}], SFAD[{{p2 - Q, 0}, {0, 1}, 1}],
      SFAD[{{p1 - Q, 0}, {0, 1}, 1}], SFAD[{{p2 + p3 - Q, 0}, {0, 1}, 1}],
      SFAD[{{p1 + p3 - Q, 0}, {0, 1}, 1}], SFAD[{{p1 + p2 + p3 - Q, 0},
        {0, 1}, 1}]}, {p1, p2, p3}, {Q}, {}, {}], FCE -> True];
```

```
res // Length
```

36

Show the first three subtopologies of this 3-loop self-energy topology

```
res[[1 ;; 3]]
```

$$\left\{\text{FCTopology}\left(\text{topo1},\left\{\frac{1}{(\text{p3}^2+i\eta)},\frac{1}{(\text{p2}^2+i\eta)},\frac{1}{(\text{p1}^2+i\eta)},\frac{1}{((\text{p2}+\text{p3})^2+i\eta)},\right.\right.\right.$$
$$\frac{1}{((\text{p2}-Q)^2+i\eta)},\frac{1}{((\text{p1}-Q)^2+i\eta)},\frac{1}{((\text{p2}+\text{p3}-Q)^2+i\eta)},\frac{1}{((\text{p1}+\text{p3}-Q)^2+i\eta)},$$
$$\left.\left.\frac{1}{((\text{p1}+\text{p2}+\text{p3}-Q)^2+i\eta)}\right\},\{\text{p1},\text{p2},\text{p3}\},\{Q\},\{\},\{\}\right),\text{FCTopology}\left(\text{topo1R1},\right.$$
$$\left\{\frac{1}{(\text{p2}^2+i\eta)},\frac{1}{(\text{p1}^2+i\eta)},\frac{1}{((\text{p2}+\text{p3})^2+i\eta)},\frac{1}{((\text{p2}-Q)^2+i\eta)},\frac{1}{((\text{p1}-Q)^2+i\eta)},\right.$$
$$\left.\frac{1}{((\text{p2}+\text{p3}-Q)^2+i\eta)},\frac{1}{((\text{p1}+\text{p3}-Q)^2+i\eta)},\frac{1}{((\text{p1}+\text{p2}+\text{p3}-Q)^2+i\eta)}\right\},\{\text{p1},$$
$$\text{p2},\text{p3}\},\{Q\},\{\},\{\text{FCGV(SubtopologyOf)}\to\text{topo1}\}\Big),\text{FCTopology}\Big(\text{topo1R2},$$
$$\left\{\frac{1}{(\text{p3}^2+i\eta)},\frac{1}{(\text{p1}^2+i\eta)},\frac{1}{((\text{p2}+\text{p3})^2+i\eta)},\frac{1}{((\text{p2}-Q)^2+i\eta)},\frac{1}{((\text{p1}-Q)^2+i\eta)},\right.$$
$$\left.\frac{1}{((\text{p2}+\text{p3}-Q)^2+i\eta)},\frac{1}{((\text{p1}+\text{p3}-Q)^2+i\eta)},\frac{1}{((\text{p1}+\text{p2}+\text{p3}-Q)^2+i\eta)}\right\},$$
$$\{\text{p1},\text{p2},\text{p3}\},\{Q\},\{\},\{\text{FCGV(SubtopologyOf)}\to\text{topo1}\}\Big)\Big\}$$

9.37 FCLoopFindTopologies

`FCLoopFindTopologies[exp, {q1, q2, ...}]` attempts to identify the loop integral topologies present in **`exp`** by looking at the propagator denominators that depend on the loop momenta **`q1, q2, ...`** . It returns a list of two entries, where the first one is the original expression with the denominators rewritten as **`GLI`**s, and the second one is the set of the identified topologies.

Each of the identified topologies must contain linearly independent propagators (unless the option **`FCLoopBasisOverdeterminedQ`** is set to True), but may lack propagators needed to form a complete basis.

Scaleless topologies are automatically removed, but this can be disabled by setting the option **`FCLoopScalelessQ`** to **`True`**.

9.37.1 See also

Overview, FCTopology, GLI.

9.37.2 Examples

Find topologies occurring in the 2-loop ghost self-energy amplitude

```
amp = Get[FileNameJoin[{$FeynCalcDirectory, "Documentation", "Examples",
      "Amplitudes", "Gh-Gh-2L.m"}]];
```

```
res = FCLoopFindTopologies[amp, {q1, q2}];
```

Number of the initial candidate topologies: 3

Number of the identified unique topologies: 3

Number of the preferred topologies among the unique topologies: 0

Number of the identified subtopologies: 0

```
res // Last
```

$$\left\{\text{FCTopology}\left(\text{fctopology1},\left\{\frac{1}{(\text{q2}^2+i\eta)},\frac{1}{(\text{q1}^2+i\eta)},\frac{1}{((\text{q1}+\text{q2})^2+i\eta)},\frac{1}{((p+\text{q1})^2+i\eta)},\frac{1}{((p-\text{q2})^2+i\eta)}\right\},\{\text{q1},\text{q2}\},\{p\},\{\},\{\}\right),\text{FCTopology}\left(\text{fctopology2},\left\{\frac{1}{(\text{q2}^2+i\eta)},\frac{1}{(\text{q1}^2+i\eta)},\frac{1}{((p+\text{q2})^2+i\eta)},\frac{1}{((p-\text{q1})^2+i\eta)}\right\},\{\text{q1},\text{q2}\},\{p\},\{\},\{\}\right),\text{FCTopology}\left(\text{fctopology3},\left\{\frac{1}{(\text{q2}^2+i\eta)},\frac{1}{(\text{q1}^2+i\eta)},\frac{1}{((p-\text{q1})^2+i\eta)},\frac{1}{((p-\text{q1}+\text{q2})^2+i\eta)}\right\},\{\text{q1},\text{q2}\},\{p\},\{\},\{\}\right)\right\}$$

Find topologies occurring in the 2-loop QCD corrections to the B_s-meson mixing

```
topos = Get[FileNameJoin[{$FeynCalcDirectory, "Documentation", "Examples",
      "Topologies", "BMixingTopos2L.m"}]];
```

```
topos // Length
```

544

```
res = FCLoopFindTopologies[topos, {k1, k2}];
```

Number of the initial candidate topologies: 18

Number of the identified unique topologies: 18

Number of the preferred topologies among the unique topologies: 0

Number of the identified subtopologies: 0

Show the first two topologies

```
(res // Last)[[1 ;; 2]]
```

$$\left\{\text{FCTopology}\left(\text{fctopology1}, \left\{\frac{1}{((\text{k2}+\text{p1})^2+i\eta)}, \frac{1}{(\text{k2}^2-\text{mb}^2+i\eta)}, \frac{1}{(\text{k1}^2-\text{mc}^2+i\eta)}, \frac{1}{((\text{k1}+\text{k2})^2-\text{mc}^2+i\eta)}, \frac{1}{((\text{k1}-\text{p1})^2-\text{mc}^2+i\eta)}\right\}, \{\text{k1}, \text{k2}\}, \{\text{p1}\}, \{\}, \{\}\right), \text{FCTopology}\left(\text{fctopology2}, \left\{\frac{1}{((\text{k2}-\text{p1})^2+i\eta)}, \frac{1}{(\text{k2}^2-\text{mb}^2+i\eta)}, \frac{1}{(\text{k1}^2-\text{mc}^2+i\eta)}, \frac{1}{((\text{k1}+\text{p1})^2-\text{mc}^2+i\eta)}, \frac{1}{((\text{k1}+\text{k2})^2-\text{mc}^2+i\eta)}\right\}, \{\text{k1}, \text{k2}\}, \{\text{p1}\}, \{\}, \{\}\right)\right\}$$

In practical calculations even the simple extraction of topologies from the given list of diagrams can take considerable amount of time. This is why it is better to parallelize this process as much as possible.

We can split this part into two steps, were we first apply **FCLoopIsolate** to the resulting amplitudes and then pass the output to **FCLoopFindTopologies**. To avoid the unnecessary application of **FCLoopIsolate** during this step, we should specify the head with which the topologies have been wrapped via the option **FCLoopIsolate**.

```
isolatedTopos = FCLoopIsolate[topos[[44 ;; 48]], {k1, k2},  Collecting ->
False, Factoring -> False, Numerator -> False, Head -> loopDen];
```

```
res = FCLoopFindTopologies[isolatedTopos, {k1, k2}, FCLoopIsolate ->
loopDen, Head -> ampDen, Collecting -> False];
```

Number of the initial candidate topologies: 2

Number of the identified unique topologies: 2

Number of the preferred topologies among the unique topologies: 0

Number of the identified subtopologies: 0

9.38 FCLoopFindTopologyMappings

FCLoopFindTopologyMappings[{topo1, topo2, ...}] finds mappings between topologies (written as **FCTopology** objects) **topo1, topo2, ...**. For each source topology the function returns a list of loop momentum shifts and a **GLI** replacement rule needed to map it to the given target topology. If you need to map everything to a particular set of target topologies, you can specify them via the **PreferredTopologies** option.

The output is a list of two lists, the former containing the mappings and the latter enumerating the final contributing topologies

To enable shifts in the external momenta you need to set the option **Momentum** to **All**.

9.38.1 See also

Overview, FCTopology, GLI, FCLoopFindTopologies.

9.38.2 Examples

Here we have a set of 5 topologies

```
topos1 = {
    FCTopology[fctopology1, {SFAD[{{p3, 0}, {0, 1}, 1}], SFAD[{{p2, 0}, {0,
1}, 1}],
      SFAD[{{p1, 0}, {0, 1}, 1}], SFAD[{{p2 + p3, 0}, {0, 1}, 1}],
SFAD[{{p2 - Q, 0}, {0, 1}, 1}],
      SFAD[{{p1 - Q, 0}, {0, 1}, 1}], SFAD[{{p2 + p3 - Q, 0}, {0, 1}, 1}],
SFAD[{{p1 + p3 - Q, 0}, {0, 1}, 1}],
      SFAD[{{p1 + p2 + p3 - Q, 0}, {0, 1}, 1}]}, {p1, p2, p3}, {Q}, {},
{}],
    FCTopology[fctopology2, {SFAD[{{p3, 0}, {0, 1}, 1}],
      SFAD[{{p2, 0}, {0, 1}, 1}], SFAD[{{p1, 0}, {0, 1}, 1}], SFAD[{{p2 +
p3, 0}, {0, 1}, 1}],
      SFAD[{{p2 - Q, 0}, {0, 1}, 1}], SFAD[{{p1 - Q, 0}, {0, 1}, 1}],
      SFAD[{{p2 + p3 - Q, 0}, {0, 1}, 1}], SFAD[{{p1 + p2 - Q, 0}, {0, 1},
1}],
      SFAD[{{p1 + p2 + p3 - Q, 0}, {0, 1}, 1}]}, {p1, p2, p3}, {Q}, {},
{}],
    FCTopology[fctopology3, {SFAD[{{p3, 0}, {0, 1}, 1}],
      SFAD[{{p2, 0}, {0, 1}, 1}], SFAD[{{p1, 0}, {0, 1}, 1}],
      SFAD[{{p2 + p3, 0}, {0, 1}, 1}], SFAD[{{p1 + p3, 0}, {0, 1}, 1}],
      SFAD[{{p2 - Q, 0}, {0, 1}, 1}], SFAD[{{p2 + p3 - Q, 0}, {0, 1}, 1}],
      SFAD[{{p1 + p3 - Q, 0}, {0, 1}, 1}], SFAD[{{p1 + p2 + p3 - Q, 0}, {0,
1}, 1}]},
     {p1, p2, p3}, {Q}, {}, {}],
    FCTopology[fctopology4, {SFAD[{{p3, 0}, {0, 1}, 1}],
      SFAD[{{p2, 0}, {0, 1}, 1}], SFAD[{{p1, 0}, {0, 1}, 1}],
      SFAD[{{p2 + p3, 0}, {0, 1}, 1}], SFAD[{{p1 + p3, 0}, {0, 1}, 1}],
      SFAD[{{p2 - Q, 0}, {0, 1}, 1}], SFAD[{{p1 - Q, 0}, {0, 1}, 1}],
```

```
      SFAD[{{p1 + p3 - Q, 0}, {0, 1}, 1}], SFAD[{{p1 + p2 + p3 - Q, 0}, {0,
1}, 1}]},
     {p1, p2, p3}, {Q}, {}, {}],
    FCTopology[fctopology5, {SFAD[{{p3, 0}, {0, 1}, 1}],
      SFAD[{{p2, 0}, {0, 1}, 1}], SFAD[{{p1, 0}, {0, 1}, 1}],
      SFAD[{{p1 + p3, 0}, {0, 1}, 1}], SFAD[{{p2 - Q, 0}, {0, 1}, 1}],
      SFAD[{{p1 - Q, 0}, {0, 1}, 1}], SFAD[{{p1 + p3 - Q, 0}, {0, 1}, 1}],
      SFAD[{{p1 + p2 - Q, 0}, {0, 1}, 1}], SFAD[{{p1 + p2 + p3 - Q, 0}, {0,
1}, 1}]},
     {p1, p2, p3}, {Q}, {}, {}]};
```

3 of them can be mapped to the other two

```
mappings1 = FCLoopFindTopologyMappings[topos1];
```

```
mappings1[[1]]
```

$$\left(\begin{array}{l}\text{FCTopology}\left(\text{fctopology3},\left\{\frac{1}{(\text{p3}^2+i\eta)},\frac{1}{(\text{p2}^2+i\eta)},\frac{1}{(\text{p1}^2+i\eta)},\frac{1}{((\text{p2}+\text{p3})^2+i\eta)},\frac{1}{((\text{p1}+\text{p3})^2+i\eta)},\frac{1}{((\text{p2}-Q)^2+i\eta)},\frac{1}{((\text{p2}+\text{p3}-Q)}\right.\right.\\ \text{FCTopology}\left(\text{fctopology4},\left\{\frac{1}{(\text{p3}^2+i\eta)},\frac{1}{(\text{p2}^2+i\eta)},\frac{1}{(\text{p1}^2+i\eta)},\frac{1}{((\text{p2}+\text{p3})^2+i\eta)},\frac{1}{((\text{p1}+\text{p3})^2+i\eta)},\frac{1}{((\text{p2}-Q)^2+i\eta)},\frac{1}{((\text{p1}-Q)^2+}\right.\right.\\ \text{FCTopology}\left(\text{fctopology5},\left\{\frac{1}{(\text{p3}^2+i\eta)},\frac{1}{(\text{p2}^2+i\eta)},\frac{1}{(\text{p1}^2+i\eta)},\frac{1}{((\text{p1}+\text{p3})^2+i\eta)},\frac{1}{((\text{p2}-Q)^2+i\eta)},\frac{1}{((\text{p1}-Q)^2+i\eta)},\frac{1}{((\text{p1}+\text{p3}-Q)^2}\right.\right.\end{array}\right.$$

And these are the final topologies

```
mappings1[[2]]
```

$$\left\{\text{FCTopology}\left(\text{fctopology1},\left\{\frac{1}{(\text{p3}^2+i\eta)},\frac{1}{(\text{p2}^2+i\eta)},\frac{1}{(\text{p1}^2+i\eta)},\right.\right.\right.$$
$$\left.\frac{1}{((\text{p2}+\text{p3})^2+i\eta)},\frac{1}{((\text{p2}-Q)^2+i\eta)},\frac{1}{((\text{p1}-Q)^2+i\eta)},\frac{1}{((\text{p2}+\text{p3}-Q)^2+i\eta)},\right.$$
$$\left.\left.\frac{1}{((\text{p1}+\text{p3}-Q)^2+i\eta)},\frac{1}{((\text{p1}+\text{p2}+\text{p3}-Q)^2+i\eta)}\right\},\{\text{p1},\text{p2},\text{p3}\},\{Q\},\right.$$
$$\left.\{\},\{\}\right),\text{FCTopology}\left(\text{fctopology2},\left\{\frac{1}{(\text{p3}^2+i\eta)},\frac{1}{(\text{p2}^2+i\eta)},\frac{1}{(\text{p1}^2+i\eta)},\right.\right.$$
$$\left.\frac{1}{((\text{p2}+\text{p3})^2+i\eta)},\frac{1}{((\text{p2}-Q)^2+i\eta)},\frac{1}{((\text{p1}-Q)^2+i\eta)},\frac{1}{((\text{p2}+\text{p3}-Q)^2+i\eta)},\right.$$
$$\left.\left.\left.\frac{1}{((\text{p1}+\text{p2}-Q)^2+i\eta)},\frac{1}{((\text{p1}+\text{p2}+\text{p3}-Q)^2+i\eta)}\right\},\{\text{p1},\text{p2},\text{p3}\},\{Q\},\{\},\{\}\right)\right\}$$

Here is another example

```
topos2 = {FCTopology[fctopology1, {SFAD[{{q2, 0}, {0, 1}, 1}],
     SFAD[{{q1, 0}, {0, 1}, 1}], SFAD[{{q1 + q2, 0}, {0, 1}, 1}], SFAD[{{p
+ q1, 0}, {0, 1}, 1}],
     SFAD[{{p - q2, 0}, {0, 1}, 1}]}, {q1, q2}, {p}, {}, {}],
   FCTopology[fctopology2, {SFAD[{{q2, 0}, {0, 1}, 1}], SFAD[{{q1, 0}, {0,
1}, 1}],
     SFAD[{{p + q2, 0}, {0, 1}, 1}], SFAD[{{p - q1, 0}, {0, 1}, 1}]}, {q1,
q2}, {p}, {}, {}],
   FCTopology[fctopology3, {SFAD[{{q2, 0}, {0, 1}, 1}], SFAD[{{q1, 0}, {0,
1}, 1}],
     SFAD[{{p - q1, 0}, {0, 1}, 1}], SFAD[{{p - q1 + q2, 0}, {0, 1}, 1}]},
{q1, q2}, {p}, {}, {}]}
```

$$\left\{\text{FCTopology}\left(\text{fctopology1}, \left\{\frac{1}{(\text{q2}^2+i\eta)}, \frac{1}{(\text{q1}^2+i\eta)}, \frac{1}{((\text{q1}+\text{q2})^2+i\eta)}, \frac{1}{((p+\text{q1})^2+i\eta)}, \frac{1}{((p-\text{q2})^2+i\eta)}\right\}, \{\text{q1}, \text{q2}\}, \{p\}, \{\}, \{\}\right), \text{FCTopology}\left(\text{fctopology2}, \left\{\frac{1}{(\text{q2}^2+i\eta)}, \frac{1}{(\text{q1}^2+i\eta)}, \frac{1}{((p+\text{q2})^2+i\eta)}, \frac{1}{((p-\text{q1})^2+i\eta)}\right\}, \{\text{q1}, \text{q2}\}, \{p\}, \{\}, \{\}\right), \text{FCTopology}\left(\text{fctopology3}, \left\{\frac{1}{(\text{q2}^2+i\eta)}, \frac{1}{(\text{q1}^2+i\eta)}, \frac{1}{((p-\text{q1})^2+i\eta)}, \frac{1}{((p-\text{q1}+\text{q2})^2+i\eta)}\right\}, \{\text{q1}, \text{q2}\}, \{p\}, \{\}, \{\}\right)\right\}$$

Yet this time we have some preferred set of topologies and want to match to them (if possible)

```
preferredTopos2 = {FCTopology[prop2L, {SFAD[{{q1, 0}, {0, 1}, 1}],
     SFAD[{{q2, 0}, {0, 1}, 1}], SFAD[{{q1 - q2, 0}, {0, 1}, 1}], SFAD[{{-p
+ q1, 0}, {0, 1}, 1}],
     SFAD[{{-p + q2, 0}, {0, 1}, 1}]}, {q1, q2}, {p}, {}, {}],
   FCTopology[prop2LX1, {SFAD[{{q2, 0}, {0, 1}, 1}], SFAD[{{q1 - q2, 0},
{0, 1}, 1}],
     SFAD[{{-p + q1, 0}, {0, 1}, 1}], SFAD[{{-p + q2, 0}, {0, 1}, 1}]},
{q1, q2}, {p}, {}, {}],
   FCTopology[prop2LX3, {SFAD[{{q1, 0}, {0, 1}, 1}], SFAD[{{q2, 0}, {0, 1},
1}],
     SFAD[{{-p + q1, 0}, {0, 1}, 1}], SFAD[{{-p + q2, 0}, {0, 1}, 1}]},
{q1, q2}, {p}, {}, {}],
   FCTopology[prop2LX15, {SFAD[{{q2, 0}, {0, 1}, 1}], SFAD[{{q1 - q2, 0},
{0, 1}, 1}],
     SFAD[{{-p + q1, 0}, {0, 1}, 1}]}, {q1, q2}, {p}, {}, {}]}
```

$$\left\{\text{FCTopology}\left(\text{prop2L}, \left\{\frac{1}{(\text{q1}^2+i\eta)}, \frac{1}{(\text{q2}^2+i\eta)}, \frac{1}{((\text{q1}-\text{q2})^2+i\eta)}, \frac{1}{((\text{q1}-p)^2+i\eta)}, \frac{1}{((\text{q2}-p)^2+i\eta)}\right\}, \{\text{q1}, \text{q2}\}, \{p\}, \{\}, \{\}\right), \text{FCTopology}\left(\text{prop2LX1}, \left\{\frac{1}{(\text{q2}^2+i\eta)}, \frac{1}{((\text{q1}-\text{q2})^2+i\eta)}, \frac{1}{((\text{q1}-p)^2+i\eta)}, \frac{1}{((\text{q2}-p)^2+i\eta)}\right\}, \{\text{q1}, \text{q2}\}, \{p\}, \{\}, \{\}\right), \text{FCTopology}\left(\text{prop2LX3}, \left\{\frac{1}{(\text{q1}^2+i\eta)}, \frac{1}{(\text{q2}^2+i\eta)}, \frac{1}{((\text{q1}-p)^2+i\eta)}, \frac{1}{((\text{q2}-p)^2+i\eta)}\right\}, \{\text{q1}, \text{q2}\}, \{p\}, \{\}, \{\}\right), \text{FCTopology}\left(\text{prop2LX15}, \left\{\frac{1}{(\text{q2}^2+i\eta)}, \frac{1}{((\text{q1}-\text{q2})^2+i\eta)}, \frac{1}{((\text{q1}-p)^2+i\eta)}\right\}, \{\text{q1}, \text{q2}\}, \{p\}, \{\}, \{\}\right)\right\}$$

```
mappings2 = FCLoopFindTopologyMappings[topos2, PreferredTopologies ->
preferredTopos2];
```

```
mappings2[[1]]
```

$$\left(\begin{array}{c} \text{FCTopology}\left(\text{fctopology1}, \left\{\frac{1}{(\text{q2}^2+i\eta)}, \frac{1}{(\text{q1}^2+i\eta)}, \frac{1}{((\text{q1}+\text{q2})^2+i\eta)}, \frac{1}{((p+\text{q1})^2+i\eta)}, \frac{1}{((p-\text{q2})^2+i\eta)}\right\}, \{\text{q1}, \text{q2}\}, \{p\}, \{\}, \{\}\right) \\ \text{FCTopology}\left(\text{fctopology2}, \left\{\frac{1}{(\text{q2}^2+i\eta)}, \frac{1}{(\text{q1}^2+i\eta)}, \frac{1}{((p+\text{q2})^2+i\eta)}, \frac{1}{((p-\text{q1})^2+i\eta)}\right\}, \{\text{q1}, \text{q2}\}, \{p\}, \{\}, \{\}\right) \\ \text{FCTopology}\left(\text{fctopology3}, \left\{\frac{1}{(\text{q2}^2+i\eta)}, \frac{1}{(\text{q1}^2+i\eta)}, \frac{1}{((p-\text{q1})^2+i\eta)}, \frac{1}{((p-\text{q1}+\text{q2})^2+i\eta)}\right\}, \{\text{q1}, \text{q2}\}, \{p\}, \{\}, \{\}\right) \end{array}\right)$$

And these are the final occurring topologies

```
mappings2[[2]]
```

$$\left\{\text{FCTopology}\left(\text{prop2L}, \left\{\frac{1}{(\text{q1}^2+i\eta)}, \frac{1}{(\text{q2}^2+i\eta)}, \frac{1}{((\text{q1}-\text{q2})^2+i\eta)}, \frac{1}{((\text{q1}-p)^2+i\eta)}, \frac{1}{((\text{q2}-p)^2+i\eta)}\right\}, \{\text{q1}, \text{q2}\}, \{p\}, \{\}, \{\}\right), \text{FCTopology}\left(\text{prop2LX1}, \left\{\frac{1}{(\text{q2}^2+i\eta)}, \frac{1}{((\text{q1}-\text{q2})^2+i\eta)}, \frac{1}{((\text{q1}-p)^2+i\eta)}, \frac{1}{((\text{q2}-p)^2+i\eta)}\right\}, \{\text{q1}, \text{q2}\}, \{p\}, \{\}, \{\}\right), \text{FCTopology}\left(\text{prop2LX3}, \left\{\frac{1}{(\text{q1}^2+i\eta)}, \frac{1}{(\text{q2}^2+i\eta)}, \frac{1}{((\text{q1}-p)^2+i\eta)}, \frac{1}{((\text{q2}-p)^2+i\eta)}\right\}, \{\text{q1}, \text{q2}\}, \{p\}, \{\}, \{\}\right)\right\}$$

If we need to match subtopologies into larger topologies, we first need to generate all possible subtopologies for each relevant topology.

```
topos3 = {
   FCTopology[fctopology1, {
     SFAD[{{l1 + l2 - q1, 0}, {0, 1}, 1}],
     SFAD[{{l2, 0}, {SMP["m_t"]^2, 1}, 1}],
     SFAD[{{l1, 0}, {SMP["m_t"]^2, 1}, 1}],
     SFAD[{{l2 + q2, 0}, {SMP["m_t"]^2, 1}, 1}],
     SFAD[{{l1 - q1, 0}, {SMP["m_t"]^2, 1}, 1}],
     SFAD[{{l1 - q1 - q2, 0}, {SMP["m_t"]^2, 1}, 1}]}, {l1, l2}, {q1, q2},
{}, {}],
   FCTopology[fctopology9, {
     SFAD[{{l1 + l2 + q2, 0}, {0, 1}, 1}],
     SFAD[{{l2, 0}, {SMP["m_t"]^2, 1}, 1}],
     SFAD[{{l1, 0}, {SMP["m_t"]^2, 1}, 1}],
     SFAD[{{l1 + q2, 0}, {SMP["m_t"]^2, 1}, 1}],
     SFAD[{{l1 - q1, 0}, {SMP["m_t"]^2, 1}, 1}]}, {l1, l2}, {q1, q2}, {},
{}]
   }
```

$$\left\{\text{FCTopology}\left(\text{fctopology1}, \left\{\frac{1}{((\text{l1}+\text{l2}-\text{q1})^2+i\eta)}, \frac{1}{(\text{l2}^2-m_t^2+i\eta)}, \frac{1}{(\text{l1}^2-m_t^2+i\eta)}, \frac{1}{((\text{l2}+\text{q2})^2-m_t^2+i\eta)}, \frac{1}{((\text{l1}-\text{q1})^2-m_t^2+i\eta)}, \frac{1}{((\text{l1}-\text{q1}-\text{q2})^2-m_t^2+i\eta)}\right\}, \{\text{l1},\text{l2}\}, \{\text{q1},\text{q2}\}, \{\}, \{\}\right), \text{FCTopology}\left(\text{fctopology9}, \left\{\frac{1}{((\text{l1}+\text{l2}+\text{q2})^2+i\eta)}, \frac{1}{(\text{l2}^2-m_t^2+i\eta)}, \frac{1}{(\text{l1}^2-m_t^2+i\eta)}, \frac{1}{((\text{l1}+\text{q2})^2-m_t^2+i\eta)}, \frac{1}{((\text{l1}-\text{q1})^2-m_t^2+i\eta)}\right\}, \{\text{l1},\text{l2}\}, \{\text{q1},\text{q2}\}, \{\}, \{\}\right)\right\}$$

```
subTopos3 = Flatten[FCLoopFindSubtopologies[topos3]];
```

```
subTopos3 // Length
```


Now we can match a smaller topology into a larger topology

```
mappings3 = FCLoopFindTopologyMappings[topos3, PreferredTopologies ->
subTopos3];
```

```
mappings3[[1]]
```

$$\left(\text{FCTopology}\left(\text{fctopology9}, \left\{ \frac{1}{((\text{l1}+\text{l2}+\text{q2})^2+i\eta)}, \frac{1}{(\text{l2}^2-m_t^2+i\eta)}, \frac{1}{(\text{l1}^2-m_t^2+i\eta)}, \frac{1}{((\text{l1}+\text{q2})^2-m_t^2+i\eta)}, \frac{1}{((\text{l1}-\text{q1})^2-m_t^2+i\eta)} \right\}, \{\text{l1},\right.\right.$$

```
mappings3[[2]]
```

$$\left\{\text{FCTopology}\left(\text{fctopology1}, \left\{ \frac{1}{((\text{l1}+\text{l2}-\text{q1})^2+i\eta)}, \frac{1}{(\text{l2}^2-m_t^2+i\eta)}, \right.\right.\right.$$
$$\frac{1}{(\text{l1}^2-m_t^2+i\eta)}, \frac{1}{((\text{l2}+\text{q2})^2-m_t^2+i\eta)}, \frac{1}{((\text{l1}-\text{q1})^2-m_t^2+i\eta)},$$
$$\left.\left.\left. \frac{1}{((\text{l1}-\text{q1}-\text{q2})^2-m_t^2+i\eta)} \right\}, \{\text{l1},\text{l2}\}, \{\text{q1},\text{q2}\}, \{\}, \{\} \right) \right\}$$

Mapping the following two topologies onto each other requires shifts in the external momenta due to the chosen kinematic constraints.

```
topos4 = {
   FCTopology[topo1, {
     SFAD[{{l1 + q1, 0}, {m^2, 1}, 1}],
     SFAD[{{l1 - l2, 0}, {0, 1}, 1}],
     SFAD[{{l2 + q1, 0}, {m^2, 1}, 1}],
     SFAD[{{l2 - q2, 0}, {m^2, 1}, 1}],
     SFAD[{{l2, 0}, {0, 1}, 1}]}, {l1, l2}, {q1, q2}, {SPD[q1, q1] -> 0,
SPD[q2, q2] -> 0, SPD[q1, q2] -> s/2}, {}],
   FCTopology[topo2, {
     SFAD[{{l1 - l2, 0}, {m^2, 1}, 1}],
     SFAD[{{l1 - q2, 0}, {0, 1}, 1}],
     SFAD[{{l2 - q2, 0}, {m^2, 1}, 1}],
     SFAD[{{l2 + q1, 0}, {m^2, 1}, 1}],
     SFAD[{{l2, 0}, {0, 1}, 1}]}, {l1, l2}, {q1, q2}, {SPD[q1, q1] -> 0,
SPD[q2, q2] -> 0, SPD[q1, q2] -> s/2}, {}]}
```

$$\left\{\text{FCTopology}\left(\text{topo1}, \left\{ \frac{1}{((\text{l1}+\text{q1})^2-m^2+i\eta)}, \frac{1}{((\text{l1}-\text{l2})^2+i\eta)}, \frac{1}{((\text{l2}+\text{q1})^2-m^2+i\eta)}, \right.\right.$$
$$\left. \frac{1}{((\text{l2}-\text{q2})^2-m^2+i\eta)}, \frac{1}{(\text{l2}^2+i\eta)} \right\}, \{\text{l1},\text{l2}\}, \{\text{q1},\text{q2}\}, \left\{\text{q1}^2 \to 0, \text{q2}^2 \to 0, \text{q1}\cdot\ \text{q2} \to \frac{s}{2}\right\},$$
$$\left.\{\} \right), \text{FCTopology}\left(\text{topo2}, \left\{ \frac{1}{((\text{l1}-\text{l2})^2-m^2+i\eta)}, \frac{1}{((\text{l1}-\text{q2})^2+i\eta)}, \frac{1}{((\text{l2}-\text{q2})^2-m^2+i\eta)}, \right.\right.$$
$$\left.\left. \frac{1}{((\text{l2}+\text{q1})^2-m^2+i\eta)}, \frac{1}{(\text{l2}^2+i\eta)} \right\}, \{\text{l1},\text{l2}\}, \{\text{q1},\text{q2}\}, \left\{\text{q1}^2 \to 0, \text{q2}^2 \to 0, \text{q1}\cdot\ \text{q2} \to \frac{s}{2}\right\}, \{\} \right) \right\}$$

```
mappings4 = FCLoopFindTopologyMappings[topos4, Momentum -> All];
```

```
mappings4[[1]]
```

$$\left(\text{FCTopology}\left(\text{topo2}, \left\{ \frac{1}{((\text{l1}-\text{l2})^2-m^2+i\eta)}, \frac{1}{((\text{l1}-\text{q2})^2+i\eta)}, \frac{1}{((\text{l2}-\text{q2})^2-m^2+i\eta)}, \frac{1}{((\text{l2}+\text{q1})^2-m^2+i\eta)}, \frac{1}{(\text{l2}^2+i\eta)} \right\}, \{\text{l1}, \text{l2}\}, \{\text{q1}, \right.$$

Otherwise no mappings exist

```
FCLoopFindTopologyMappings[topos4][[1]]
```

$$\{\}$$

9.39 FCLoopPakOrder

`FCLoopPakOrder[poly, {x1, x2, ...}]` determines a canonical ordering of the Feynman parameters **`x1, x2, ...`** in the polynomial **`poly`**.

The function uses the algorithm of Alexey Pak arXiv:1111.0868. Cf. also the PhD thesis of Jens Hoff 10.5445/IR/1000047447 for the detailed description of a possible implementation.

The current implementation is based on the **`PolyOrdering`** function from FIRE 6 arXiv:1901.07808

The function can also directly perform the renaming of the Feynman parameter variables returning the original polynomial in the canonical form. This is done by setting the option **`Rename`** to **`True`**.

9.39.1 See also

Overview, FCTopology, GLI, FCLoopToPakForm, FCLoopPakOrder.

9.39.2 Examples

Canonicalizing a polynomial

Let us consider the following product of **`U`** and **`F`** polynomials of some loop integral

```
poly = (x[1]*x[2] + x[1]*x[3] + x[2]*x[3] + x[2]*x[4] + x[3]*x[4] +
x[1]*x[5] +
     x[2]*x[5] + x[4]*x[5])* (m1^2*x[1]^2*x[2] + m3^2*x[1]*x[2]^2 +
     m1^2*x[1]^2*x[3] +
     m1^2*x[1]*x[2]*x[3] + m2^2*x[1]*x[2]*x[3] +   m3^2*x[1]*x[2]*x[3] +
     m3^2*x[2]^2*x[3] + m2^2*x[1]*x[3]^2 + m2^2*x[2]*x[3]^2 +
m1^2*x[1]*x[2]*x[4] -
      SPD[q, q]*x[1]*x[2]*x[4] + m3^2*x[2]^2*x[4] + m1^2*x[1]*x[3]*x[4] -
      SPD[q, q]*x[1]*x[3]*x[4] +   m2^2*x[2]*x[3]*x[4] +
m3^2*x[2]*x[3]*x[4] -
```

```
      SPD[q, q]*x[2]*x[3]*x[4] + m2^2*x[3]^2*x[4] + m1^2*x[1]^2*x[5] +
      m1^2*x[1]*x[2]*x[5] + m3^2*x[1]*x[2]*x[5] - SPD[q, q]*x[1]*x[2]*x[5]
+
      m3^2*x[2]^2*x[5] + m2^2*x[1]*x[3]*x[5] -   SPD[q, q]*x[1]*x[3]*x[5] +
      m2^2*x[2]*x[3]*x[5] - SPD[q, q]*x[2]*x[3]*x[5] + m1^2*x[1]*x[4]*x[5]
-
      SPD[q, q]*x[1]*x[4]*x[5] + m3^2*x[2]*x[4]*x[5] + m2^2*x[3]*x[4]*x[5]
-
      SPD[q, q]*x[3]*x[4]*x[5])
```

$$(x(1)x(2)+x(3)x(2)+x(4)x(2)+x(5)x(2)+x(1)x(3)+x(3)x(4)+x(1)x(5)+x(4)x(5))\left(\text{m1}^2x(1)^2x(2)+\text{m1}^2x(1)^2x(3)+\text{m1}^2x(1)x(2)x(3)+\text{m1}^2x(1)x(2)x(4)+\text{m1}^2x(1)x(3)x(4)+\text{m1}^2x(1)^2x(5)+\text{m1}^2x(1)x(2)x(5)+\text{m1}^2x(1)x(4)x(5)+\text{m2}^2x(1)x(3)^2+\text{m2}^2x(2)x(3)^2+\text{m2}^2x(1)x(2)x(3)+\text{m2}^2x(3)^2x(4)+\text{m2}^2x(2)x(3)x(4)+\text{m2}^2x(1)x(3)x(5)+\text{m2}^2x(2)x(3)x(5)+\text{m2}^2x(3)x(4)x(5)+\text{m3}^2x(1)x(2)^2+\text{m3}^2x(2)^2x(3)+\text{m3}^2x(1)x(2)x(3)+\text{m3}^2x(2)^2x(4)+\text{m3}^2x(2)x(3)x(4)+\text{m3}^2x(2)^2x(5)+\text{m3}^2x(1)x(2)x(5)+\text{m3}^2x(2)x(4)x(5)-q^2x(1)x(2)x(4)-q^2x(1)x(3)x(4)-q^2x(2)x(3)x(4)-q^2x(1)x(2)x(5)-q^2x(1)x(3)x(5)-q^2x(2)x(3)x(5)-q^2x(1)x(4)x(5)-q^2x(3)x(4)x(5)\right)$$

Using **FCLoopPakOrder** we can obtain a canonical ordering for this polynomial

```
sigma = FCLoopPakOrder[poly, x]
```

$$\begin{pmatrix} 1 & 3 & 2 & 5 & 4 \end{pmatrix}$$

This output implies that the polynomial will become canonically ordered upon renaming the Feynman parameter variables as follows

```
fpVars = Table[x[i], {i, 1, 5}]
```

$$\{x(1), x(2), x(3), x(4), x(5)\}$$

```
repRule = Thread[Rule[Extract[fpVars, List /@ First[sigma]], fpVars]]
```

$$\{x(1) \to x(1), x(3) \to x(2), x(2) \to x(3), x(5) \to x(4), x(4) \to x(5)\}$$

This way we obtain the canonical form of our polynomial **poly**

```
poly /. repRule
```

$$(x(1)x(2)+x(3)x(2)+x(5)x(2)+x(1)x(3)+x(1)x(4)+x(3)x(4)+x(3)x(5)+x(4)x(5))\left(\text{m1}^2x(1)^2x(2)+\text{m1}^2x(1)^2x(3)+\text{m1}^2x(1)x(2)x(3)+\text{m1}^2x(1)^2x(4)+\text{m1}^2x(1)x(3)x(4)+\text{m1}^2x(1)x(2)x(5)+\text{m1}^2x(1)x(3)x(5)+\text{m1}^2x(1)x(4)x(5)+\text{m2}^2x(1)x(2)^2+\text{m2}^2x(2)^2x(3)+\text{m2}^2x(1)x(2)x(3)+\text{m2}^2x(1)x(2)x(4)+\text{m2}^2x(2)x(3)x(4)+\text{m2}^2x(2)^2x(5)+\text{m2}^2x(2)x(3)x(5)+\text{m2}^2x(2)x(4)x(5)+\text{m3}^2x(1)x(3)^2+\text{m3}^2x(2)x(3)^2+\text{m3}^2x(1)x(2)x(3)+\text{m3}^2x(3)^2x(4)+\text{m3}^2x(1)x(3)x(4)+\text{m3}^2x(3)^2x(5)+\text{m3}^2x(2)x(3)x(5)+\text{m3}^2x(3)x(4)x(5)-q^2x(1)x(2)x(4)-q^2x(1)x(3)x(4)-q^2x(2)x(3)x(4)-q^2x(1)x(2)x(5)-q^2x(1)x(3)x(5)-q^2x(2)x(3)x(5)-q^2x(1)x(4)x(5)-q^2x(2)x(4)x(5)\right)$$

Checking equivalence

Let us consider the following two polynomials

```
poly1 = -1/4*(x[2]^2*x[3]) - (x[1]^2*x[4])/4 -
   (x[1]^2*x[5])/4 + (x[1]*x[2]*x[5])/2 - (x[2]^2*x[5])/4 + x[3]*x[4]*x[5]
```

$$-\frac{1}{4}x(4)x(1)^2-\frac{1}{4}x(5)x(1)^2+\frac{1}{2}x(2)x(5)x(1)-\frac{1}{4}x(2)^2x(3)-\frac{1}{4}x(2)^2x(5)+x(3)x(4)x(5)$$

```
poly2 = -1/4*(x[1]^2*x[2]) - (x[1]^2*x[3])/4 +
   x[2]*x[3]*x[4] + (x[1]*x[3]*x[5])/2 - (x[3]*x[5]^2)/4 - (x[4]*x[5]^2)/4
```

$$-\frac{1}{4}x(2)x(1)^2-\frac{1}{4}x(3)x(1)^2+\frac{1}{2}x(3)x(5)x(1)-\frac{1}{4}x(3)x(5)^2-\frac{1}{4}x(4)x(5)^2+x(2)x(3)x(4)$$

These polynomials are not identical

```
poly1 === poly2
```

False

However, one can easily recognize that they are actually the same upon renaming Feynman parameters **x[i]** in a suitable way. **FCLoopPakOrder** can do such renamings automatically

```
canoPoly1 = FCLoopPakOrder[poly1, x, Rename -> True]

canoPoly2 = FCLoopPakOrder[poly2, x, Rename -> True]
```

$$-\frac{1}{4}x(3)x(1)^2 - \frac{1}{4}x(5)x(1)^2 + \frac{1}{2}x(2)x(3)x(1) - \frac{1}{4}x(2)^2x(3) - \frac{1}{4}x(2)^2x(4) + x(3)x(4)x(5)$$

$$-\frac{1}{4}x(3)x(1)^2 - \frac{1}{4}x(5)x(1)^2 + \frac{1}{2}x(2)x(3)x(1) - \frac{1}{4}x(2)^2x(3) - \frac{1}{4}x(2)^2x(4) + x(3)x(4)x(5)$$

When comparing the canonicalized versions of both polynomials we see that they are indeed identical

```
canoPoly1 === canoPoly2
```

True

9.40 FCLoopToPakForm

FCLoopToPakForm[int, {p1, p2, ...}] determines a canonical UF-based representation for the scalar multi-loop integral **int** that depend on the loop momenta **p1, p2, ...** using the algorithm of Alexey Pak arXiv:1111.0868.

The current implementation is based on the **FindEquivalents** function from FIRE 6 arXiv:1901.07808. **FCLoopToPakForm** is a backend function used in **FCLoopPakScalelessQ**, **FCLoopFindIntegralMappings**, **FCLoopFindTopologyMappings** etc.

It is also possible to invoke the function as **FCLoopToPakForm[GLI[...], FCTopology[...]]** or FCLoopToPakForm[FCTopology[...]]. Notice that in this case the value of the option **FinalSubstitutions** is ignored, as replacement rules will be extracted directly from the definition of the topology.

9.40.1 See also

Overview, FCTopology, GLI, FCLoopToPakForm, FCLoopPakScalelessQ, FCLoopScalelessQ, FCLoopFindIntegralMappings, FCLoopFindTopologyMappings.

9.40.2 Examples

```
FCLoopToPakForm[FAD[p1, {p3, m1}, {p1 - p4, m1}, p1 + q1, p1 + q1, p3 + q1,
p1 - p3 - p4],
  {p1, p3, p4}, Names -> x, Head -> ph, Power -> pow]
```

$$\left\{\frac{1}{\text{p1}^2.\left(\text{p3}^2-\text{m1}^2\right).\left((\text{p1}-\text{p4})^2-\text{m1}^2\right).(\text{p1}+\text{q1})^4.(\text{p3}+\text{q1})^2.(\text{p1}-\text{p3}-\text{p4})^2},\right.$$

$$\text{ph}\left(\text{m1}^2\ \text{pow}(2)x(2)x(4)^2x(6)+\text{m1}^2\ \text{pow}(2)x(3)x(4)^2x(6)+\text{m1}^2\ \text{pow}(2)x(2)^2x(3)x(6)\right.$$
$$+\text{m1}^2\ \text{pow}(2)x(2)^2x(4)x(6)+2\ \text{m1}^2\ \text{pow}(2)x(2)x(3)x(4)x(6)+\text{m1}^2\ \text{pow}(2)x(2)^2x(5)x(6)$$
$$+\text{m1}^2\ \text{pow}(2)x(2)x(3)x(5)x(6)+\text{m1}^2\ \text{pow}(2)x(2)x(4)x(5)x(6)+\text{m1}^2\ \text{pow}(2)x(3)x(4)x(5)x(6)$$
$$+\text{m1}^2x(1)x(2)x(4)^2+\text{m1}^2x(1)x(3)x(4)^2+\text{m1}^2x(1)x(2)^2x(3)+\text{m1}^2x(1)x(2)^2x(4)+2\ \text{m1}^2x(1)x(2)x(3)x(4)$$
$$+\text{m1}^2x(1)x(2)^2x(5)+\text{m1}^2x(1)x(2)x(3)x(5)+\text{m1}^2x(1)x(2)x(4)x(5)+\text{m1}^2x(1)x(3)x(4)x(5)$$
$$-\text{pow}(2)\ \text{q1}^2x(1)x(2)x(3)x(6)-\text{pow}(2)\ \text{q1}^2x(1)x(2)x(4)x(6)-\text{pow}(2)\ \text{q1}^2x(1)x(3)x(4)x(6)$$
$$-\text{pow}(2)\ \text{q1}^2x(1)x(2)x(5)x(6)-\text{pow}(2)\ \text{q1}^2x(1)x(3)x(5)x(6)-\text{pow}(2)\ \text{q1}^2x(2)x(3)x(5)x(6)$$
$$-\text{pow}(2)\ \text{q1}^2x(2)x(4)x(5)x(6)-\text{pow}(2)\ \text{q1}^2x(3)x(4)x(5)x(6)+\text{pow}(2)x(2)x(3)x(6)+\text{pow}(2)x(2)x(4)x(6)$$
$$+\text{pow}(2)x(3)x(4)x(6)+\text{pow}(2)x(2)x(5)x(6)+\text{pow}(2)x(3)x(5)x(6)-\text{q1}^2x(1)x(2)x(3)x(5)$$
$$-\text{q1}^2x(1)x(2)x(4)x(5)-\text{q1}^2x(1)x(3)x(4)x(5)+x(1)x(2)x(3)+x(1)x(2)x(4)+x(1)x(3)x(4)+x(1)x(2)x(5)$$
$$\left.\left.+x(1)x(3)x(5),\begin{pmatrix} x(1) & x(3) & x(4) & x(2) & x(6) & x(5) \\ \frac{1}{\text{p1}^2} & \frac{1}{(\text{p1}-\text{p4})^2-\text{m1}^2} & \frac{1}{(\text{p1}-\text{p3}-\text{p4})^2} & \frac{1}{\text{p3}^2-\text{m1}^2} & \frac{1}{(\text{p3}+\text{q1})^2} & \frac{1}{(\text{p1}+\text{q1})^2} \\ 1 & 1 & 1 & 1 & 1 & 2 \end{pmatrix}\right)\right\}$$

```
topo1 = FCTopology["prop2Lv1", {SFAD[{p1, m1^2}], SFAD[{p2, m2^2}], SFAD[p1
- q], SFAD[p2 - q], SFAD[{p1 - p2, m3^2}]}, {p1, p2}, {Q}, {}, {}]

topo2 = FCTopology["prop2Lv2", {SFAD[{p1, m1^2}], SFAD[{p2, m2^2}],
SFAD[{p1 - q, M^2}], SFAD[{p2 - q, M^2}], SFAD[p1 - p2]}, {p1, p2}, {Q},
{}, {}]
```

$$\text{FCTopology}\left(\text{prop2Lv1},\left\{\frac{1}{(\text{p1}^2-\text{m1}^2+i\eta)},\frac{1}{(\text{p2}^2-\text{m2}^2+i\eta)},\frac{1}{((\text{p1}-q)^2+i\eta)},\right.\right.$$
$$\left.\left.\frac{1}{((\text{p2}-q)^2+i\eta)},\frac{1}{((\text{p1}-\text{p2})^2-\text{m3}^2+i\eta)}\right\},\{\text{p1},\text{p2}\},\{Q\},\{\},\{\}\right)$$

$$\text{FCTopology}\left(\text{prop2Lv2},\left\{\frac{1}{(\text{p1}^2-\text{m1}^2+i\eta)},\frac{1}{(\text{p2}^2-\text{m2}^2+i\eta)},\frac{1}{((\text{p1}-q)^2-M^2+i\eta)},\right.\right.$$
$$\left.\left.\frac{1}{((\text{p2}-q)^2-M^2+i\eta)},\frac{1}{((\text{p1}-\text{p2})^2+i\eta)}\right\},\{\text{p1},\text{p2}\},\{Q\},\{\},\{\}\right)$$

```
FCLoopToPakForm[topo1, Names -> x, Head -> ph, Power -> pow]
```

$$\left\{\text{FCTopology}\left(\text{prop2Lv1},\left\{\frac{1}{(\text{p1}^2-\text{m1}^2+i\eta)},\frac{1}{(\text{p2}^2-\text{m2}^2+i\eta)},\frac{1}{((\text{p1}-q)^2+i\eta)},\right.\right.\right.$$
$$\left.\left.\left.\frac{1}{((\text{p2}-q)^2+i\eta)},\frac{1}{((\text{p1}-\text{p2})^2-\text{m3}^2+i\eta)}\right\},\{\text{p1},\text{p2}\},\{Q\},\{\},\{\}\right),$$
$$\text{ph}\left(\text{m1}^2x(1)x(2)^2+\text{m1}^2x(1)x(2)x(3)+\text{m1}^2x(2)^2x(4)+\text{m1}^2x(1)x(2)x(4)\right.$$
$$+\text{m1}^2x(2)x(3)x(4)+\text{m1}^2x(2)^2x(5)+\text{m1}^2x(1)x(2)x(5)+\text{m1}^2x(2)x(3)x(5)$$
$$+\text{m2}^2x(1)x(4)^2+\text{m2}^2x(2)x(4)^2+\text{m2}^2x(3)x(4)^2+\text{m2}^2x(1)x(2)x(4)+\text{m2}^2x(1)x(3)x(4)$$
$$+\text{m2}^2x(1)x(4)x(5)+\text{m2}^2x(2)x(4)x(5)+\text{m2}^2x(3)x(4)x(5)+\text{m3}^2x(1)^2x(2)+\text{m3}^2x(1)^2x(3)$$
$$+\text{m3}^2x(1)^2x(4)+\text{m3}^2x(1)x(2)x(4)+\text{m3}^2x(1)x(3)x(4)+\text{m3}^2x(1)^2x(5)+\text{m3}^2x(1)x(2)x(5)$$
$$+\text{m3}^2x(1)x(3)x(5)-q^2x(1)x(2)x(3)-q^2x(1)x(3)x(4)-q^2x(2)x(3)x(4)-q^2x(1)x(2)x(5)$$
$$-q^2x(2)x(3)x(5)-q^2x(1)x(4)x(5)-q^2x(2)x(4)x(5)-q^2x(3)x(4)x(5)+x(1)x(2)$$
$$+x(1)x(3)+x(1)x(4)+x(2)x(4)+x(3)x(4)+x(1)x(5)+x(2)x(5)+x(3)x(5),$$
$$\left.\left.\begin{pmatrix} x(5) & x(1) & x(3) & x(2) & x(4) \\ \frac{1}{((\text{p1}-\text{p2})^2-\text{m3}^2+i\eta)} & \frac{1}{(\text{p1}^2-\text{m1}^2+i\eta)} & \frac{1}{((\text{p1}-q)^2+i\eta)} & \frac{1}{(\text{p2}^2-\text{m2}^2+i\eta)} & \frac{1}{((\text{p2}-q)^2+i\eta)} \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}\right)\right)\right\}$$

```
FCLoopToPakForm[{GLI["prop2Lv1", {1, 1, 1, 1, 0}], GLI["prop2Lv2", {1, 1,
0, 0, 1}]}, {topo1, topo2}, Names -> x, Head -> ph, Power -> pow]
```

$$\begin{pmatrix} G^{\text{prop2Lv1}}(1,1,1,1,0) & \text{ph}\left(\text{m1}^2x(1)^2x(3)+\text{m1}^2x(1)x(2)x(3)+\text{m1}^2x(1)^2x(4)+\text{m1}^2x(1)x(2)x(4)+\text{m2}^2x(1)x(3)^2+\right. \\ G^{\text{prop2Lv2}}(1,1,0,0,1) & \text{ph}\left(\text{m1}^2x(1)^2x(2)+\text{m1}\right. \end{pmatrix}$$

Products of **GLI**s are also supported.

```
FCLoopToPakForm[{GLI["prop2Lv1", {1, 1, 0, 0, 0}]^2}, {topo1, topo2}, Names
-> x, Head -> ph, Power -> pow]
```

$$\begin{pmatrix} G^{\text{prop2Lv1}}(1,1,0,0,0)^2 & \text{ph}\left(\text{m1}^2x(2)x(3)x(4)x(1)^2+\text{m1}^2x(2)^2x(3)x(4)x(1)+\text{m2}^2x(2)x(3)x(4)^2x(1)+\text{m2}^2x(2)x(3\right. \end{pmatrix}$$

9.41 FCGraphCuttableQ

FCGraphCuttableQ[{edges, labels}, {m1, m2, ...}] checks whether the given graph representing a loop integral can be cut such, that no propagator containing masses **{m1, m2, ...}** goes on shell. To that aim **labels** must contain masses occurring in the respective propagators.

FCGraphCuttableQ uses **FCGraphFindPath** as the back-end.

The list **{edges, labels}** can be the output of **FCLoopIntegralToGraph**.

9.41.1 See also

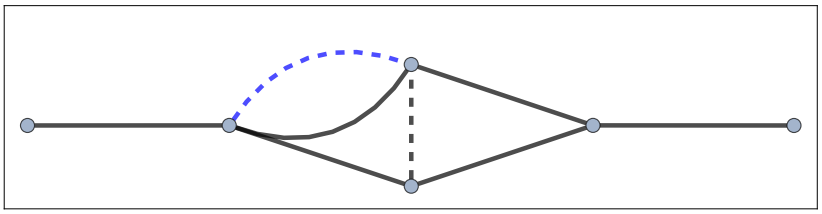
Overview, FCGraphFindPath, FCLoopIntegralToGraph, SameSideExternalEdges.

9.41.2 Examples

This integral has no imaginary part due to the massive **m1**-line that cannot be cut

```
graph1 = {{-3 -> 2, -1 -> 1, 1 -> 3, 1 -> 4, 2 -> 3, 2 -> 4, 2 -> 4, 3 ->
4}, {q1, q1, {p3, 1, m1^2}, {p3 + q1, 1, m1^2},
     {p2, 1, m1^2}, {p1 + q1, 1, m2^2}, {p1 - p2, 1, m1^2}, {p2 - p3, 1,
     0}}, {0, 0, SFAD[{{I*p3, 0}, {-m1^2, -1}, 1}],
     SFAD[{{I*p2, 0}, {-m1^2, -1}, 1}], SFAD[{{I*(p3 + q1), 0}, {-m1^2,
-1}, 1}], SFAD[{{I*(p1 + q1), 0},
       {-m2^2, -1}, 1}], SFAD[{{I*(p2 - p3), 0}, {0, -1}, 1}], SFAD[{{I*(p1
       - p2), 0}, {-m1^2, -1}, 1}]}, 1};
```

```
FCLoopGraphPlot[graph1, GraphPlot -> {MultiedgeStyle -> 0.35, Frame ->
True}, Style -> {
    {"InternalLine", _, _, mm_ /; ! FreeQ[mm, mg | m3]} -> {Red, Thick,
    Dashed},
    {"InternalLine", _, _, mm_ /; ! FreeQ[mm, mc | m2]} -> {Blue, Thick,
    Dashed},
    {"InternalLine", _, _, mm_ /; ! FreeQ[mm, mb | m1]} -> {Black, Thick}
   }]
```



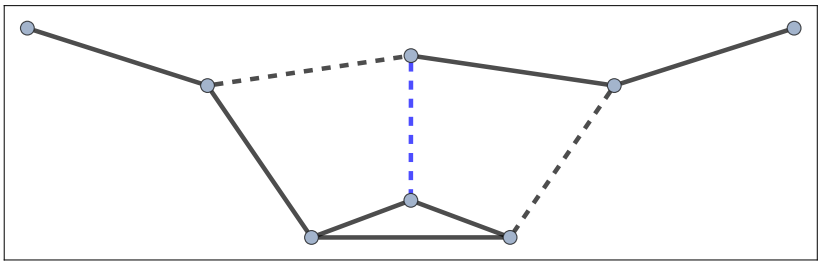
```
FCGraphCuttableQ[graph1, {m1}]
```

False

```
graph2 = {{-3 -> 2, -1 -> 1, 1 -> 3, 1 -> 4, 2 -> 4, 2 -> 5, 3 -> 5, 3 ->
6, 4 -> 6, 5 -> 6}, {q1, q1, {p3, 1, 0}, {p3 + q1, 1, m1^2},
     {p1 + q1, 1, 0}, {p1, 1, m1^2}, {p2, 1, m1^2}, {p2 - p3, 1, m1^5},
     {-p1 + p3, 1, m2^2}, {p1 - p2, 1, m1^2}},
    {0, 0, SFAD[{{I*p3, 0}, {0, -1}, 1}], SFAD[{{I*(p1 + q1), 0}, {0, -1},
    1}], SFAD[{{I*p2, 0},
       {-m1^2, -1}, 1}], SFAD[{{I*p1, 0}, {-m1^2, -1}, 1}], SFAD[{{I*(p3 +
       q1), 0}, {-m1^2, -1}, 1}],
     SFAD[{{I*(-p1 + p3), 0}, {-m2^2, -1}, 1}], SFAD[{{I*(p2 - p3), 0},
{-m1^5, -1}, 1}],
     SFAD[{{I*(p1 - p2), 0}, {-m1^2, -1}, 1}]}, 1};
```

This graph can be cut through the dashed blue and black lines, hence **`FCGraphCuttableQ`** returns **`True`**

```
FCLoopGraphPlot[graph2, GraphPlot -> {MultiedgeStyle -> 0.35, Frame ->
True}, Style -> {
    {"InternalLine", _, _, mm_ /; ! FreeQ[mm, mg | m3]} -> {Red, Thick,
    Dashed},
    {"InternalLine", _, _, mm_ /; ! FreeQ[mm, mc | m2]} -> {Blue, Thick,
    Dashed},
    {"InternalLine", _, _, mm_ /; ! FreeQ[mm, mb | m1]} -> {Black, Thick}
   }]
```



```
FCGraphCuttableQ[graph2, {m1}]
```

True

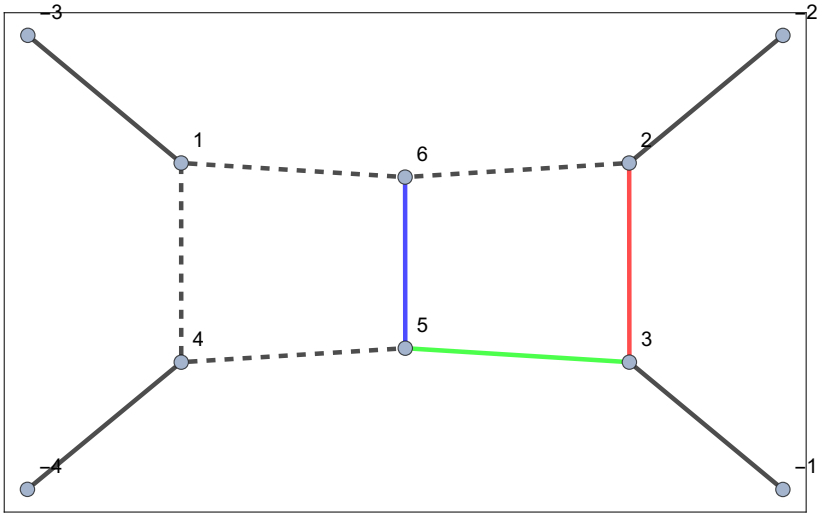
In the case of graphs with more than 2 external legs, the situation is somewhat more involved

```
graph3 = {{-4 -> 4, -3 -> 1, -2 -> 2, -1 -> 3, 1 -> 4, 1 -> 6, 2 -> 3, 2 ->
6, 3 -> 5, 4 -> 5, 5 -> 6},
   {Q1 - Q2 - Q3, Q1, Q2, Q3, {-p1 - p2 + Q2 + Q3, 1, 0}, {-p1 - p2 + Q2,
   1, 0}, {p1, 1, -m1^2}, {-p1 + Q2, 1, 0},
     {p1 + Q1, 1, -m3^2}, {p1 + p2 + Q1, 1, 0}, {p2, 1, -m2^2}}, {0, 0, 0,
     0, SFAD[{{p1 + p2 + Q1, 0}, {0, 1}, 1}],
```

```
     SFAD[{{-p1 + Q2, 0}, {0, 1}, 1}], SFAD[{{p2, 0}, {m2^2, 1}, 1}],
SFAD[{{p1, 0}, {m1^2, 1}, 1}],
     SFAD[{{p1 + Q1, 0}, {m3^2, 1}, 1}], SFAD[{{-p1 - p2 + Q2, 0}, {0, 1},
1}],
     SFAD[{{-p1 - p2 + Q2 + Q3, 0}, {0, 1}, 1}]}, 1}
```

$$\left\{\{-4\to 4,-3\to 1,-2\to 2,-1\to 3,1\to 4,1\to 6,2\to 3,2\to 6,3\to 5,4\to 5,5\to 6\},\right.$$
$$\{Q1-Q2-Q3,Q1,Q2,Q3,\{-p1-p2+Q2+Q3,1,0\},\{-p1-p2+Q2,1,0\},\{p1,1,$$
$$-m1^2\},\{Q2-p1,1,0\},\{p1+Q1,1,-m3^2\},\{p1+p2+Q1,1,0\},\{p2,1,-m2^2\}\},\left\{0,\right.$$
$$0,0,0,\frac{1}{((p1+p2+Q1)^2+i\eta)},\frac{1}{((Q2-p1)^2+i\eta)},\frac{1}{(p2^2-m2^2+i\eta)},\frac{1}{(p1^2-m1^2+i\eta)},$$
$$\left.\left.\frac{1}{((p1+Q1)^2-m3^2+i\eta)},\frac{1}{((-p1-p2+Q2)^2+i\eta)},\frac{1}{((-p1-p2+Q2+Q3)^2+i\eta)}\right\},1\right\}$$

```
FCLoopGraphPlot[graph3, GraphPlot -> {MultiedgeStyle -> 0.35, Frame ->
True, VertexLabels -> "Name"}, Style -> {
       {"InternalLine", _, _, mm_ /; ! FreeQ[mm, m1]} -> {Red, Thick},
       {"InternalLine", _, _, mm_ /; ! FreeQ[mm, m2]} -> {Blue, Thick},
       {"InternalLine", _, _, mm_ /; ! FreeQ[mm, m3]} -> {Green, Thick},
       {"InternalLine", _, _, mm_ /; ! FreeQ[mm, m4]} -> {Purple, Thick},
       {"ExternalLine", q1} -> {Brown, Thick, Dashed}
      }]
```



By default **FCGraphCuttableQ** thinks that the graph is not cuttable, since it choses the path connecting two external edges on the same side

```
FCGraphCuttableQ[graph3, {m1, m2, m3}]
```

False

```
FCGraphFindPath[graph3[[1]], {1, 1, 1, 1, -1, -1, 1, -1, 1, -1, 1}]
```

$$\left(\{-2 \to 2, 3\} \quad \{2 \to 3, 7\} \quad \{-1 \to 3, 4\} \right)$$

We can exclude such paths by letting the function know which external edges are on the same side via the option **SameSideExternalEdges**. In this case **FCGraphCuttableQ** correctly reports that the graph is cuttable

```
FCGraphCuttableQ[graph3, {m1, m2, m3}, SameSideExternalEdges -> {-2, -1}]
```

True

9.42 FCGraphFindPath

FCGraphFindPath[graph, weights] determines whether the given graph can be traversed by starting and finishing at one of the external edges.

The respective external edges must differ and **{1}** is returned for all graphs with less than two such edges, since tadpoles have no cut by definition.

The only supported weights are 1 and -1, with -1 meaning that the given edge cannot be passed.

Only directed graphs are supported but the direction of edges does not matter when searching for the path. The path is understood to be free of any cycles (loops).

9.42.1 See also

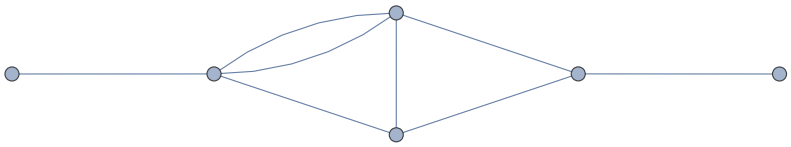
Overview, FCGraphCuttableQ, FCLoopIntegralToGraph, SameSideExternalEdges.

9.42.2 Examples

This integral has no imaginary part due to the massive **m1**-line that cannot be cut

```
graph1 = {-3 -> 2, -1 -> 1, 1 -> 3, 1 -> 4, 2 -> 3, 2 -> 4, 2 -> 4, 3 ->
4};
```

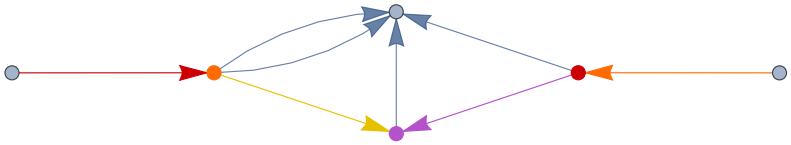
```
GraphPlot[graph1]
```



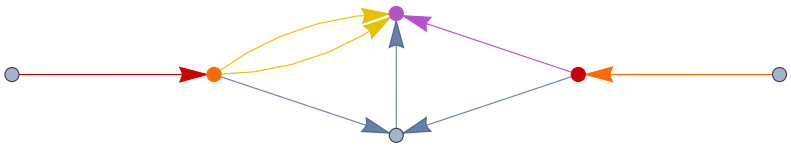
```
res1 = FCGraphFindPath[graph1, {1, 1, 1, 1, 1, -1, 1, -1}]
```

$$\begin{pmatrix} \{-3 \to 2, 1\} & \{2 \to 3, 5\} & \{1 \to 3, 3\} & \{-1 \to 1, 2\} \\ \{-3 \to 2, 1\} & \{2 \to 4, 7\} & \{1 \to 4, 4\} & \{-1 \to 1, 2\} \end{pmatrix}$$

```
HighlightGraph[graph1, res1[[1]], GraphLayout ->
"SpringElectricalEmbedding"]
```

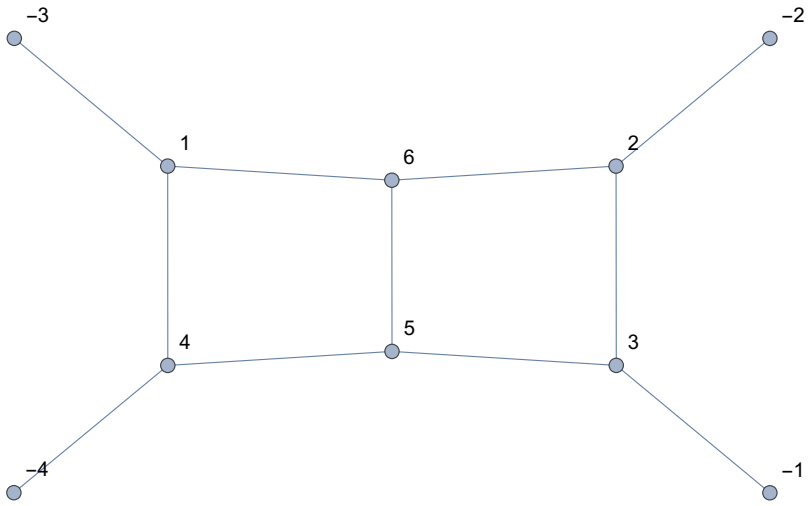


```
HighlightGraph[graph1, res1[[2]], GraphLayout ->
"SpringElectricalEmbedding"]
```



```
graph2 = {-4 -> 4, -3 -> 1, -2 -> 2, -1 -> 3, 1 -> 4, 1 -> 6, 2 -> 3,
    2 -> 6, 3 -> 5, 4 -> 5, 5 -> 6};
```

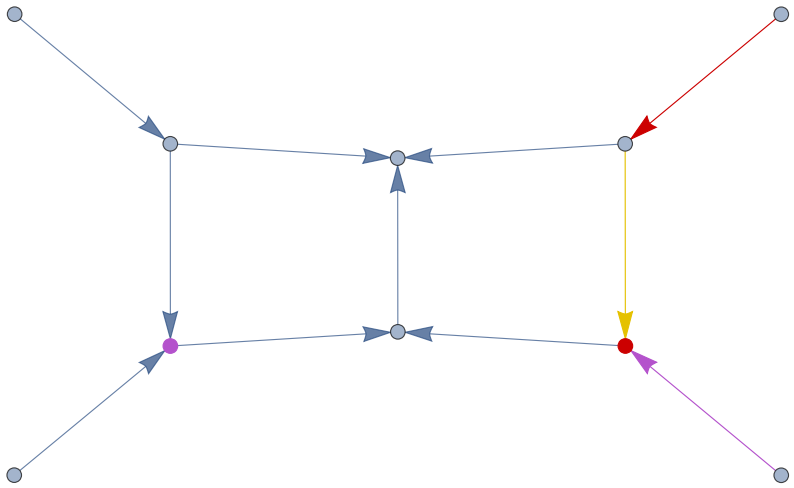
```
GraphPlot[graph2, VertexLabels -> "Name"]
```




```
res2 = FCGraphFindPath[graph2, {1, 1, 1, 1, -1, -1, 1, -1, 1, -1, 1}]
```

$$\left(\{-2 \to 2, 3\} \quad \{2 \to 3, 7\} \quad \{-1 \to 3, 4\} \right)$$

```
HighlightGraph[graph2, res2[[1]], GraphLayout ->
"SpringElectricalEmbedding"]
```



```
FCGraphFindPath[graph2, {1, 1, 1, 1, -1, -1, 1, -1, 1, -1, 1},
SameSideExternalEdges -> {-1, -2}]
```

$$\{\}$$

9.43 FCLoopBasisCreateScalarProducts

`FCLoopBasisCreateScalarProducts {q1, q2, ...}, {p1, p2, ...}, {d1, d2, ...}, head]` generates a list of all loop-momentum dependent scalar products made out of the loop momenta **`q1, q2, ...`** and external momenta **`p1, p2, ...`** in the space-time dimensions **`d1, d2, ...`**. The argument **`head`** can be **`Pair`** to generate Lorentzian scalar products or **`CartesianPair`** to generate Cartesian scalar products.

9.43.1 See also

Overview

9.43.2 Examples

```
FCLoopBasisCreateScalarProducts[{l}, {}, {D}, Pair]
```

$$\{l^2\}$$

```
FCLoopBasisCreateScalarProducts[{l}, {p1, p2}, {4}, Pair]
```

$$\left\{\bar{l}^2, \bar{l}\cdot\overline{\text{p1}}, \bar{l}\cdot\overline{\text{p2}}\right\}$$

```
FCLoopBasisCreateScalarProducts[{l}, {}, {D - 1}, CartesianPair]
```

$$\{l^2\}$$

9.44 FCLoopBasisFindCompletion

`FCLoopBasisFindCompletion[int, {q1, q2, ...}]` determines propagators that need to be included in the loop integral **`int`** (that depends on the loop momenta **`q1`**, **`q2`**, ...), to ensure that the propagators of **`int`** form a basis.

For integrals with propagators that do not form a basis, such a completion must be found prior to processing those integrals with tools that do Integration-By-Parts (IBP) reduction (e.g. FIRE, KIRA or LiteRed). Furthermore, **`int`** may not contain linearly dependent propagators.

The input can also consist of an **`FCTopology`** object or a list thereof.

9.44.1 See also

Overview, FCLoopBasisIncompleteQ.

9.44.2 Examples

```
FAD[q, {q - p + l, m}]

FCLoopBasisFindCompletion[%, {q}]
```

$$\frac{1}{q^2.((l-p+q)^2-m^2)}$$

$$\left\{\frac{1}{q^2.((l-p+q)^2-m^2)},\{l\cdot q\}\right\}$$

```
FAD[{q1, m1}, {q2, m2}]

FCLoopBasisFindCompletion[%, {q1, q2}]
```

$$\frac{1}{(\text{q1}^2-\text{m1}^2).(\text{q2}^2-\text{m2}^2)}$$

$$\left\{\frac{1}{(\text{q1}^2-\text{m1}^2).(\text{q2}^2-\text{m2}^2)},\{\text{q1}\cdot\ \text{q2}\}\right\}$$

```
FAD[q1 + p, q2 - k] SPD[q1, q2]

FCLoopBasisFindCompletion[%, {q1, q2}, Method -> {FAD[{q2 + k, m}], FAD[{q1
- p, m}], SPD[p, q2], SPD[k, q1]}]
```

$$\frac{\text{q1}\cdot\ \text{q2}}{(p+\text{q1})^2.(\text{q2}-k)^2}$$

$$\left\{\frac{\text{q1}\cdot\ \text{q2}}{(p+\text{q1})^2.(\text{q2}-k)^2},\left\{\frac{1}{(k+\text{q2})^2-m^2},\frac{1}{(\text{q1}-p)^2-m^2},p\cdot\ \text{q2},k\cdot\ \text{q1}\right\}\right\}$$

Cartesian integrals are also supported.

```
CFAD[q1, q2, {q1 - l1, m1}, {q2 - l2, m2}]

FCLoopBasisFindCompletion[%, {q1, q2}]
```

$$\frac{1}{(\text{q1}^2 - i\eta).(\text{q2}^2 - i\eta).((\text{q1} - \text{l1})^2 + \text{m1} - i\eta).((\text{q2} - \text{l2})^2 + \text{m2} - i\eta)}$$

$$\left\{\frac{1}{(\text{q1}^2 - i\eta).(\text{q2}^2 - i\eta).((\text{q1} - \text{l1})^2 + \text{m1} - i\eta).((\text{q2} - \text{l2})^2 + \text{m2} - i\eta)}, \{\text{l1} \cdot\ \text{q2}, \text{l2} \cdot\ \text{q1}, \text{q1} \cdot\ \text{q2}\}\right\}$$

Extending **FCTopology** objects

```
FCLoopBasisFindCompletion[FCTopology[topo, {FAD[p1], FAD[p2], FAD[p1 - q],
FAD[p2 - q]}, {p1, p2}, {q}, {}, {}]]
```

$$\text{FCTopology}\left(\text{topo}, \left\{\frac{1}{\text{p1}^2}, \frac{1}{\text{p2}^2}, \frac{1}{(\text{p1} - q)^2}, \frac{1}{(\text{p2} - q)^2}, \frac{1}{(\text{p1} \cdot\ \text{p2} + i\eta)}\right\}, \{\text{p1}, \text{p2}\}, \{q\}, \{\}, \{\}\right)$$

```
FCLoopBasisFindCompletion[{
   FCTopology[topo1, {FAD[p1], FAD[p2], FAD[p1 - q], FAD[p2 - q]}, {p1,
p2}, {q}, {}, {}],
   FCTopology[topo2, {FAD[p1], FAD[p2], FAD[p1 - q], FAD[p2 - p1]}, {p1,
p2}, {q}, {}, {}]
  }]
```

$$\left\{\text{FCTopology}\left(\text{topo1}, \left\{\frac{1}{\text{p1}^2}, \frac{1}{\text{p2}^2}, \frac{1}{(\text{p1} - q)^2}, \frac{1}{(\text{p2} - q)^2}, \frac{1}{(\text{p1} \cdot\ \text{p2} + i\eta)}\right\}, \{\text{p1}, \text{p2}\}, \{q\}, \{\}, \{\}\right),\right.$$
$$\left.\text{FCTopology}\left(\text{topo2}, \left\{\frac{1}{\text{p1}^2}, \frac{1}{\text{p2}^2}, \frac{1}{(\text{p1} - q)^2}, \frac{1}{(\text{p2} - \text{p1})^2}, \frac{1}{(\text{p2} \cdot q + i\eta)}\right\}, \{\text{p1}, \text{p2}\}, \{q\}, \{\}, \{\}\right)\right\}$$

The function pays attention to the $i\eta$ signs in the propagators

```
FCLoopBasisFindCompletion[{FCTopology[
    asyR1prop2Ltopo13011X11011NAux1, {SFAD[{{I*p1, 0}, {0, {-1}}, 1}],
        SFAD[{{(-I)*p3, 0}, {-mb^2, -1}, 1}],
        SFAD[{{I*(p1 + p3), 0}, {-mb^2, -1}, 1}],
        SFAD[{{I*(p1 - q), 0}, {-mb^2, -1}, 1}]}, {p1,
        p3}, {q}, {SPD[q, q] -> mb^2}, {}]}, FCE -> True]
```

$$\left\{\text{FCTopology}\left(\text{asyR1prop2Ltopo13011X11011NAux1},\right.\right.$$
$$\left\{\frac{1}{(-\text{p1}^2 - i\eta)}, \frac{1}{(-\text{p3}^2 + \text{mb}^2 - i\eta)}, \frac{1}{(-(\text{p1} + \text{p3})^2 + \text{mb}^2 - i\eta)},\right.$$
$$\left.\left.\left.\frac{1}{(-(\text{p1} - q)^2 + \text{mb}^2 - i\eta)}, \frac{1}{(\text{p3} \cdot q - i\eta)}\right\}, \{\text{p1}, \text{p3}\}, \{q\}, \{q^2 \to\ \text{mb}^2\}, \{\}\right)\right\}$$

9.45 FCLoopBasisGetSize

`FCLoopBasisGetSize[n1, n2]` returns the number of linearly independent propagators for a topology that contains **`n1`** loop momenta and **`n2`** external momenta.

9.45.1 See also

Overview

9.45.2 Examples

```
FCLoopBasisGetSize[1, 0]
```

1

```
FCLoopBasisGetSize[2, 1]
```

5

```
FCLoopBasisGetSize[3, 2]
```

12

```
FCLoopBasisGetSize[4, 1]
```

14

The third argument (if given) is simply added to the final result.

```
FCLoopBasisGetSize[4, 1, 1]
```

15

9.46 FCLoopBasisIncompleteQ

`FCLoopBasisIncompleteQ[int, {q1, q2, ...}]` checks whether the loop integral or topology **`int`** lacks propagators need to have a linearly independent basis .

The input can also consist of an **`FCTopology`** object or a list thereof.

9.46.1 See also

Overview, FCLoopBasisOverdeterminedQ.

9.46.2 Examples

```
FAD[{q1, m1}]

FCLoopBasisIncompleteQ[%, {q1}]
```

$$\frac{1}{\text{q1}^2 - \text{m1}^2}$$

False

```
SPD[q1, l] FAD[{q1, m1}, {q1 - l + p, m}]

FCLoopBasisIncompleteQ[%, {q1}]
```

$$\frac{l \cdot \text{q1}}{(\text{q1}^2 - \text{m1}^2) \,.\, ((-l + p + \text{q1})^2 - m^2)}$$

False

```
FAD[{q1, m1}, {q2, m2}]

FCLoopBasisIncompleteQ[%, {q1, q2}]
```

$$\frac{1}{(\text{q1}^2 - \text{m1}^2) \,.\, (\text{q2}^2 - \text{m2}^2)}$$

True

```
FAD[q1, q2, {q1 - l1, m1}, {q2 - l2, m2}]

FCLoopBasisIncompleteQ[%, {q1, q2}]
```

$$\frac{1}{\text{q1}^2.\text{q2}^2.\left((\text{q1}-\text{l1})^2-\text{m1}^2\right).\left((\text{q2}-\text{l2})^2-\text{m2}^2\right)}$$

True

```
CSPD[q1, l] CFAD[{q1, m1}, {q1 - l + p, m}]

FCLoopBasisIncompleteQ[%, {q1}]
```

$$\frac{l\cdot \text{q1}}{(\text{q1}^2+\text{m1}-i\eta).((-l+p+\text{q1})^2+m-i\eta)}$$

False

```
SFAD[{q1, m1}, {q2, m2}]

FCLoopBasisIncompleteQ[%, {q1, q2}]
```

$$\frac{1}{(\text{q1}^2-\text{m1}+i\eta).(\text{q2}^2-\text{m2}+i\eta)}$$

True

```
FCLoopBasisIncompleteQ[FCTopology[topo, {FAD[p1],
    FAD[p2], FAD[p1 - q], FAD[p2 - q]}, {p1, p2}, {q}, {}, {}]]
```

True

```
FCLoopBasisIncompleteQ[{
   FCTopology[topo1, {FAD[p1], FAD[p2], FAD[p1 - q], FAD[p2 - q]}, {p1,
p2}, {q}, {}, {}],
   FCTopology[topo2, {FAD[p1], FAD[p2], FAD[p1 - q], FAD[p2 - p1]}, {p1,
p2}, {q}, {}, {}],
   FCTopology[topo3, {FAD[p1], FAD[p1 - q]}, {p1}, {q}, {}, {}]
  }]
```

{True, True, False}

9.47 FCLoopBasisOverdeterminedQ

`FCLoopBasisOverdeterminedQ[int, {q1, q2, ...}]` checks whether the loop integral or topology **`int`** contains linearly dependent propagators.

The input can also consist of an **`FCTopology`** object or a list thereof.

9.47.1 See also

Overview, FCLoopBasisIncompleteQ.

9.47.2 Examples

```
FAD[{q1, m1}, {q1 - l + p, m}]

FCLoopBasisOverdeterminedQ[%, {q1}]
```

$$\frac{1}{(\text{q1}^2 - \text{m1}^2) \cdot ((-l + p + \text{q1})^2 - m^2)}$$

False

```
FAD[q1, {q1, m1}]

FCLoopBasisOverdeterminedQ[%, {q1}]
```

$$\frac{1}{\text{q1}^2 \cdot (\text{q1}^2 - \text{m1}^2)}$$

True

```
FAD[q1, q2, {q1 + l, m1}, {q1 - l, m1}, {q2 + l, m1}, {q2 - l, m1}]

FCLoopBasisOverdeterminedQ[%, {q1, q2}]
```

$$\frac{1}{\text{q1}^2 \cdot \text{q2}^2 \cdot ((l + \text{q1})^2 - \text{m1}^2) \cdot ((\text{q1} - l)^2 - \text{m1}^2) \cdot ((l + \text{q2})^2 - \text{m1}^2) \cdot ((\text{q2} - l)^2 - \text{m1}^2)}$$

True

```
FCLoopBasisOverdeterminedQ[FCTopology[topo1, {FAD[p1], FAD[p2],
    FAD[p1 - q], FAD[p2 - q], FAD[p1 - p2], FAD[p1 + p2 + q]}, {p1, p2},
{q}, {}, {}]]
```

True

```
FCLoopBasisOverdeterminedQ[{FCTopology[topo1, {FAD[p1], FAD[p2],
     FAD[p1 - q], FAD[p2 - q], FAD[p1 - p2], FAD[p1 + p2 + q]}, {p1, p2},
{q}, {}, {}],
   FCTopology[topo2, {FAD[p1], FAD[p2],
     FAD[p1 - q], FAD[p2 - q], FAD[p1 - p2]}, {p1, p2}, {q}, {}, {}]
  }]
```

{True, False}

9.48 FCLoopBasisSplit

`FCLoopBasisSplit[int, {q1, q2, ...}]` checks if the given loop integral factorizes and if so splits it into independent integrals.

9.48.1 See also

Overview

9.48.2 Examples

```
FCI@FAD[{q1, m}, {q2, m}, {p1 - p2, 0}]

FCLoopBasisSplit[%, {q1, q2}, Head -> loopInt]
```

$$\frac{1}{(\text{q1}^2 - m^2)\,.\,(\text{q2}^2 - m^2)\,.(\text{p1} - \text{p2})^2}$$

$$\left\{\text{loopInt}\left(\frac{1}{\text{q1}^2 - m^2}, \{\text{q1}\}\right), \text{loopInt}\left(\frac{1}{\text{q2}^2 - m^2}, \{\text{q2}\}\right), \text{loopInt}\left(\frac{1}{(\text{p1} - \text{p2})^2}, 0\right)\right\}$$

```
FCI[SFAD[q1, q1 - q2, q2, {q3, m^2}]]

FCLoopBasisSplit[%, {q1, q2, q3}, Head -> loop, FCE -> True]
```

$$\frac{1}{(\text{q1}^2 + i\eta).((\text{q1} - \text{q2})^2 + i\eta).(\text{q2}^2 + i\eta).(\text{q3}^2 - m^2 + i\eta)}$$

$$\left\{\text{loop}\left(\frac{1}{(\text{q3}^2 - m^2 + i\eta)}, \{\text{q3}\}\right), \text{loop}\left(\frac{1}{(\text{q1}^2 + i\eta).(\text{q2}^2 + i\eta).((\text{q1} - \text{q2})^2 + i\eta)}, \{\text{q1}, \text{q2}\}\right)\right\}$$

9.49 FCLoopBasisExtract

`FCLoopBasisExtract[int, {q1, q2, ...}]` is an auxiliary function that extracts the scalar products that form the basis of the loop integral in int. It needs to know the loop momenta on which the integral depends and the dimensions of the momenta that may occur in the integral.

9.49.1 See also

Overview

9.49.2 Examples

```
SPD[q, p] SFAD[q, q - p, q - p]

FCLoopBasisExtract[%, {q}, SetDimensions -> {4, D}]
```

$$\frac{p \cdot q}{(q^2 + i\eta).((q - p)^2 + i\eta)^2}$$

$$\left\{\{p \cdot q, q^2, -2(p \cdot q) + p^2 + q^2\}, \{p \cdot q, q^2\}, \{-1, 1, 2\}, \left\{p \cdot q, \frac{1}{(q^2 + i\eta)}, \frac{1}{((q - p)^2 + i\eta)}\right\}\right\}$$

```
SFAD[p1]

FCLoopBasisExtract[%, {p1, p2, p3}, FCTopology -> True, FCE -> True]
```

$$\frac{1}{(\text{p1}^2 + i\eta)}$$

$$\left\{\{\text{p1}^2\}, \{\text{p1}^2, \text{p1} \cdot \text{p2}, \text{p1} \cdot \text{p3}, \text{p2}^2, \text{p2} \cdot \text{p3}, \text{p3}^2\}, \{1\}, \left\{\frac{1}{(\text{p1}^2 + i\eta)}\right\}\right\}$$

9.50 FCLoopCanonicalize

`FCLoopCanonicalize[exp, {q1, q2, ...}, loopHead]` is an auxiliary internal function that canonicalizes indices of multi-loop integrals with loop momenta **`q1, q2, ...`** that are wrapped with **`loopHead`**. The output is given as a list of 4 entries, of which the last one contains a list of all the unique loop integrals in the given expression. After those are simplified, the original output of **`FCLoopCanonicalize`** together with the list of the simplified unique integrals should be inserted into **`FCLoopSolutionList`** to obtain the final replacement list that will be applied to the original expression.

9.50.1 See also

Overview, FCLoopSolutionList.

9.50.2 Examples

```
FCLoopCanonicalize[myHead[FVD[q, \[Mu]]], q, myHead]
```

$$\begin{pmatrix} \text{myHead}\left(q^{\mu}\right) \\ \{\text{FCGV(cli191)} \to \mu\} \\ \text{myHead}\left(q^{\text{FCGV(cli191)}}\right) \\ \text{myHead}\left(q^{\text{FCGV(cli191)}}\right) \end{pmatrix}$$

```
FCLoopCanonicalize[myHead[FVD[q, \[Mu]] FVD[q, \[Nu]] FAD[q, {q + p, m}]] +
myHead[FVD[q, \[Rho]] FVD[q, \[Sigma]] FAD[q, {q + p, m}]], q, myHead]
```

$$\begin{aligned} &\left\{\left\{\text{myHead}\left(\frac{q^{\mu}q^{\nu}}{q^2.\left((p+q)^2-m^2\right)}\right), \text{myHead}\left(\frac{q^{\rho}q^{\sigma}}{q^2.\left((p+q)^2-m^2\right)}\right)\right\},\right. \\ &\{\{\text{FCGV(cli201)} \to \mu, \text{FCGV(cli202)} \to \nu\}, \{\text{FCGV(cli201)} \to \rho, \\ &\left.\text{FCGV(cli202)} \to \sigma\}\}, \left\{\text{myHead}\left(\frac{q^{\text{FCGV(cli201)}}q^{\text{FCGV(cli202)}}}{q^2.\left((p+q)^2-m^2\right)}\right),\right. \\ &\left.\text{myHead}\left(\frac{q^{\text{FCGV(cli201)}}q^{\text{FCGV(cli202)}}}{q^2.\left((p+q)^2-m^2\right)}\right)\right\}, \left\{\text{myHead}\left(\frac{q^{\text{FCGV(cli201)}}q^{\text{FCGV(cli202)}}}{q^2.\left((p+q)^2-m^2\right)}\right)\right\}\Bigg\} \end{aligned}$$

```
FCLoopCanonicalize[myHead[FVD[q1, \[Mu]] FVD[q2, \[Nu]]], {q1, q2}, myHead]
```

$$\begin{pmatrix} \text{myHead}\left(\text{q1}^{\mu}\ \text{q2}^{\nu}\right) \\ \{\text{FCGV(cli211)} \to \mu, \text{FCGV(cli212)} \to \nu\} \\ \text{myHead}\left(\text{q1}^{\text{FCGV(cli211)}}\ \text{q2}^{\text{FCGV(cli212)}}\right) \\ \text{myHead}\left(\text{q1}^{\text{FCGV(cli211)}}\ \text{q2}^{\text{FCGV(cli212)}}\right) \end{pmatrix}$$

9.51 FCLoopCreateRulesToGLI

FCLoopCreateRulesToGLI[topo] creates replacement rules for converting numerators from the given topology to GLI objects with inverse propagators.

It is also possible to use **FCLoopCreateRulesToGLI[{topo1, topo2, ...}]**.

9.51.1 See also

Overview, FCTopology, GLI, FCLoopFindTopologies, FCLoopFindTopologyMappings.

9.51.2 Examples

1-loop tadpole

```
FCLoopCreateRulesToGLI[FCTopology[topo1, {SFAD[{p1, m^2}]}, {p1}, {}, {},
{}]]
```

$$\left\{\text{p1}^2 \to G^{\text{topo1}}(-1) + m^2\right\}$$

2-loop tadpole with 3 different masses

```
FCLoopCreateRulesToGLI[FCTopology[topo1, {SFAD[{p1, m1^2}], SFAD[{p2,
m2^2}],
    SFAD[{p1 - p2, m3^2}]}, {p1, p2}, {}, {}, {}]]
```

$$\left\{\text{p1}^2 \to G^{\text{topo1}}(-1,0,0) + \text{m1}^2, \text{p2}^2 \to G^{\text{topo1}}(0,-1,0) + \text{m2}^2, \text{p1}\cdot\ \text{p2} \to \frac{1}{2}\left(G^{\text{topo1}}(-1,\right.\right.$$
$$\left.\left.0,0) + G^{\text{topo1}}(0,-1,0) - G^{\text{topo1}}(0,0,-1) + \text{m1}^2 + \text{m2}^2 - \text{m3}^2\right)\right\}$$

2-loop self-energy

```
FCLoopCreateRulesToGLI[FCTopology["prop2Lv1",
   {SFAD[{p1, m1^2}], SFAD[{p2, m2^2}], SFAD[p1 - q], SFAD[p2 - q],
    SFAD[{p1 - p2, m3^2}]}, {p1, p2}, {Q}, {}, {}]]
```

$$\Big\{\mathrm{p1}^2 \to G^{\mathrm{prop2Lv1}}(-1,0,0,0,0) + \mathrm{m1}^2, \mathrm{p2}^2 \to G^{\mathrm{prop2Lv1}}(0,-1,0,0,0) + \mathrm{m2}^2,$$
$$\mathrm{p1}\cdot\ \mathrm{p2} \to \frac{1}{2}\left(G^{\mathrm{prop2Lv1}}(-1,0,0,0,0) + G^{\mathrm{prop2Lv1}}(0,-1,0,0,0) - G^{\mathrm{prop2Lv1}}(0,0,0,0,\right.$$
$$\left.-1) + \mathrm{m1}^2 + \mathrm{m2}^2 - \mathrm{m3}^2\right), \mathrm{p1}\cdot q \to \frac{1}{2}\left(G^{\mathrm{prop2Lv1}}(-1,0,0,0,0) - G^{\mathrm{prop2Lv1}}(0,0,-1,0,\right.$$
$$\left.0) + \mathrm{m1}^2 + q^2\right), \mathrm{p2}\cdot q \to \frac{1}{2}\left(G^{\mathrm{prop2Lv1}}(0,-1,0,0,0) - G^{\mathrm{prop2Lv1}}(0,0,0,-1,0) + \mathrm{m2}^2 + q^2\right)\Big\}$$

A list of 3-loop self-energy topologies

```
topoList = {
   FCTopology["prop3Lv1", {SFAD[p1], SFAD[p2], SFAD[p3], SFAD[p1 - p2],
SFAD[p2 - p3], SFAD[p1 + q1],
        SFAD[p2 + q1], SFAD[p3 + q1], SFAD[{{0, p1 . p3}}]}, {p1, p2, p3},
{q1}, {}, {}],

   FCTopology["prop3L2", {SFAD[p1], SFAD[p2], SFAD[p3], SFAD[p1 - p2],
SFAD[p2 - p3], SFAD[p1 - p3],
        SFAD[p1 + q1], SFAD[p3 + q1], SFAD[{{0, (p1 - p2) . q1}}]}, {p1,
p2, p3}, {q1}, {}, {}],

   FCTopology["prop3L3", {SFAD[p1], SFAD[p1 - p4], SFAD[p3], SFAD[p4],
SFAD[p1 - p3 - p4], SFAD[p1 + q1],
        SFAD[p3 + p4 + q1], SFAD[p3 + q1], SFAD[{{0, (p4) . q1}}]}, {p1,
p3, p4}, {q1}, {}, {}]
  }
```

$$\Big\{\mathrm{FCTopology}\left(\mathrm{prop3Lv1}, \left\{\frac{1}{(\mathrm{p1}^2+i\eta)}, \frac{1}{(\mathrm{p2}^2+i\eta)}, \frac{1}{(\mathrm{p3}^2+i\eta)}, \frac{1}{((\mathrm{p1}-\mathrm{p2})^2+i\eta)},\right.\right.$$
$$\left.\frac{1}{((\mathrm{p2}-\mathrm{p3})^2+i\eta)}, \frac{1}{((\mathrm{p1}+\mathrm{q1})^2+i\eta)}, \frac{1}{((\mathrm{p2}+\mathrm{q1})^2+i\eta)}, \frac{1}{((\mathrm{p3}+\mathrm{q1})^2+i\eta)}, \frac{1}{(\mathrm{p1}\cdot\ \mathrm{p3}+i\eta)}\right\},$$
$$\{\mathrm{p1},\mathrm{p2},\mathrm{p3}\},\{\mathrm{q1}\},\{\},\{\}\Big), \mathrm{FCTopology}\left(\mathrm{prop3L2}, \left\{\frac{1}{(\mathrm{p1}^2+i\eta)}, \frac{1}{(\mathrm{p2}^2+i\eta)}, \frac{1}{(\mathrm{p3}^2+i\eta)},\right.\right.$$
$$\frac{1}{((\mathrm{p1}-\mathrm{p2})^2+i\eta)}, \frac{1}{((\mathrm{p2}-\mathrm{p3})^2+i\eta)}, \frac{1}{((\mathrm{p1}-\mathrm{p3})^2+i\eta)}, \frac{1}{((\mathrm{p1}+\mathrm{q1})^2+i\eta)}, \frac{1}{((\mathrm{p3}+\mathrm{q1})^2+i\eta)},$$
$$\left.\frac{1}{((\mathrm{p1}-\mathrm{p2})\cdot\ \mathrm{q1}+i\eta)}\right\}, \{\mathrm{p1},\mathrm{p2},\mathrm{p3}\},\{\mathrm{q1}\},\{\},\{\}\Big), \mathrm{FCTopology}\left(\mathrm{prop3L3}, \left\{\frac{1}{(\mathrm{p1}^2+i\eta)},\right.\right.$$
$$\frac{1}{((\mathrm{p1}-\mathrm{p4})^2+i\eta)}, \frac{1}{(\mathrm{p3}^2+i\eta)}, \frac{1}{(\mathrm{p4}^2+i\eta)}, \frac{1}{((\mathrm{p1}-\mathrm{p3}-\mathrm{p4})^2+i\eta)}, \frac{1}{((\mathrm{p1}+\mathrm{q1})^2+i\eta)},$$
$$\left.\left.\frac{1}{((\mathrm{p3}+\mathrm{p4}+\mathrm{q1})^2+i\eta)}, \frac{1}{((\mathrm{p3}+\mathrm{q1})^2+i\eta)}, \frac{1}{(\mathrm{p4}\cdot\ \mathrm{q1}+i\eta)}\right\}, \{\mathrm{p1},\mathrm{p3},\mathrm{p4}\},\{\mathrm{q1}\},\{\},\{\}\Big)\Big\}$$

```
FCLoopCreateRulesToGLI[topoList]
```

$$\left\{\left\{\mathrm{p1}^2 \to G^{\mathrm{prop3Lv1}}(-1,0,0,0,0,0,0,0,0,0),\mathrm{p2}^2 \to G^{\mathrm{prop3Lv1}}(0,\right.\right.$$
$$-1,0,0,0,0,0,0,0,0),\mathrm{p3}^2 \to G^{\mathrm{prop3Lv1}}(0,0,-1,0,0,0,0,0,0),$$
$$\mathrm{p1}\cdot\ \mathrm{p2} \to \frac{1}{2}\left(G^{\mathrm{prop3Lv1}}(-1,0,0,0,0,0,0,0,0,0)+G^{\mathrm{prop3Lv1}}(0,-1,0,0,\right.$$
$$0,0,0,0,0)-G^{\mathrm{prop3Lv1}}(0,0,0,-1,0,0,0,0,0)\Big),\mathrm{p1}\cdot\ \mathrm{p3} \to G^{\mathrm{prop3Lv1}}(0,$$
$$0,0,0,0,0,0,0,0,-1),\mathrm{p2}\cdot\ \mathrm{p3} \to \frac{1}{2}\left(G^{\mathrm{prop3Lv1}}(0,-1,0,0,0,0,0,0,\right.$$
$$0)+G^{\mathrm{prop3Lv1}}(0,0,-1,0,0,0,0,0,0)-G^{\mathrm{prop3Lv1}}(0,0,0,0,-1,0,0,0,$$
$$0)\Big),\mathrm{p1}\cdot\ \mathrm{q1} \to \frac{1}{2}\left(-G^{\mathrm{prop3Lv1}}(-1,0,0,0,0,0,0,0,0,0)+G^{\mathrm{prop3Lv1}}(0,\right.$$
$$0,0,0,0,-1,0,0,0)-\mathrm{q1}^2\Big),\mathrm{p2}\cdot\ \mathrm{q1} \to \frac{1}{2}\left(-G^{\mathrm{prop3Lv1}}(0,-1,\right.$$
$$0,0,0,0,0,0,0,0)+G^{\mathrm{prop3Lv1}}(0,0,0,0,0,0,0,-1,0,0)-\mathrm{q1}^2\Big),$$
$$\mathrm{p3}\cdot\ \mathrm{q1} \to \frac{1}{2}\left(-G^{\mathrm{prop3Lv1}}(0,0,-1,0,0,0,0,0,0,0)+G^{\mathrm{prop3Lv1}}(0,0,\right.$$
$$\left.\left.0,0,0,0,0,0,-1,0)-\mathrm{q1}^2\Big)\right\},\left\{\mathrm{p1}^2 \to G^{\mathrm{prop3L2}}(-1,0,0,0,0,0,0,0,\right.$$
$$0,0),\mathrm{p2}^2 \to G^{\mathrm{prop3L2}}(0,-1,0,0,0,0,0,0,0,0),\mathrm{p3}^2 \to G^{\mathrm{prop3L2}}(0,$$
$$0,-1,0,0,0,0,0,0,0),\mathrm{p1}\cdot\ \mathrm{p2} \to \frac{1}{2}\left(G^{\mathrm{prop3L2}}(-1,0,0,0,0,0,\right.$$
$$0,0,0)+G^{\mathrm{prop3L2}}(0,-1,0,0,0,0,0,0,0,0)-G^{\mathrm{prop3L2}}(0,0,0,$$
$$-1,0,0,0,0,0)\Big),\mathrm{p1}\cdot\ \mathrm{p3} \to \frac{1}{2}\left(G^{\mathrm{prop3L2}}(-1,0,0,0,0,0,0,0,\right.$$
$$0,0)+G^{\mathrm{prop3L2}}(0,0,-1,0,0,0,0,0,0,0)-G^{\mathrm{prop3L2}}(0,0,0,0,$$
$$0,-1,0,0,0)\Big),\mathrm{p2}\cdot\ \mathrm{p3} \to \frac{1}{2}\left(G^{\mathrm{prop3L2}}(0,-1,0,0,0,0,0,0,0,$$
$$0)+G^{\mathrm{prop3L2}}(0,0,-1,0,0,0,0,0,0,0)-G^{\mathrm{prop3L2}}(0,0,0,0,-1,0,0,0,$$
$$0)\Big),\mathrm{p1}\cdot\ \mathrm{q1} \to \frac{1}{2}\left(-G^{\mathrm{prop3L2}}(-1,0,0,0,0,0,0,0,0,0)+G^{\mathrm{prop3L2}}(0,0,\right.$$
$$0,0,0,0,-1,0,0)-\mathrm{q1}^2\Big),\mathrm{p2}\cdot\ \mathrm{q1} \to \frac{1}{2}\left(-G^{\mathrm{prop3L2}}(-1,0,0,0,0,$$
$$0,0,0,0)+G^{\mathrm{prop3L2}}(0,0,0,0,0,0,-1,0,0)-2G^{\mathrm{prop3L2}}(0,0,0,0,0,$$
$$0,0,0,-1)-\mathrm{q1}^2\Big),\mathrm{p3}\cdot\ \mathrm{q1} \to \frac{1}{2}\left(-G^{\mathrm{prop3L2}}(0,0,-1,0,0,0,0,0,\right.$$
$$0)+G^{\mathrm{prop3L2}}(0,0,0,0,0,0,0,0,-1,0)-\mathrm{q1}^2\Big)\Big\},\Big\{\mathrm{p1}^2 \to G^{\mathrm{prop3L3}}(-1,$$
$$0,0,0,0,0,0,0,0,0),\mathrm{p3}^2 \to G^{\mathrm{prop3L3}}(0,0,-1,0,0,0,0,0,0,0),$$
$$\mathrm{p4}^2 \to G^{\mathrm{prop3L3}}(0,0,0,-1,0,0,0,0,0),\mathrm{p1}\cdot\ \mathrm{p3} \to \frac{1}{2}\left(G^{\mathrm{prop3L3}}(0,-1,\right.$$
$$0,0,0,0,0,0,0)+G^{\mathrm{prop3L3}}(0,0,-1,0,0,0,0,0,0,0)-G^{\mathrm{prop3L3}}(0,0,0,$$
$$-1,0,0,0,0,0)-G^{\mathrm{prop3L3}}(0,0,0,0,-1,0,0,0,0)+G^{\mathrm{prop3L3}}(0,0,$$
$$0,0,0,0,-1,0,0)-G^{\mathrm{prop3L3}}(0,0,0,0,0,0,0,-1,0)-2G^{\mathrm{prop3L3}}(0,$$
$$0,0,0,0,0,0,0,0,-1)\Big),\mathrm{p1}\cdot\ \mathrm{p4} \to \frac{1}{2}\left(G^{\mathrm{prop3L3}}(-1,0,0,0,0,0,0,0,\right.$$
$$0,0)-G^{\mathrm{prop3L3}}(0,-1,0,0,0,0,0,0,0,0)+G^{\mathrm{prop3L3}}(0,0,0,-1,$$
$$0,0,0,0,0)\Big),\mathrm{p3}\cdot\ \mathrm{p4} \to \frac{1}{2}\left(-G^{\mathrm{prop3L3}}(0,0,0,-1,0,0,0,0,\right.$$
$$0)+G^{\mathrm{prop3L3}}(0,0,0,0,0,0,0,-1,0,0)-G^{\mathrm{prop3L3}}(0,0,0,0,0,0,0,0,-1,$$
$$0)-2G^{\mathrm{prop3L3}}(0,0,0,0,0,0,0,0,0,-1)\Big),\mathrm{p1}\cdot\ \mathrm{q1} \to \frac{1}{2}\left(-G^{\mathrm{prop3L3}}(-1,\right.$$
$$0,0,0,0,0,0,0,0,0)+G^{\mathrm{prop3L3}}(0,0,0,0,0,0,-1,0,0,0)-\mathrm{q1}^2\Big),$$
$$\mathrm{p3}\cdot\ \mathrm{q1} \to \frac{1}{2}\left(-G^{\mathrm{prop3L3}}(0,0,-1,0,0,0,0,0,0,0)+G^{\mathrm{prop3L3}}(0,0,0,0,0,\right.$$
$$\left.\left.0,0,-1,0)-\mathrm{q1}^2\Big),\mathrm{p4}\cdot\ \mathrm{q1} \to G^{\mathrm{prop3L3}}(0,0,0,0,0,0,0,0,0,-1)\right\}\right\}$$

9.52 FCLoopEikonalPropagatorFreeQ

FCLoopEikonalPropagatorFreeQ[exp] checks if the integral is free of eikonal propagators $\frac{1}{p\cdot q+x}$. If the option **First** is set to **False**, propagators that have both a quadratic and linear piece, e.g. $\frac{1}{p^2+p\cdot q+x}$ will also count as eikonal propagators. The option **Momentum** can be used to check for the presence of eikonal propagators only with respect to particular momenta. The check is performed only for **StandardPropagatorDenominator** and **CartesianPropagatorDenominator**.

9.52.1 See also

Overview

9.52.2 Examples

```
FCI@SFAD[p, p - q]

FCLoopEikonalPropagatorFreeQ[%]
```

$$\frac{1}{(p^2+i\eta).((p-q)^2+i\eta)}$$

True

```
FCI@SFAD[{{0, p . q}}]

FCLoopEikonalPropagatorFreeQ[%]
```

$$\frac{1}{(p\cdot q+i\eta)}$$

False

```
FCI@CFAD[{{0, p . q}}]

FCLoopEikonalPropagatorFreeQ[%, Momentum -> {q}]
```

$$\frac{1}{(p\cdot q-i\eta)}$$

False

```
FCI@SFAD[{{q, q . p}}]

FCLoopEikonalPropagatorFreeQ[%, First -> False]
```

$$\frac{1}{(q^2 + p \cdot q + i\eta)}$$

False

9.53 FCLoopExtract

FCLoopExtract[expr, {q1, q2, ...}, loopHead] exctracts loop integrals that depend on **q1, q2, ...** from the given expression. The output is given as a list of three entries. The first one contains part of the original expression that consists of irrelevant loop integrals and terms that are free of any loop integrals. The second entry contains relevant loop integrals, where each integral is wrapped into **loopHead**. The third entry is a list of all the unique loop integrals from the second entry and can be used as an input to another function. Note that if loop integrals contain free indices, those will not be canonicalized.

9.53.1 See also

Overview, FCLoopIsolate.

9.53.2 Examples

```
FCI[GSD[q - p1] . (GSD[q - p2] + M) . GSD[p3] SPD[q, p2] FAD[q, q - p1, {q
- p2, m}]]

FCLoopExtract[%, {q}, loopInt]
```

$$\frac{(\text{p2} \cdot q)(\gamma \cdot (q - \text{p1})).(M + \gamma \cdot (q - \text{p2})).(\gamma \cdot \text{p3})}{q^2.(q - \text{p1})^2.((q - \text{p2})^2 - m^2)}$$

$$\left\{0, ((\gamma\cdot\ \text{p1}).(\gamma\cdot\ \text{p2}).(\gamma\cdot\ \text{p3}) - M(\gamma\cdot\ \text{p1}).(\gamma\cdot\ \text{p3}))\ \text{loopInt}\left(\frac{\text{p2}\cdot q}{q^2.(q-\text{p1})^2.((q-\text{p2})^2-m^2)}\right)\right.$$
$$+ M\ \text{loopInt}\left(\frac{(\text{p2}\cdot q)(\gamma\cdot q).(\gamma\cdot\ \text{p3})}{q^2.(q-\text{p1})^2.((q-\text{p2})^2-m^2)}\right) - \text{loopInt}\left(\frac{(\text{p2}\cdot q)(\gamma\cdot\ \text{p1}).(\gamma\cdot q).(\gamma\cdot\ \text{p3})}{q^2.(q-\text{p1})^2.((q-\text{p2})^2-m^2)}\right)$$
$$- \text{loopInt}\left(\frac{(\text{p2}\cdot q)(\gamma\cdot q).(\gamma\cdot\ \text{p2}).(\gamma\cdot\ \text{p3})}{q^2.(q-\text{p1})^2.((q-\text{p2})^2-m^2)}\right) + \text{loopInt}\left(\frac{(\text{p2}\cdot q)(\gamma\cdot q).(\gamma\cdot q).(\gamma\cdot\ \text{p3})}{q^2.(q-\text{p1})^2.((q-\text{p2})^2-m^2)}\right),$$
$$\left\{\text{loopInt}\left(\frac{\text{p2}\cdot q}{q^2.(q-\text{p1})^2.((q-\text{p2})^2-m^2)}\right), \text{loopInt}\left(\frac{(\text{p2}\cdot q)(\gamma\cdot q).(\gamma\cdot\ \text{p3})}{q^2.(q-\text{p1})^2.((q-\text{p2})^2-m^2)}\right),\right.$$
$$\text{loopInt}\left(\frac{(\text{p2}\cdot q)(\gamma\cdot\ \text{p1}).(\gamma\cdot q).(\gamma\cdot\ \text{p3})}{q^2.(q-\text{p1})^2.((q-\text{p2})^2-m^2)}\right), \text{loopInt}\left(\frac{(\text{p2}\cdot q)(\gamma\cdot q).(\gamma\cdot\ \text{p2}).(\gamma\cdot\ \text{p3})}{q^2.(q-\text{p1})^2.((q-\text{p2})^2-m^2)}\right),$$
$$\left.\left.\text{loopInt}\left(\frac{(\text{p2}\cdot q)(\gamma\cdot q).(\gamma\cdot q).(\gamma\cdot\ \text{p3})}{q^2.(q-\text{p1})^2.((q-\text{p2})^2-m^2)}\right)\right\}\right\}$$

`FCLoopFromGLI[exp, topologies]` replaces **`GLI`**s in **`exp`** with the corresponding loop integrals in the **`FeynAmpDenominator`** notation according to the information provided in topologies.

9.53.3 See also

Overview, FCTopology, GLI, FCLoopValidTopologyQ.

9.53.4 Examples

```
topos = {
   FCTopology["topoBox1L", {FAD[{q, m0}], FAD[{q + p1, m1}], FAD[{q + p2,
m2}], FAD[{q + p2, m3}]},
    {q}, {p1, p2, p3}, {}, {}],
   FCTopology["topoTad2L", {FAD[{q1, m1}], FAD[{q2, m2}], FAD[{q1 - q2,
0}]}, {q1, q2}, {}, {}, {}]}
```

$$\left\{\text{FCTopology}\left(\text{topoBox1L}, \left\{\frac{1}{q^2-\text{m0}^2}, \frac{1}{(\text{p1}+q)^2-\text{m1}^2}, \frac{1}{(\text{p2}+q)^2-\text{m2}^2},\right.\right.\right.$$
$$\left.\left.\frac{1}{(\text{p2}+q)^2-\text{m3}^2}\right\}, \{q\}, \{\text{p1}, \text{p2}, \text{p3}\}, \{\}, \{\}\right), \text{FCTopology}\left(\text{topoTad2L},$$
$$\left.\left\{\frac{1}{\text{q1}^2-\text{m1}^2}, \frac{1}{\text{q2}^2-\text{m2}^2}, \frac{1}{(\text{q1}-\text{q2})^2}\right\}, \{\text{q1}, \text{q2}\}, \{\}, \{\}, \{\}\right)\right\}$$

```
exp = a1 GLI["topoBox1L", {1, 1, 1, 1}] + a2 GLI["topoTad2L", {1, 2, 2}]
```

$$\text{a1}\,G^{\text{topoBox1L}}(1,1,1,1) + \text{a2}\,G^{\text{topoTad2L}}(1,2,2)$$

```
FCLoopFromGLI[exp, topos]
```

$$\frac{\text{a1}}{\left(q^2-\text{m0}^2\right)\left((\text{p1}+q)^2-\text{m1}^2\right)\left((\text{p2}+q)^2-\text{m2}^2\right)\left((\text{p2}+q)^2-\text{m3}^2\right)}+\frac{\text{a2}}{\left(\text{q1}^2-\text{m1}^2\right)\left(\text{q2}^2-\text{m2}^2\right)^2(\text{q1}-\text{q2})^4}$$

Notice that it is necessary to specify all topologies present in **exp**. The function will not accept **GLI**s defined for unknown topologies

```
FCLoopFromGLI[GLI["topoXYZ", {1, 1, 1, 1, 1}], topos]
```

FCLoopSelectTopology: Error! FCLoopSelectTopology has encountered a fatal problem and must abort the computation. The problem reads: There are no topologies with the id topoXYZ

$Aborted

FCLoopFromGLI can also handle products of **GLI**s (currently only for standalone integrals or lists of integrals but not for amplitudes). In this case it will automatically introduce dummy names for the loop momenta.

```
FCLoopFromGLI[GLI["topoBox1L", {1, 0, 1, 0}] GLI["topoBox1L", {0, 1, 0,
1}], topos]
```

$$\frac{1}{\left(\text{FCGV}(\text{lmom21})^2-\text{m0}^2\right)\left((\text{p1}+\text{FCGV}(\text{lmom11}))^2-\text{m1}^2\right)\left((\text{p2}+\text{FCGV}(\text{lmom11}))^2-\text{m3}^2\right)\left((\text{p2}+\text{FCGV}(\text{lmom2}\right.$$

You can customize the naming scheme for the momenta via the **LoopMomentum** option. The first argument gives the number of the loop integral, while the second corresponds to a particular loop momentum this integral depends on.

```
SelectNotFree[Options[FCLoopFromGLI], LoopMomenta]
```

{LoopMomenta → ({FeynCalc`FCLoopFromGLI`Private`x, FeynCalc`FCLoopFromGLI`Private`y} → FCGV(lmom <> ToString[FeynCalc`FCLoopFromGLI`Private`x] <> ToString[F

```
FCLoopFromGLI[GLI["topoBox1L", {1, 0, 1, 0}] GLI["topoBox1L", {0, 1, 0,
1}], topos,
  LoopMomenta -> Function[{x, y}, "p" <> ToString[x] <> ToString[x]]]
```

$$\frac{1}{\left(\text{p22}^2-\text{m0}^2\right)\left((\text{p11}+\text{p1})^2-\text{m1}^2\right)\left((\text{p22}+\text{p2})^2-\text{m2}^2\right)\left((\text{p11}+\text{p2})^2-\text{m3}^2\right)}$$

9.54 FCLoopGetEtaSigns

FCLoopGetEtaSigns[exp] is an auxiliary function that extracts the signs of $i\eta$ from propagators present in the input expression. The result is returned as a list, e.g. **{}**, **{1}**, **{-1}** or **{-1, 1}**.

This is useful if one wants ensure that all propagators of the given integral or topology follow a particular $i\eta$ sign convention.

9.54.1 See also

Overview, FCTopology, FCLoopSwitchEtaSign.

9.54.2 Examples

```
FAD[{p, m}]

FCLoopGetEtaSigns[%]
```

$$\frac{1}{p^2 - m^2}$$

$$\{1\}$$

```
SFAD[{p, m^2}]

FCLoopGetEtaSigns[%]
```

$$\frac{1}{(p^2 - m^2 + i\eta)}$$

$$\{1\}$$

```
SFAD[{I p, -m^2}, EtaSign -> -1]

FCLoopGetEtaSigns[%]
```

$$\frac{1}{(-p^2 + m^2 - i\eta)}$$

$$\{-1\}$$

```
CFAD[{p, m^2}]

FCLoopGetEtaSigns[%]
```

$$\frac{1}{(p^2+m^2-i\eta)}$$

$$\{-1\}$$

9.55 FCLoopGLIDifferentiate

`FCLoopGLIDifferentiate[exp, topos, inv]` calculates the partial derivative of GLIs present in **`exp`** with respect to the scalar quantity **`inv`**. Here **`inv`** can be a constant (e.g. mass), a scalar product of some momenta or a 4-vector.

The list topos must contain the topologies describing all of the occurring GLIs.

To calculate multiple derivatives, use the notation **`FCLoopGLIDifferentiate[exp , topos, {inv, n}]`** for scalars and **`FCLoopGLIDifferentiate[exp , topos, {vec1, vec2, ...}]`** for vectors.

9.55.1 See also

Overview, FCTopology, GLI, FCLoopGLIExpand.

9.55.2 Examples

```
FCLoopGLIDifferentiate[x GLI[tad2l, {1, 1, 1}],
  {FCTopology[tad2l, {FAD[{p1, m1}], FAD[{p2, m2}], FAD[{p1 - p2, m3}]},
  {p1, p2}, {}, {}, {}]}, m1]
```

$$2\ \mathrm{m1}xG^{\mathrm{tad2l}}(2,1,1)$$

```
FCLoopGLIDifferentiate[x GLI[tad2l, {1, 1, 1}],
  {FCTopology[tad2l, {FAD[{p1, m1}], FAD[{p2, m2}], FAD[{p1 - p2, m3}]},
  {p1, p2}, {}, {}, {}]}, {m1, 5}]
```

$$3840\ \mathrm{m1}^5xG^{\mathrm{tad2l}}(6,1,1)+3840\ \mathrm{m1}^3xG^{\mathrm{tad2l}}(5,1,1)+720\ \mathrm{m1}xG^{\mathrm{tad2l}}(4,1,1)$$

```
FCLoopGLIDifferentiate[m2^2 GLI[tad2l, {1, 1, 1}],
  {FCTopology[tad2l, {FAD[{p1, m1}], FAD[{p2, m2}], FAD[{p1 - p2, m3}]},
  {p1, p2}, {}, {}, {}]}, m2]
```

$$2\,\text{m2}^3 G^{\text{tad2l}}(1,2,1) + 2\,\text{m2} G^{\text{tad2l}}(1,1,1)$$

```
FCLoopGLIDifferentiate[ GLI[prop1l, {1, 1}] + SPD[q] GLI[prop1l, {1, 0}],
  {FCTopology[prop1l, {FAD[{p1, m1}], FAD[{p1 + q, m2}]}, {p1}, {q}, {},
  {}]}, SPD[q]]
```

$$G^{\text{prop1l}}(1,0) - G^{\text{prop1l}}(1,2)$$

```
FCLoopGLIDifferentiate[SPD[p1, p2] GLI[topo1, {1, 1, 1}],
  {FCTopology[topo1, {SFAD[{p1, m1^2}], SFAD[{p2, m2^2}], SFAD[p1 - p2]},
  {p1, p2}, {}, {}, {}]},
  FVD[p1, mu], FCE -> True]
```

$$-2G^{\text{topo1}}(1,1,2)\,(\text{p1}^{\text{mu}} - \text{p2}^{\text{mu}})\,(\text{p1}\cdot\,\text{p2}) - 2\,\text{p1}^{\text{mu}} G^{\text{topo1}}(2,1,1)(\text{p1}\cdot\,\text{p2}) + \text{p2}^{\text{mu}} G^{\text{topo1}}(1,1,1)$$

```
FCLoopGLIDifferentiate[SPD[p1, p2] GLI[topo1, {1, 1, 1}],
{FCTopology[topo1,
    {SFAD[{p1, m1^2}], SFAD[{p2, m2^2}], SFAD[p1 - p2]}, {p1, p2}, {}, {},
    {}]}
 , {FVD[p1, mu], FVD[p2, nu]}, FCE -> True]
```

$$-2G^{\text{topo1}}(1,1,2)\,(-g^{\text{mu nu}}(\text{p1}\cdot\,\text{p2}) - 2\,\text{p2}^{\text{mu}}\,\text{p1}^{\text{nu}} + \text{p1}^{\text{mu}}\,\text{p1}^{\text{nu}} + \text{p2}^{\text{mu}}\,\text{p2}^{\text{nu}}) + G^{\text{topo1}}(1,1,1)g^{\text{mu nu}} - 8G^{\text{topo1}}(1,1,3)\,(\text{p1}^{\text{mu}} - \text{p2}^{\text{mu}})\,(\text{p1}^{\text{nu}} - \text{p2}^{\text{nu}})\,(\text{p1}\cdot\,\text{p2}) + 4\,\text{p2}^{\text{nu}} G^{\text{topo1}}(1,2,2)\,(\text{p1}^{\text{mu}} - \text{p2}^{\text{mu}})\,(\text{p1}\cdot\,\text{p2}) - 4\,\text{p1}^{\text{mu}} G^{\text{topo1}}(2,1,2)\,(\text{p1}^{\text{nu}} - \text{p2}^{\text{nu}})\,(\text{p1}\cdot\,\text{p2}) + 4\,\text{p1}^{\text{mu}}\,\text{p2}^{\text{nu}} G^{\text{topo1}}(2,2,1)(\text{p1}\cdot\,\text{p2}) - 2\,\text{p1}^{\text{mu}}\,\text{p1}^{\text{nu}} G^{\text{topo1}}(2,1,1) - 2\,\text{p2}^{\text{mu}}\,\text{p2}^{\text{nu}} G^{\text{topo1}}(1,2,1)$$

9.56 FCLoopAddScalingParameter

`FCLoopAddScalingParameter[topo, la, rules]` multiplies masses and momenta in the propagators of the topology **`topo`** by the scaling parameter **`la`** according to the scaling rules in **`rules`**. The **`id`** of the topology remains unchanged. This is useful e.g. for asymptotic expansions of the corresponding loop integrals given as GLIs.

The scaling variable should be declared as **`FCVariable`** via the **`DataType`** mechanism.

Notice that if all terms in a propagator have the same scaling, the scaling variable in the respective propagator will be set to unity.

9.56.1 See also

Overview, FCTopology, GLI.

9.56.2 Examples

```
DataType[la, FCVariable] = True;
```

We declare the external 4-momentum **q** as our hard scale, while the mass **mc** is soft

```
topoScaled = FCLoopAddScalingParameter[FCTopology[prop1LtopoC11, {SFAD[{{I
p1, 0}, {-mc^2, -1}, 1}],
     SFAD[{{I (p1 - q), 0}, {-mc^2, -1}, 1}]}, {p1}, {q}, {SPD[q, q]
->mb^2}, {}], la,
   {q -> la^0 q, mc -> la^1 mc}]
```

Scalings of momenta and masses in the propagators of prop1LtopoC11 : $\{\text{mb} \to \text{la}^0\ \text{mb}, \text{mc} \to \text{la}\ \text{mc}, \text{p1} \to \text{la}^0\ \text{p1}, q \to \text{la}^0 q\}$

$$\text{FCTopology}\left(\text{prop1LtopoC11}, \left\{\frac{1}{(\text{la}^2\ \text{mc}^2 - \text{p1}^2 - i\eta)}, \frac{1}{(-\text{mb}^2 + \text{la}^2\ \text{mc}^2 - \text{p1}^2 + 2(\text{p1}\cdot q) - i\eta)}\right\}, \{\text{p1}\}, \{q\}, \{q^2 \to \text{mb}^2\}, \{\}\right)$$

Having set up the scaling we can now use **FCLoopGLIExpand** to expand the loop integrals belonging to this topology up to the desired order in **la**. Here we choose $\mathcal{O}(\lambda^4)$

```
FCLoopGLIExpand[GLI[prop1LtopoC11, {1, 1}], {topoScaled}, {la, 0, 4}]
```

$$\left\{\text{la}^4\ \text{mc}^4 G^{\text{prop1LtopoC11}}(1,3) + \text{la}^4\ \text{mc}^4 G^{\text{prop1LtopoC11}}(2,2) + \text{la}^4\ \text{mc}^4 G^{\text{prop1LtopoC11}}(3,1) - \text{la}^2\ \text{mc}^2 G^{\text{prop1LtopoC11}}(1,2) - \text{la}^2\ \text{mc}^2 G^{\text{prop1LtopoC11}}(2,1) + G^{\text{prop1LtopoC11}}(1,1), \left\{\text{FCTopology}\left(\text{prop1LtopoC11}, \left\{\frac{1}{(-\text{p1}^2 - i\eta)}, \frac{1}{(-\text{mb}^2 - \text{p1}^2 + 2(\text{p1}\cdot q) - i\eta)}\right\}, \{\text{p1}\}, \{q\}, \{q^2 \to \text{mb}^2\}, \{\}\right)\right\}\right\}$$

9.57 FCLoopGLIExpand

`FCLoopGLIExpand[exp, topos, {x, x0, n}]` expands **`GLI`**s defined via the list of topologies **`topos`** in **`exp`** around **`x=x0`** to order **`n`**. Here **`x`** must be a scalar quantity, e.g. a mass or a scalar product.

This routine is particularly useful for doing asymptotic expansions of integrals or amplitudes.

Notice that the series is assumed to be well-defined. The function has no built-in checks against singular behavior.

9.57.1 See also

Overview, FCTopology, GLI, FCLoopGLIDifferentiate, ToGFAD.

9.57.2 Examples

```
FCLoopGLIExpand[x GLI[tad2l, {1, 1, 1}],
  {FCTopology[tad2l, {FAD[{p1, m1}], FAD[{p2, m2}], FAD[{p1 - p2, m3}]},
    {p1, p2}, {}, {}, {}]}, {m1, 0, 2}]
```

$$\left\{\text{m1}^2 xG^{\text{tad2l}}(2,1,1)+xG^{\text{tad2l}}(1,1,1),\left\{\text{FCTopology}\left(\text{tad2l},\left\{\frac{1}{\text{p1}^2},\frac{1}{\text{p2}^2-\text{m2}^2},\frac{1}{(\text{p1}-\text{p2})^2-\text{m3}^2}\right\},\{\text{p1},\text{p2}\},\{\},\{\},\{\}\right)\right\}\right\}$$

```
FCLoopGLIExpand[x GLI[tad2l, {1, 1, 1}],
  {FCTopology[tad2l, {FAD[{p1, m1}], FAD[{p2, m2}], FAD[{p1 - p2, m3}]},
    {p1, p2}, {}, {}, {}]}, {m1, M, 4}]
```

$$\begin{aligned}&\left\{16M^8 xG^{\text{tad2l}}(5,1,1)-64M^7\,\text{m1}\,xG^{\text{tad2l}}(5,1,1)+96M^6\,\text{m1}^2 xG^{\text{tad2l}}(5,1,\right.\\&1)+4M^6 xG^{\text{tad2l}}(4,1,1)-64M^5\,\text{m1}^3 xG^{\text{tad2l}}(5,1,1)-24M^5\,\text{m1}\,xG^{\text{tad2l}}(4,1,\\&1)+16M^4\,\text{m1}^4 xG^{\text{tad2l}}(5,1,1)+48M^4\,\text{m1}^2 xG^{\text{tad2l}}(4,1,1)+M^4 xG^{\text{tad2l}}(3,1,\\&1)-40M^3\,\text{m1}^3 xG^{\text{tad2l}}(4,1,1)+12M^2\,\text{m1}^4 xG^{\text{tad2l}}(4,1,1)-2M^2\,\text{m1}^2 xG^{\text{tad2l}}(3,1,\\&1)-M^2 xG^{\text{tad2l}}(2,1,1)+\text{m1}^4 xG^{\text{tad2l}}(3,1,1)+\text{m1}^2 xG^{\text{tad2l}}(2,1,1)+xG^{\text{tad2l}}(1,1,1),\\&\left.\left\{\text{FCTopology}\left(\text{tad2l},\left\{\frac{1}{\text{p1}^2-M^2},\frac{1}{\text{p2}^2-\text{m2}^2},\frac{1}{(\text{p1}-\text{p2})^2-\text{m3}^2}\right\},\{\text{p1},\text{p2}\},\{\},\{\},\{\}\right)\right\}\right\}\end{aligned}$$

```
FCLoopGLIExpand[m2^2 GLI[tad2l, {1, 1, 1}],
  {FCTopology[tad2l, {FAD[{p1, m1}], FAD[{p2, m2}], FAD[{p1 - p2, m3}]},
    {p1, p2}, {}, {}, {}]}, {m2, 0, 6}]
```

$$\left\{\text{m2}^6 G^{\text{tad2l}}(1,3,1)+\text{m2}^4 G^{\text{tad2l}}(1,2,1)+\text{m2}^2 G^{\text{tad2l}}(1,1,1),\left\{\text{FCTopology}\left(\text{tad2l},\left\{\frac{1}{\text{p1}^2-\text{m1}^2},\frac{1}{\text{p2}^2},\frac{1}{(\text{p1}-\text{p2})^2-\text{m3}^2}\right\},\{\text{p1},\text{p2}\},\{\},\{\},\{\}\right)\right\}\right\}$$

```
FCLoopGLIExpand[ GLI[prop1l, {1, 1}] + SPD[q] GLI[prop1l, {1, 0}],
  {FCTopology[prop1l, {FAD[{p1, m1}], FAD[{p1 + q, m2}]}, {p1}, {q}, {},
  {}]}, {SPD[q], 0, 1}]
```

$$\left\{q^2 G^{\text{prop1l}}(1,0)-q^2 G^{\text{prop1l}}(1,2)+G^{\text{prop1l}}(1,1),\left\{\text{FCTopology}\left(\text{prop1l},\left\{\frac{1}{(\text{p1}^2-\text{m1}^2+i\eta)},\frac{1}{(-\text{m2}^2+\text{p1}^2+2(\text{p1}\cdot q)+i\eta)}\right\},\{\text{p1}\},\{q\},\{\},\{\}\right)\right\}\right\}$$

9.58 FCLoopIBPReducableQ

`FCLoopIBPReducableQ[int]` checks if the integral contains propagators raised to integer powers.

9.58.1 See also

Overview

9.58.2 Examples

```
FAD[q, q - p]

FCLoopIBPReducableQ[FCI[%]]
```

$$\frac{1}{q^2.(q-p)^2}$$

False

```
FAD[{q, 0, 2}, q - p]

FCLoopIBPReducableQ[FCI[%]]
```

$$\frac{1}{(q^2)^2.(q-p)^2}$$

True

9.59 FCLoopIntegralToPropagators

`FCLoopIntegralToPropagators[int, {q1, q2, ...}]` is an auxiliary function that converts the loop integral **`int`** that depends on the loop momenta **`q1, q2, ...`** to a list of propagators and scalar products.

All propagators and scalar products that do not depend on the loop momenta are discarded, unless the **`Rest`** option is set to **`True`**.

9.59.1 See also

Overview, FCLoopPropagatorsToTopology

9.59.2 Examples

```
SFAD[p1]

FCLoopIntegralToPropagators[%, {p1}]
```

$$\frac{1}{(\text{p1}^2 + i\eta)}$$

$$\left\{\frac{1}{(\text{p1}^2 + i\eta)}\right\}$$

```
SFAD[p1, p2]

FCLoopIntegralToPropagators[%, {p1, p2}]
```

$$\frac{1}{(\text{p1}^2 + i\eta).(\text{p2}^2 + i\eta)}$$

$$\left\{\frac{1}{(\text{p1}^2 + i\eta)}, \frac{1}{(\text{p2}^2 + i\eta)}\right\}$$

If the integral contains propagators raised to integer powers, only one propagator will appear in the output.

```
int = SPD[q, p] SFAD[q, q - p, q - p]
```

$$\frac{p \cdot q}{(q^2 + i\eta).((q-p)^2 + i\eta)^2}$$

```
FCLoopIntegralToPropagators[int, {q}]
```

$$\left\{(p\cdot q+i\eta),\frac{1}{(q^2+i\eta)},\frac{1}{((q-p)^2+i\eta)}\right\}$$

However, setting the option **`Tally`** to **`True`** will count the powers of the appearing propagators.

```
FCLoopIntegralToPropagators[int, {q}, Tally -> True]
```

$$\begin{pmatrix}\frac{1}{(q^2+i\eta)} & 1\\ (p\cdot q+i\eta) & 1\\ \frac{1}{((q-p)^2+i\eta)} & 2\end{pmatrix}$$

Here is a more realistic 3-loop example

```
int = SFAD[{{-k1, 0}, {mc^2, 1}, 1}] SFAD[{{-k1 - k2, 0}, {mc^2, 1}, 1}]
SFAD[{{-k2, 0}, {0, 1}, 1}] SFAD[{{-k2, 0}, {0, 1}, 2}] SFAD[{{-k3, 0},
{mc^2, 1}, 1}] *SFAD[{{k1 - k3 - p1, 0}, {0, 1}, 1}] SFAD[{{-k1 - k2 + k3 +
p1, 0}, {0, 1}, 1}] SFAD[{{-k1 - k2 + k3 + p1, 0}, {0, 1}, 2}]
```

$$\frac{1}{(\text{k2}^2+i\eta)^3(\text{k1}^2-\text{mc}^2+i\eta)(\text{k3}^2-\text{mc}^2+i\eta)((-\text{k1}-\text{k2})^2-\text{mc}^2+i\eta)((\text{k1}-\text{k3}-\text{p1})^2+i\eta)((-\text{k1}-\text{k2}+\text{k3}+\text{p1})^2$$

```
FCLoopIntegralToPropagators[int, {k1, k2, k3}, Tally -> True]
```

$$\begin{pmatrix}\frac{1}{(\text{k2}^2+i\eta)} & 3\\ \frac{1}{(\text{k3}^2-\text{mc}^2+i\eta)} & 1\\ \frac{1}{(\text{k1}^2-\text{mc}^2+i\eta)} & 1\\ \frac{1}{((\text{k1}-\text{k3}-\text{p1})^2+i\eta)} & 1\\ \frac{1}{((-\text{k1}-\text{k2}+\text{k3}+\text{p1})^2+i\eta)} & 3\\ \frac{1}{((-\text{k1}-\text{k2})^2-\text{mc}^2+i\eta)} & 1\end{pmatrix}$$

9.60 FCLoopIsolate

`FCLoopIsolate[expr, {q1, q2, ...}]` wraps loop integrals into heads specified by the user. This is useful when you want to know which loop integrals appear in the given expression.

9.60.1 See also

Overview, FCLoopExtract.

9.60.2 Examples

```
FCI[GSD[q - p1] . (GSD[q - p2] + M) . GSD[p3] SPD[q, p2] FAD[q, q - p1, {q
- p2, m}]]

FCLoopIsolate[%, {q}, Head -> loopInt]

Cases2[%, loopInt]
```

$$\frac{(\text{p2}\cdot q)(\gamma\cdot(q-\text{p1})).(M+\gamma\cdot(q-\text{p2})).(\gamma\cdot\ \text{p3})}{q^2.(q-\text{p1})^2.((q-\text{p2})^2-m^2)}$$

$$\begin{aligned}&((\gamma\cdot\ \text{p1}).(\gamma\cdot\ \text{p2}).(\gamma\cdot\ \text{p3})-M(\gamma\cdot\ \text{p1}).(\gamma\cdot\ \text{p3}))\ \text{loopInt}\left(\frac{\text{p2}\cdot q}{q^2.(q-\text{p1})^2.((q-\text{p2})^2-m^2)}\right)\\&+M\ \text{loopInt}\left(\frac{(\text{p2}\cdot q)(\gamma\cdot q).(\gamma\cdot\ \text{p3})}{q^2.(q-\text{p1})^2.((q-\text{p2})^2-m^2)}\right)-\text{loopInt}\left(\frac{(\text{p2}\cdot q)(\gamma\cdot\ \text{p1}).(\gamma\cdot q).(\gamma\cdot\ \text{p3})}{q^2.(q-\text{p1})^2.((q-\text{p2})^2-m^2)}\right)\\&-\text{loopInt}\left(\frac{(\text{p2}\cdot q)(\gamma\cdot q).(\gamma\cdot\ \text{p2}).(\gamma\cdot\ \text{p3})}{q^2.(q-\text{p1})^2.((q-\text{p2})^2-m^2)}\right)+\text{loopInt}\left(\frac{(\text{p2}\cdot q)(\gamma\cdot q).(\gamma\cdot q).(\gamma\cdot\ \text{p3})}{q^2.(q-\text{p1})^2.((q-\text{p2})^2-m^2)}\right)\end{aligned}$$

$$\begin{aligned}&\left\{\text{loopInt}\left(\frac{\text{p2}\cdot q}{q^2.(q-\text{p1})^2.((q-\text{p2})^2-m^2)}\right),\text{loopInt}\left(\frac{(\text{p2}\cdot q)(\gamma\cdot q).(\gamma\cdot\ \text{p3})}{q^2.(q-\text{p1})^2.((q-\text{p2})^2-m^2)}\right),\right.\\&\text{loopInt}\left(\frac{(\text{p2}\cdot q)(\gamma\cdot\ \text{p1}).(\gamma\cdot q).(\gamma\cdot\ \text{p3})}{q^2.(q-\text{p1})^2.((q-\text{p2})^2-m^2)}\right),\text{loopInt}\left(\frac{(\text{p2}\cdot q)(\gamma\cdot q).(\gamma\cdot\ \text{p2}).(\gamma\cdot\ \text{p3})}{q^2.(q-\text{p1})^2.((q-\text{p2})^2-m^2)}\right),\\&\left.\text{loopInt}\left(\frac{(\text{p2}\cdot q)(\gamma\cdot q).(\gamma\cdot q).(\gamma\cdot\ \text{p3})}{q^2.(q-\text{p1})^2.((q-\text{p2})^2-m^2)}\right)\right\}\end{aligned}$$

```
TID[FVD[q, \[Mu]] FVD[q, \[Nu]] FAD[{q, m}, {q + p, m}, {q + r, m}], q,
UsePaVeBasis -> True]

FCLoopIsolate[%, {q}, Head -> l]

Cases2[%, l]
```

$$\begin{aligned}&i\pi^2 g^{\mu\nu}\ \text{C}_{00}\left(p^2,r^2,-2(p\cdot r)+p^2+r^2,m^2,m^2,m^2\right)+i\pi^2 p^\mu p^\nu\ \text{C}_{11}\left(p^2,\right.\\&\left.-2(p\cdot r)+p^2+r^2,r^2,m^2,m^2,m^2\right)+i\pi^2 r^\mu r^\nu\ \text{C}_{11}\left(r^2,-2(p\cdot r)+p^2+r^2,p^2,\right.\\&\left.m^2,m^2,m^2\right)+i\pi^2\left(p^\nu r^\mu+p^\mu r^\nu\right)\ \text{C}_{12}\left(p^2,-2(p\cdot r)+p^2+r^2,r^2,m^2,m^2,m^2\right)\end{aligned}$$

$$i\pi^2 g^{\mu\nu} l\left(\mathrm{C}_{00}\left(p^2, r^2, -2(p\cdot r)+p^2+r^2, m^2, m^2, m^2\right)\right)+i\pi^2 p^\mu p^\nu l\left(\mathrm{C}_{11}\left(p^2,\right.\right.$$
$$\left.\left.-2(p\cdot r)+p^2+r^2, r^2, m^2, m^2, m^2\right)\right)+i\pi^2 r^\mu r^\nu l\left(\mathrm{C}_{11}\left(r^2, -2(p\cdot r)+p^2+r^2, p^2,\right.\right.$$
$$\left.\left.m^2, m^2, m^2\right)\right)+i\pi^2\left(p^\nu r^\mu+p^\mu r^\nu\right) l\left(\mathrm{C}_{12}\left(p^2, -2(p\cdot r)+p^2+r^2, r^2, m^2, m^2, m^2\right)\right)$$

$$\left\{l\left(\mathrm{C}_{00}\left(p^2, r^2, -2(p\cdot r)+p^2+r^2, m^2, m^2, m^2\right)\right), l\left(\mathrm{C}_{11}\left(p^2, -2(p\cdot r)+p^2+r^2, r^2, m^2, m^2, m^2\right)\right),\right.$$
$$\left. l\left(\mathrm{C}_{11}\left(r^2, -2(p\cdot r)+p^2+r^2, p^2, m^2, m^2, m^2\right)\right), l\left(\mathrm{C}_{12}\left(p^2, -2(p\cdot r)+p^2+r^2, r^2, m^2, m^2, m^2\right)\right)\right\}$$

```
SPD[q, q]^2 FAD[{q, m}] + SPD[q, q]

FCLoopIsolate[%, {q}, DropScaleless -> True]
```

$$\frac{q^4}{q^2-m^2}+q^2$$

$$\mathrm{FCGV(LoopInt)}\left(\frac{q^4}{q^2-m^2}\right)$$

```
a FAD[{q1, m}, {q2, m}] + b FAD[{q1, m, 2}]

FCLoopIsolate[%, {q1, q2}]

FCLoopIsolate[%%, {q1, q2}, MultiLoop -> True]
```

$$\frac{a}{\left(\mathrm{q1}^2-m^2\right).\left(\mathrm{q2}^2-m^2\right)}+\frac{b}{\left(\mathrm{q1}^2-m^2\right)^2}$$

$$a\ \mathrm{FCGV(LoopInt)}\left(\frac{1}{\left(\mathrm{q1}^2-m^2\right).\left(\mathrm{q2}^2-m^2\right)}\right)+b\ \mathrm{FCGV(LoopInt)}\left(\frac{1}{\left(\mathrm{q1}^2-m^2\right)^2}\right)$$

$$a\ \mathrm{FCGV(LoopInt)}\left(\frac{1}{\left(\mathrm{q1}^2-m^2\right).\left(\mathrm{q2}^2-m^2\right)}\right)+\frac{b}{\left(\mathrm{q1}^2-m^2\right)^2}$$

9.61 FCLoopMixedIntegralQ

`FCLoopMixedIntegralQ[exp]` returns **`True`** if the integral contains both Lorentz and Cartesian indices and momenta.

9.61.1 See also

Overview

9.61.2 Examples

```
FCI[FVD[p, mu] CFAD[q, q - p]]

FCLoopMixedIntegralQ[%]
```

$$\frac{p^{\text{mu}}}{(q^2 - i\eta).((q-p)^2 - i\eta)}$$

True

```
FCI[FVD[p, mu] FAD[q, q - p]]

FCLoopMixedIntegralQ[%]
```

$$\frac{p^{\text{mu}}}{q^2.(q-p)^2}$$

False

9.62 FCLoopMixedToCartesianAndTemporal

`FCLoopMixedToCartesianAndTemporal[int, {q1, q2, ...}]` attempts to convert loop integrals that contain both Lorentz and Cartesian or temporal indices/momenta to pure temporal and Cartesian indices.

9.62.1 See also

Overview

9.62.2 Examples

```
FCI@SFAD[q]

FCLoopMixedToCartesianAndTemporal[%, {q}, FCE -> True]
```

$$\frac{1}{(q^2+i\eta)}$$

$$-\frac{1}{(q^2-(q^0)^2-i\eta)}$$

```
FCI@SFAD[{q1 + q2 + p, m^2}]

FCLoopMixedToCartesianAndTemporal[%, {q1, q2}]
```

$$\frac{1}{((p+\text{q1}+\text{q2})^2-m^2+i\eta)}$$

$$-\frac{1}{((p+\text{q1}+\text{q2})^2+m^2-((p+\text{q1}+\text{q2})^0)^2-i\eta)}$$

```
FCI[TC[k] FVD[k, mu] FAD[k, k + p]]

FCLoopMixedToCartesianAndTemporal[%, {k}]
```

$$\frac{k^0 k^{\text{mu}}}{k^2.(k+p)^2}$$

$$\frac{k^0\left(k^0 \bar{g}^{0\text{mu}}-k^{\$} g^{\$\text{mu}}\right)}{(k^2-(k^0)^2-i\eta).((k+p)^2-((k+p)^0)^2-i\eta)}$$

9.63 FCLoopNonIntegerPropagatorPowersFreeQ

`FCLoopNonIntegerPropagatorPowersFreeQ[int]` checks if the integral contains propagators raised to noninteger (i.e. fractional or symbolic) powers.

9.63.1 See also

Overview, FCLoopRemoveNegativePropagatorPowers.

9.63.2 Examples

```
SFAD[{q + p, m^2, 2}]

FCLoopNonIntegerPropagatorPowersFreeQ[FCI[%]]
```

$$\frac{1}{((p+q)^2 - m^2 + i\eta)^2}$$

True

```
SFAD[{q + p, m^2, n}]

FCLoopNonIntegerPropagatorPowersFreeQ[FCI[%]]
```

$$((p+q)^2 - m^2 + i\eta)^{-n}$$

False

```
CFAD[{l, m^2, 1/2}]

FCLoopNonIntegerPropagatorPowersFreeQ[FCI[%]]
```

$$\frac{1}{\sqrt{(l^2 + m^2 - i\eta)}}$$

False

9.64 FCLoopPakScalelessQ

FCLoopPakScalelessQ[poly, x] checks whether the characteristic polynomial poly (in the $U \times xF$ form) with the Feynman parameters **x[1], x[2], ...** corresponds to a scaleless loop integral or loop integral topology. The polynomial does not need to be canonically ordered.

The function uses the algorithm of Alexey Pak arXiv:1111.0868. Cf. also the PhD thesis of Jens Hoff 10.5445/IR/1000047447 for the detailed description of a possible implementation. **FCLoopPakScalelessQ** is a backend function used in **FCLoopScalelessQ**, **FCLoopFindSubtopologies** etc.

9.64.1 See also

Overview, FCTopology, GLI, FCLoopToPakForm, FCLoopScalelessQ.

9.64.2 Examples

A scaleless 2-loop tadpole is a clear case, since here the characteristic polynomial vanishes

```
int = FAD[p1, p2, p1 - p2]

pf = FCLoopToPakForm[int, {p1, p2}, Names -> x,

     CharacteristicPolynomial -> Function[{u, f}, u f]][[2]][[1]]
```

$$\frac{1}{\text{p1}^2.\text{p2}^2.(\text{p1}-\text{p2})^2}$$

$$0$$

```
FCLoopPakScalelessQ[pf, x]
```

True

A somewhat less obvious (but still simple) case is this 1-loop eikonal integral.

```
int = SFAD[{{0, 2 p . q}, 0}, p]

pf = FCLoopToPakForm[int, {p}, Names -> x,

     CharacteristicPolynomial -> Function[{u, f}, u f]][[2]][[1]]
```

$$\frac{1}{(2(p\cdot q)+i\eta).(p^2+i\eta)}$$

$$q^2 x(1)^2 x(2)$$

```
FCLoopPakScalelessQ[pf, x]
```

True

Adding a mass term to the quadratic propagator makes this integral nonvanishing

```
int = SFAD[{{0, 2 p . q}, 0}, {p, m^2}]

pf = FCLoopToPakForm[int, {p}, Names -> x,

     CharacteristicPolynomial -> Function[{u, f}, u f]][[2]][[1]]
```

$$\frac{1}{(2(p \cdot q) + i\eta).(p^2 - m^2 + i\eta)}$$

$$m^2 x(1)^3 + q^2 x(2)^2 x(1)$$

```
FCLoopPakScalelessQ[pf, x]
```

False

Notice that **FCLoopPakScalelessQ** is more of an auxiliary function. The corresponding end-user function is called **FCLoopScalelessQ**

```
FCLoopScalelessQ[SFAD[{{0, 2 p . q}, 0}, p], {p}]
```

True

9.65 FCLoopScalelessQ

FCLoopScalelessQ[int, {p1, p2, ...}] checks whether the loop integral **int** depending on the loop momenta **p1, p2, ...** is scaleless. Only integrals that admit a Feynman parametrization with proper U and F polynomials are supported.

The function uses the algorithm of Alexey Pak arXiv:1111.0868. Cf. also the PhD thesis of Jens Hoff 10.5445/IR/1000047447 for the detailed description of a possible implementation.

9.65.1 See also

Overview, FCTopology, GLI, FCLoopToPakForm, FCLoopPakOrder, FCLoopScalelessQ.

9.65.2 Examples

A scaleless 2-loop tadpole

```
FCLoopScalelessQ[FAD[p1, p2, p1 - p2], {p1, p2}]
```

True

A 1-loop massless eikonal integral.

```
FCLoopScalelessQ[SFAD[{{0, 2 p . q}, 0}, p], {p}]
```

True

A 1-loop massive eikonal integral.

```
FCLoopScalelessQ[SFAD[{{0, 2 p . q}, 0}, p], {p}]
```

True

A scaleless topology

```
FCTopology[topo, {SFAD[{{I p3, 0}, {0, 1}, 1}], SFAD[{{I p1, 0}, {0, 1},
1}],
    SFAD[{{0, -2 p1 . q}, {0, 1}, 1}], SFAD[{{I p3 + I q, 0}, {-mb^2, 1},
1}],
    SFAD[{{0, p1 . p3}, {0, 1}, 1}]}, {p1, p3}, {q}, {}, {}]

FCLoopScalelessQ[%]
```

$$\text{FCTopology}\left(\text{topo}, \left\{\frac{1}{(-\text{p3}^2+i\eta)}, \frac{1}{(-\text{p1}^2+i\eta)}, \frac{1}{(-2(\text{p1}\cdot q)+i\eta)}, \frac{1}{((i\ \text{p3}+iq)^2+\text{mb}^2+i\eta)}, \frac{1}{(\text{p1}\cdot\ \text{p3}+i\eta)}\right\}, \{\text{p1}, \text{p3}\}, \{q\}, \{\}, \{\}\right)$$

True

9.66 FCLoopPropagatorPowersCombine

`FCLoopPropagatorPowersCombine[exp]` combines the same propagators in a **`FeynAmpDenominator`** to one propagator raised to an integer power.

9.66.1 See also

Overview, FCLoopPropagatorPowersExpand.

9.66.2 Examples

```
SFAD[{{q, 0}, {m, 1}, 1}, {{q, 0}, {m, 1}, 1}]

ex = FCLoopPropagatorPowersCombine[%]
```

$$\frac{1}{(q^2 - m + i\eta)^2}$$

$$\frac{1}{(q^2 - m + i\eta)^2}$$

```
ex // StandardForm

(*FeynAmpDenominator[StandardPropagatorDenominator[Momentum[q, D], 0, -m,
{2, 1}]]*)
```

```
SFAD[{{q, 0}, {m, 1}, -1}, {{q, 0}, {m, 1}, 1}]

ex = FCLoopPropagatorPowersCombine[%]
```

$$\frac{1}{\frac{1}{(q^2-m+i\eta)} \cdot (q^2 - m + i\eta)}$$

$$1$$

```
ex // StandardForm

(*1*)
```

The function automatically employs **FeynAmpDenominatorCombine**.

```
int = SFAD[{{-k1, 0}, {mc^2, 1}, 1}]  SFAD[{{-k1 - k2 + k3 + p1, 0}, {0,
1}, 1}] SFAD[{{-k1 - k2 + k3 + p1, 0}, {0, 1}, 2}]
```

$$\frac{1}{(\text{k1}^2 - \text{mc}^2 + i\eta)((-\text{k1} - \text{k2} + \text{k3} + \text{p1})^2 + i\eta)^3}$$

```
int // FCI // StandardForm

(*FeynAmpDenominator[StandardPropagatorDenominator[-Momentum[k1, D], 0,
-mc^2, {1, 1}]]
FeynAmpDenominator[StandardPropagatorDenominator[-Momentum[k1, D] -
Momentum[k2, D] + Momentum[k3, D] + Momentum[p1, D], 0, 0, {1, 1}]]
FeynAmpDenominator[StandardPropagatorDenominator[-Momentum[k1, D] -
Momentum[k2, D] + Momentum[k3, D] + Momentum[p1, D], 0, 0, {2, 1}]]*)
```

```
res = FCLoopPropagatorPowersCombine[int]
```

$$\frac{1}{(\text{k1}^2 - \text{mc}^2 + i\eta).((-\text{k1} - \text{k2} + \text{k3} + \text{p1})^2 + i\eta)^3}$$

```
res // FCI // StandardForm

(*FeynAmpDenominator[StandardPropagatorDenominator[-Momentum[k1, D], 0,
-mc^2, {1, 1}], StandardPropagatorDenominator[-Momentum[k1, D] -
Momentum[k2, D] + Momentum[k3, D] + Momentum[p1, D], 0, 0, {3, 1}]]*)
```

9.67 FCLoopPropagatorPowersExpand

FCLoopPropagatorPowersExpand[exp] rewrites propagators raised to integer powers as products.

9.67.1 See also

Overview, FCLoopPropagatorPowersCombine.

9.67.2 Examples

```
SFAD[{q, m, 2}]

ex = FCLoopPropagatorPowersExpand[%]
```

$$\frac{1}{(q^2 - m + i\eta)^2}$$

$$\frac{1}{(q^2 - m + i\eta)^2}$$

```
ex // StandardForm

(*FeynAmpDenominator[StandardPropagatorDenominator[Momentum[q, D], 0, -m,
{1, 1}], StandardPropagatorDenominator[Momentum[q, D], 0, -m, {1, 1}]]*)
```

```
SFAD[{q, m, 2}, q + p]

ex = FCLoopPropagatorPowersExpand[%]
```

$$\frac{1}{(q^2 - m + i\eta)^2.((p+q)^2 + i\eta)}$$

$$\frac{1}{(q^2 - m + i\eta)^2.((p+q)^2 + i\eta)}$$

```
ex // StandardForm

(*FeynAmpDenominator[StandardPropagatorDenominator[Momentum[q, D], 0, -m,
{1, 1}], StandardPropagatorDenominator[Momentum[q, D], 0, -m, {1, 1}],
StandardPropagatorDenominator[Momentum[p, D] + Momentum[q, D], 0, 0, {1,
1}]]*)
```

9.68 FCLoopPropagatorsToTopology

FCLoopPropagatorsToTopology[{prop1, prop2, ...}] takes a list of **Pair**s and **FeynAmpDenominator**s and converts it into a list of propagators that can be used to describe a topology.

The input can also consist of an **FCTopology** object or a list thereof.

9.68.1 See also

Overview, FCLoopIntegralToPropagators.

9.68.2 Examples

```
{FAD[q]}

FCLoopPropagatorsToTopology[%]
```

$$\left\{\frac{1}{q^2}\right\}$$

$$\{q^2\}$$

```
{FAD[{q, m}]}

FCLoopPropagatorsToTopology[%]
```

$$\left\{\frac{1}{q^2-m^2}\right\}$$

$$\{q^2-m^2\}$$

```
{FAD[{q, m}], SPD[q, p]}

FCLoopPropagatorsToTopology[%]
```

$$\left\{\frac{1}{q^2-m^2}, p\cdot q\right\}$$

$$\{q^2-m^2, p\cdot q\}$$

```
FCLoopPropagatorsToTopology[{FCTopology[topo1, {SFAD[{{p1, 0}, {0, 1}, 1}],
     SFAD[{{p3, 0}, {mb^2, 1}, 1}], SFAD[{{p1 + p3, 0}, {mb^2, 1}, 1}],
SFAD[{{p1 - q, 0},
       {mb^2, 1}, 1}], SFAD[{{0, p3 . q}, {0, 1}, 1}]},  {p1, p3}, {q}, {},
       {}],
   FCTopology[topo1, {SFAD[{{p1, 0}, {mb^2, 1}, 1}], SFAD[{{p3, 0}, {mb^2,
1}, 1}],
     SFAD[{{p1 + p3, 0}, {mb^2, 1}, 1}], SFAD[{{p1 - q, 0}, {mb^2, 1}, 1}],
     SFAD[{{0, (p3 + p1) . q}, {0, 1}, 1}]},  {p1, p3}, {q}, {}, {}]}]
```

$$\begin{pmatrix} \text{p1}^2 & \text{p3}^2-\text{mb}^2 & (\text{p1}+\text{p3})^2-\text{mb}^2 & (\text{p1}-q)^2-\text{mb}^2 & \text{p3}\cdot q \\ \text{p1}^2-\text{mb}^2 & \text{p3}^2-\text{mb}^2 & (\text{p1}+\text{p3})^2-\text{mb}^2 & (\text{p1}-q)^2-\text{mb}^2 & (\text{p1}+\text{p3})\cdot q \end{pmatrix}$$

9.69 FCLoopSamePropagatorHeadsQ

FCLoopSamePropagatorHeadsQ[exp] returns **True** if the **FeynAmpDenominator** of **exp** contains only propagator denominators of the same type (e.g. only **StandardPropagatorDenominator** or only **CartesianPropagatorDenominator**).

9.69.1 See also

Overview, FeynAmpDenominator.

9.69.2 Examples

```
FCI@SFAD[q, q - p]

FCLoopSamePropagatorHeadsQ[%]
```

$$\frac{1}{(q^2+i\eta).((q-p)^2+i\eta)}$$

True

```
FeynAmpDenominatorCombine[CFAD[q, q - p] SFAD[l, l + k]]

FCLoopSamePropagatorHeadsQ[%]
```

$$\frac{1}{(q^2-i\eta).((q-p)^2-i\eta).(l^2+i\eta).((k+l)^2+i\eta)}$$

False

9.70 FCLoopRemoveNegativePropagatorPowers

FCLoopRemoveNegativePropagatorPowers[exp] rewrites propagators raised to integer powers as products.

9.70.1 See also

Overview

9.70.2 Examples

```
SFAD[{q, m, -1}]

ex = FCLoopRemoveNegativePropagatorPowers[%]
```

$$(q^2 - m + i\eta)$$

$$q^2 - m$$

```
ex // StandardForm

(*-m + Pair[Momentum[q, D], Momentum[q, D]]*)
```

```
SFAD[{q, m}, q + p, {q, m, -2}]

ex = FCLoopRemoveNegativePropagatorPowers[%]
```

$$\frac{1}{(q^2 - m + i\eta).((p+q)^2 + i\eta).\frac{1}{(q^2-m+i\eta)^2}}$$

$$\frac{q^2 - m}{((p+q)^2 + i\eta)}$$

```
ex // StandardForm

(*FeynAmpDenominator[StandardPropagatorDenominator[Momentum[p, D] +
Momentum[q, D], 0, 0, {1, 1}]] (-m + Pair[Momentum[q, D], Momentum[q,
D]])*)
```

`FCLoopSelectTopology[int, topos]` selects the topology that matches the **`GLI int`** from a list of topologies **`topos`**.

The first argument can be also a list, in which case the function will return a list of matching topologies.

9.70.3 See also

Overview, FCTopology, GLI.

9.70.4 Examples

```
topos = {FCTopology[topo2, {FAD[{p1, m}], FAD[{p1 - q, m}]}, {p1}, {q}, {},
{}],
   FCTopology[topo1, {FAD[{p1, m}], FAD[{p1 - q, m}]}, {p1}, {q}, {SPD[q]
-> M^2,SPD[q2] -> M2^2}, {}]}
```

$$\left\{\text{FCTopology}\left(\text{topo2},\left\{\frac{1}{\text{p1}^2-m^2},\frac{1}{(\text{p1}-q)^2-m^2}\right\},\{\text{p1}\},\{q\},\{\},\{\}\right),\text{FCTopology}\left(\text{topo1},\left\{\frac{1}{\text{p1}^2-m^2},\frac{1}{(\text{p1}-q)^2-m^2}\right\},\{\text{p1}\},\{q\},\{q^2\to M^2,\text{q2}^2\to\text{M2}^2\},\{\}\right)\right\}$$

```
FCLoopSelectTopology[{GLI[topo2, {2, 2}]}, topos]
```

$$\left\{\text{FCTopology}\left(\text{topo2},\left\{\frac{1}{\text{p1}^2-m^2},\frac{1}{(\text{p1}-q)^2-m^2}\right\},\{\text{p1}\},\{q\},\{\},\{\}\right)\right\}$$

```
topos = {FCTopology[prop2Ltopo13311, {SFAD[{{I*p1, 0}, {-m1^2, -1}, 1}],
     SFAD[{{I*(p1 + q1), 0}, {-m3^2, -1}, 1}], SFAD[{{I*p3, 0}, {-m3^2,
-1}, 1}],
     SFAD[{{I*(p3 + q1), 0}, {-m1^2, -1}, 1}], SFAD[{{I*(p1 - p3), 0},
       {-m1^2, -1}, 1}]}, {p1, p3}, {q1}, {SPD[q1, q1] -> m1^2}, {}]}
```

$$\left\{\text{FCTopology}\left(\text{prop2Ltopo13311},\left\{\frac{1}{(-\text{p1}^2+\text{m1}^2-i\eta)},\frac{1}{(-(\text{p1}+\text{q1})^2+\text{m3}^2-i\eta)},\frac{1}{(-\text{p3}^2+\text{m3}^2-i\eta)},\frac{1}{(-(\text{p3}+\text{q1})^2+\text{m1}^2-i\eta)},\frac{1}{(-(\text{p1}-\text{p3})^2+\text{m1}^2-i\eta)}\right\},\{\text{p1},\text{p3}\},\{\text{q1}\},\{\text{q1}^2\to\text{m1}^2\},\{\}\right)\right\}$$

Products of **GLI**s in the first argument are also supported.

```
FCLoopSelectTopology[{GLI[prop2Ltopo13311, {1, 0, 0, 0, 0}]^2,
   GLI[prop2Ltopo13311, {1, 1, 1, 1, 1}]}, topos, FCE -> True]
```

$$\left\{\text{FCTopology}\left(\text{prop2Ltopo13311},\left\{\frac{1}{(-\text{p1}^2+\text{m1}^2-i\eta)},\frac{1}{(-(\text{p1}+\text{q1})^2+\text{m3}^2-i\eta)},\frac{1}{(-\text{p3}^2+\text{m3}^2-i\eta)},\frac{1}{(-(\text{p3}+\text{q1})^2+\text{m1}^2-i\eta)},\frac{1}{(-(\text{p1}-\text{p3})^2+\text{m1}^2-i\eta)}\right\},\{\text{p1},\text{p3}\},\{\text{q1}\},\{\text{q1}^2\to\text{m1}^2\},\{\}\right)\right\}$$

9.71 FCLoopSwitchEtaSign

FCLoopSwitchEtaSign[exp, s] switches the sign of $i\eta$ in all integrals to **s**, where **s** can be **+1** or **-1**. Notice to change the sign of $i\eta$ the function pulls out a factor -1 from the propagator.

9.71.1 See also

Overview, FCTopology, FCLoopGetEtaSigns.

9.71.2 Examples

FADs are automatically converted to **SFAD**s, since otherwise their $i\eta$ prescription cannot be modified

```
FAD[{p, m}]

FCLoopSwitchEtaSign[%, 1]
```

$$\frac{1}{p^2 - m^2}$$

$$\frac{1}{(p^2 - m^2 + i\eta)}$$

```
FAD[{p, m}]

FCLoopSwitchEtaSign[%, -1]
```

$$\frac{1}{p^2 - m^2}$$

$$-\frac{1}{(-p^2 + m^2 - i\eta)}$$

```
SFAD[{p, m^2}]

FCLoopSwitchEtaSign[%, -1]
```

$$\frac{1}{(p^2 - m^2 + i\eta)}$$

$$-\frac{1}{(-p^2 + m^2 - i\eta)}$$

```
CFAD[{p, m^2}]

FCLoopSwitchEtaSign[%, 1]
```

$$\frac{1}{(p^2 + m^2 - i\eta)}$$

$$-\frac{1}{(-p^2 - m^2 + i\eta)}$$

9.72 FCLoopSingularityStructure

`FCLoopSingularityStructure[int, {q1, q2, ...}]` returns a list of expressions **`{pref, U, F, gbF}`** that are useful to analyze the singular behavior of the loop integral **`int`**.

- **`pref`** is the ε-dependent prefactor of the Feynman parameter integral that can reveal an overall UV-singularity
- **`U`** and **`F`** denote the first and second Symanzik polynomials respectively
- **`gbF`** is the Groebner basis of $F, \partial F/\partial x_i$ with respect to the Feynman parameters

The idea to search for solutions of Landau equations for the F-polynomial using Groebner bases was adopted from 1810.06270 and 2003.02451 by B. Ananthanarayan, Abhishek Pal, S. Ramanan Ratan Sarkar and Abhijit B. Das.

9.72.1 See also

Overview, FCFeynmanPrepare, FCFeynmanParametrize

9.72.2 Examples

1-loop tadpole

```
out = FCLoopSingularityStructure[FAD[{q, m}], {q}, Names -> x]
```

$$\left\{\Gamma(\varepsilon-1)\left(-\left(m^2\right)^{1-\varepsilon}\right), x(1), m^2 x(1)^2, \left\{m^2 x(1)\right\}\right\}$$

The integral has an apparent UV-singularity from the prefactor

```
Normal[Series[out[[1]], {Epsilon, 0, -1}]]
```

$$\frac{m^2}{\varepsilon}$$

Massless 1-loop 2-point function

```
out = FCLoopSingularityStructure[FAD[q, q - p], {q}, Names -> x]
```

$$\left\{\Gamma(\varepsilon), x(1)+x(2), -p^2 x(1)x(2), \left\{p^2 x(2), p^2 x(1)\right\}\right\}$$

The integral has an apparent UV-singularity from the prefactor

```
Normal[Series[out[[1]], {Epsilon, 0, -1}]]
```

$$\frac{1}{\varepsilon}$$

but there is also an IR-divergence for $p^2 = 0$ (the trivial solution with all x_i being 0 is not relevant here)

```
Reduce[Equal[#, 0] & /@ out[[4]]]
```

$$(x(2) = 0 \wedge x(1) = 0) \vee p^2 = 0$$

1-loop massless box

```
out = FCLoopSingularityStructure[FAD[p, p + q1, p + q1 + q2, p + q1 + q2 +
q3], {p},
   Names -> x, FinalSubstitutions -> {SPD[q1] -> 0, SPD[q2] -> 0, SPD[q3]
   -> 0}]
```

$$\{2^{-\varepsilon-2}\Gamma(\varepsilon+2), x(1)+x(2)+x(3)+x(4),$$
$$-2(x(1)x(3)(\text{q1}\cdot\text{q2})+x(1)x(4)(\text{q1}\cdot\text{q2})+x(1)x(4)(\text{q1}\cdot\text{q3})+x(1)x(4)(\text{q2}\cdot\text{q3})+x(2)x(4)(\text{q2}\cdot\text{q3})),$$
$$\{x(4)(\text{q2}\cdot\text{q3}), x(3)(\text{q1}\cdot\text{q2})+x(4)(\text{q1}\cdot\text{q2})+x(4)(\text{q1}\cdot\text{q3}), x(2)(\text{q1}\cdot\text{q2})(\text{q2}\cdot\text{q3}),$$
$$x(1)(\text{q1}\cdot\text{q3})+x(1)(\text{q2}\cdot\text{q3})+x(2)(\text{q2}\cdot\text{q3}), x(1)(\text{q1}\cdot\text{q2})\}\}$$

As expected a 1-loop box has no overall UV-divergence

```
Normal[Series[out[[1]], {Epsilon, 0, -1}]]
```

$$0$$

The form of the U-polynomial readily suggests that there is no UV-subdivergence (again as expected)

```
Reduce[out[[2]] == 0, {x[1], x[2], x[3], x[4]}]
```

$$x(4) = -x(1) - x(2) - x(3)$$

As far as the IR-divergences are concerned, we find a rather nontrivial set of solutions satisfying Landau equations

```
Reduce[Equal[#, 0] & /@ out[[4]]]
```

$$\begin{aligned}
&(\text{q2}\cdot\text{q3} = 0 \wedge x(1) \neq 0 \wedge\ \text{q1}\cdot\text{q3} = 0 \wedge\ \text{q1}\cdot\text{q2} = 0)\\
&\vee \left(x(1) = 0 \wedge\ \text{q2}\cdot\text{q3} = 0 \wedge x(3)+x(4) \neq 0 \wedge\ \text{q1}\cdot\text{q2} = -\frac{x(4)(\text{q1}\cdot\text{q3})}{x(3)+x(4)}\right)\\
&\vee (x(4) = 0 \wedge x(1) = 0 \wedge\ \text{q2}\cdot\text{q3} = 0 \wedge\ \text{q1}\cdot\text{q2} = 0) \vee (x(4) = 0 \wedge x(2) = 0 \wedge x(1) = 0 \wedge\ \text{q1}\cdot\text{q2} = 0)\\
&\vee (x(4) = 0 \wedge x(3) = 0 \wedge x(1) = 0 \wedge\ \text{q2}\cdot\text{q3} = 0) \vee (x(4) = 0 \wedge x(3) = 0 \wedge x(2) = 0 \wedge x(1) = 0)\\
&\vee \left(x(4) = 0 \wedge x(1) \neq 0 \wedge\ \text{q1}\cdot\text{q3} = \frac{x(1)(-(\text{q2}\cdot\text{q3}))-x(2)(\text{q2}\cdot\text{q3})}{x(1)} \wedge\ \text{q1}\cdot\text{q2} = 0\right)\\
&\vee (x(1) = 0 \wedge\ \text{q2}\cdot\text{q3} = 0 \wedge x(4) \neq 0 \wedge\ \text{q1}\cdot\text{q3} = 0 \wedge\ \text{q1}\cdot\text{q2} = 0)\\
&\vee (x(3) = -x(4) \wedge x(1) = 0 \wedge\ \text{q2}\cdot\text{q3} = 0 \wedge x(4) \neq 0 \wedge\ \text{q1}\cdot\text{q3} = 0)\\
&\vee (x(4) = 0 \wedge x(1) = 0 \wedge x(2) \neq 0 \wedge\ \text{q2}\cdot\text{q3} = 0 \wedge\ \text{q1}\cdot\text{q2} = 0)\\
&\vee (x(4) = 0 \wedge x(2) = 0 \wedge x(1) = 0 \wedge x(3) \neq 0 \wedge\ \text{q1}\cdot\text{q2} = 0)
\end{aligned}$$

A 2-loop eikonal integral with massive and massless lines

```
out = FCLoopSingularityStructure[SFAD[{ p1, m^2}] SFAD[{ p3, m^2}]
SFAD[{{0,
       2 p1 . n}}] SFAD[{{0, 2 (p1 + p3) . n}}], {p1, p3}, Names -> x,
   FinalSubstitutions -> {SPD[n] -> 1, m -> 1}]
```

$$\{\Gamma(2\varepsilon), x(3)x(4), x(4)x(1)^2 + 2x(2)x(4)x(1) + x(3)x(4)^2 + x(2)^2x(3) + x(2)^2x(4) + x(3)^2x(4), \{x(3)x(4)^2, x(3)^2x(4), x(2)x(3), x(2)^2 + x(4)^2 + 2x(3)x(4), x(1)x(4) + x(2)x(4), x(1)^2 + 2x(2)x(1) + x(3)^2 - x(4)^2\}\}$$

The integral has no IR-divergence, the only solution to the Landau equations is a trivial one

```
Reduce[Equal[#, 0] & /@ out[[4]], Reals]
```

$$\begin{aligned} x(4) &= 0 \wedge x(3) \\ &= 0 \wedge x(2) \\ &= 0 \wedge x(1) \\ &= 0 \end{aligned}$$

Notice that the mass is acting as an IR regulator here. Setting it to 0 makes the IR pole resurface

```
out = FCLoopSingularityStructure[SFAD[{ p1, m^2}] SFAD[{ p3, m^2}]
SFAD[{{0,
       2 p1 . n}}] SFAD[{{0, 2 (p1 + p3) . n}}], {p1, p3}, Names -> x,
   FinalSubstitutions -> {SPD[n] -> 1, m -> 0}]
```

$$\{0, x(3)x(4), x(4)x(1)^2 + 2x(2)x(4)x(1) + x(2)^2x(3) + x(2)^2x(4), \{x(2)x(3), x(2)^2, x(1)x(4) + x(2)x(4), x(1)^2 + 2x(2)x(1)\}\}$$

and here is our nontrivial solution

```
Reduce[Equal[#, 0] & /@ out[[4]], Reals]
```

$$\begin{aligned} x(1) &= 0 \wedge x(2) \\ &= 0 \end{aligned}$$

9.73 FCLoopSolutionList

FCLoopSolutionList[loopList, reversedRepIndexList, canIndexList, uniqueCanIndexList}, solsList] is an auxiliary internal function that uses the output of FCLoopCanonicalize and the list of simplified integrals solsList to create the substitution list of type **"Integral" -> "simplified Integral"**.

9.73.1 See also

Overview, FCLoopCanonicalize.

9.73.2 Examples

```
li = FCLoopCanonicalize[myHead[FVD[q, \[Mu]] FVD[q, \[Nu]] FAD[q, {q + p,
m}]] + myHead[FVD[q, \[Rho]] FVD[q, \[Sigma]] FAD[q, {q + p, m}]], q,
myHead]
```

$$\left\{\left\{\text{myHead}\left(\frac{q^{\mu}q^{\nu}}{q^2.((p+q)^2-m^2)}\right),\text{myHead}\left(\frac{q^{\rho}q^{\sigma}}{q^2.((p+q)^2-m^2)}\right)\right\},\right.$$
$$\{\{\text{FCGV(cli191)}\to\mu,\text{FCGV(cli192)}\to\nu\},\{\text{FCGV(cli191)}\to\rho,$$
$$\text{FCGV(cli192)}\to\sigma\}\},\left\{\text{myHead}\left(\frac{q^{\text{FCGV(cli191)}}q^{\text{FCGV(cli192)}}}{q^2.((p+q)^2-m^2)}\right),\right.$$
$$\left.\left.\text{myHead}\left(\frac{q^{\text{FCGV(cli191)}}q^{\text{FCGV(cli192)}}}{q^2.((p+q)^2-m^2)}\right)\right\},\left\{\text{myHead}\left(\frac{q^{\text{FCGV(cli191)}}q^{\text{FCGV(cli192)}}}{q^2.((p+q)^2-m^2)}\right)\right\}\right\}$$

```
FCLoopSolutionList[li, prefactor (li[[4]] /. myHead -> Identity /. q -> p),
Dispatch -> False]
```

$$\left\{\text{myHead}\left(\frac{q^{\mu}q^{\nu}}{q^2.((p+q)^2-m^2)}\right)\to\frac{\text{prefactor}p^{\mu}p^{\nu}}{p^2.(4p^2-m^2)},\text{myHead}\left(\frac{q^{\rho}q^{\sigma}}{q^2.((p+q)^2-m^2)}\right)\to\frac{\text{prefactor}p^{\rho}p^{\sigma}}{p^2.(4p^2-m^2)}\right\}$$

9.74 FCLoopSplit

`FCLoopSplit[exp, {q1, q2, ...}]` separates **`exp`** into the following four pieces:

1) terms that are free of loop integrals
2) terms with scalar loop integrals
3) terms with tensor loop integrals, where all loop momenta are contracted
4) terms with tensor loop integrals, where at least some loop momenta have free indices

The result is returned as a list with the 4 above elements.

9.74.1 See also

Overview

9.74.2 Examples

```
FVD[q, \[Mu]] FAD[{q, m}]

FCLoopSplit[%, {q}]
```

$$\frac{q^\mu}{q^2 - m^2}$$

$$\left\{0, 0, 0, \frac{q^\mu}{q^2 - m^2}\right\}$$

```
x + GSD[p + q] FAD[{q, m}]

FCLoopSplit[%, {q}]
```

$$\frac{\gamma \cdot (p + q)}{q^2 - m^2} + x$$

$$\left\{x, \frac{\gamma \cdot p}{q^2 - m^2}, 0, \frac{\gamma \cdot q}{q^2 - m^2}\right\}$$

9.75 FCLoopTensorReduce

`FCLoopTensorReduce[exp, topos]` performs tensor reduction for the numerators of multi-loop integrals present in **`exp`**. Notice that **`exp`** is expected to be the output of **`FCLoopFindTopologies`**where all loop integrals have been written as **`fun[num, GLI[...]]`** with **`num`** being the numerator to be acted upon.

The reduction is done only for loop momenta contracted with Dirac matrices, polarization vectors or Levi-Civita tensors. Scalar products with external momenta are left untouched. The goal is to rewrite everything in terms of scalar products involving only loop momenta and external momenta appearing in the given topology. These quantities can be then rewritten in terms of inverse propagators (**`GLI`**s with negative indices), so that the complete dependence on loop momenta will go into the **`GLI`**s.

Unlike **`FCMultiLoopTID`**, this function does not perform any partial fractioning or shifts in the loop momenta.

The default value for **`fun`** is FCGV["GLIProduct"] set by the option **`Head`**

9.75.1 See also

Overview, FCLoopFindTopologies.

9.75.2 Examples

1-loop tadpole topology

```
topo1 = FCTopology["tad1l", {SFAD[{q, m^2}]}, {q}, {}, {}, {}]
```

$$\text{FCTopology}\left(\text{tad1l}, \left\{\frac{1}{(q^2 - m^2 + i\eta)}\right\}, \{q\}, \{\}, \{\}, \{\}\right)$$

```
amp1 = FCGV["GLIProduct"][GSD[q] . GAD[\[Mu]] . GSD[q], GLI["tad1l", {1}]]
```

$$\text{FCGV(GLIProduct)}\left((\gamma \cdot q).\gamma^{\mu}.(\gamma \cdot q), G^{\text{tad1l}}(1)\right)$$

```
amp1Red = FCLoopTensorReduce[amp1, {topo1}]
```

$$\text{FCGV(GLIProduct)}\left(\frac{(2 - D)q^2\gamma^{\mu}}{D}, G^{\text{tad1l}}(1)\right)$$

```
topo2 = FCTopology[prop1l, {SFAD[{q, m^2}, {q - p, m^2}]}, {q}, {p}, {},
{}]
```

$$\text{FCTopology}\left(\text{prop1l}, \left\{\frac{1}{(q^2 - m^2 + i\eta).((q - p)^2 - m^2 + i\eta)}\right\}, \{q\}, \{p\}, \{\}, \{\}\right)$$

1-loop self-energy topology

```
amp2 = gliProduct[GSD[q] . GAD[\[Mu]] . GSD[q], GLI[prop1l, {1, 2}]]
```

$$\text{gliProduct}\left((\gamma \cdot q).\gamma^{\mu}.(\gamma \cdot q), G^{\text{prop1l}}(1, 2)\right)$$

```
amp2Red = FCLoopTensorReduce[amp2, {topo2}, Head -> gliProduct]
```

$$\text{gliProduct}\left(-\frac{2Dp^{\mu}\gamma \cdot p(p \cdot q)^2 - 2p^2\gamma^{\mu}(p \cdot q)^2 - Dp^4q^2\gamma^{\mu} + 3p^4q^2\gamma^{\mu} - 2p^2q^2p^{\mu}\gamma \cdot p}{(1 - D)p^4}, G^{\text{prop1l}}(1, 2)\right)$$

If the loop momenta are contracted with some external momenta that do not appear in the given integral topologies, they should be listed via the option **`Uncontract`**

```
amp3 = gliProduct[SPD[q, x], GLI[prop1l, {1, 2}]]
```

$$\text{gliProduct}\left(q \cdot x, G^{\text{prop1l}}(1,2)\right)$$

```
FCLoopTensorReduce[amp3, {topo2}, Uncontract -> {x}, Head -> gliProduct]
```

$$\text{gliProduct}\left(\frac{(p \cdot q)(p \cdot x)}{p^2}, G^{\text{prop1l}}(1,2)\right)$$

9.76 FCLoopValidTopologyQ

`FCLoopValidTopologyQ[topo]` returns **`True`** if **`topo`** is a valid **`FCTopology`** object or a list thereof.

9.76.1 See also

Overview, FCTopology.

9.76.2 Examples

This is a valid topology: it has an id, a list of propagators, a list of loop and external momenta, a list of possible substitutions for kinematic invariants and an empty list reserved for future applications

```
{FAD[p1], FAD[p2], FAD[p3], FAD[Q - p1 - p2 - p3], FAD[Q - p1 - p2], FAD[Q
- p1], FAD[Q - p2], FAD[p1 + p3]}
```

$$\left\{\frac{1}{\text{p1}^2}, \frac{1}{\text{p2}^2}, \frac{1}{\text{p3}^2}, \frac{1}{(-\text{p1}-\text{p2}-\text{p3}+Q)^2}, \frac{1}{(-\text{p1}-\text{p2}+Q)^2}, \frac{1}{(Q-\text{p1})^2}, \frac{1}{(Q-\text{p2})^2}, \frac{1}{(\text{p1}+\text{p3})^2}\right\}$$

```
topo = FCTopology[topo1, {FAD[p1], FAD[p2], FAD[p3], FAD[Q - p1 - p2 - p3],
FAD[Q - p1 - p2],

    FAD[Q - p1], FAD[Q - p2], FAD[p1 + p3]}, {p1, p2, p3}, {Q}, {}, {}]
```

$$\text{FCTopology}\left(\text{topo1}, \left\{\frac{1}{\text{p1}^2}, \frac{1}{\text{p2}^2}, \frac{1}{\text{p3}^2}, \frac{1}{(-\text{p1}-\text{p2}-\text{p3}+Q)^2}, \frac{1}{(-\text{p1}-\text{p2}+Q)^2}, \frac{1}{(Q-\text{p1})^2}, \frac{1}{(Q-\text{p2})^2}, \frac{1}{(\text{p1}+\text{p3})^2}\right\}, \{\text{p1}, \text{p2}, \text{p3}\}, \{Q\}, \{\}, \{\}\right)$$

```
FCLoopValidTopologyQ[topo]
```

True

This topology is missing information about loop and external momenta

```
topoWrong = FCTopology[topo1, {FAD[p1], FAD[p2], FAD[Q - p1 - p2 - p3],
FAD[Q - p1 - p2],

    FAD[Q - p1], FAD[p1 + p3]}, {}, {}]
```

$$\text{FCTopology}\left(\text{topo1}, \left\{\frac{1}{\text{p1}^2}, \frac{1}{\text{p2}^2}, \frac{1}{(-\text{p1}-\text{p2}-\text{p3}+Q)^2}, \frac{1}{(-\text{p1}-\text{p2}+Q)^2}, \frac{1}{(Q-\text{p1})^2}, \frac{1}{(\text{p1}+\text{p3})^2}\right\}, \{\}, \{\}\right)$$

```
FCLoopValidTopologyQ[topoWrong]
```

FCLoopValidTopologyQ: Topology validation failed: FCTopology[topo1,
{FeynAmpDenominator[PropagatorDenominator[Momentum[p1, D], 0]],
FeynAmpDenominator[PropagatorDenominator[Mom ... Q, D], 0]],
FeynAmpDenominator[PropagatorDenominator[Momentum[p1, D] + Momentum[p3, D], 0]]},
{}, {}] is not a proper topology.

False

9.77 FCMultiLoopTID

`FCMultiLoopTID[amp, {q1, q2, ...}]` does a multi-loop tensor integral decomposition, transforming the Lorentz indices away from the loop momenta **`q1, q2, ...`** The decomposition is applied only to the loop integrals where loop momenta are contracted with Dirac matrices or epsilon tensors.

9.77.1 See also

Overview, TID.

9.77.2 Examples

```
FCI[FVD[q1, \[Mu]] FVD[q2, \[Nu]] FAD[q1, q2, {q1 - p1}, {q2 - p1}, {q1 -
q2}]]

FCMultiLoopTID[%, {q1, q2}]
```

$$\frac{\text{q1}^{\mu}\ \text{q2}^{\nu}}{\text{q1}^2.\text{q2}^2.(\text{q1}-\text{p1})^2.(\text{q2}-\text{p1})^2.(\text{q1}-\text{q2})^2}$$

$$\frac{\text{p1}^{\mu}\ \text{p1}^{\nu}-\text{p1}^2 g^{\mu\nu}}{(1-D)\ \text{p1}^2\ \text{q1}^2.\text{q2}^2.(\text{q1}-\text{p1})^2.(\text{q1}-\text{q2})^2}-\frac{\text{p1}^{\mu}\ \text{p1}^{\nu}-\text{p1}^2 g^{\mu\nu}}{2(1-D)\ \text{p1}^2\ \text{q1}^2.\text{q2}^2.(\text{q1}-\text{p1})^2.(\text{q2}-\text{p1})^2}$$
$$-\frac{D\ \text{p1}^{\mu}\ \text{p1}^{\nu}-\text{p1}^2 g^{\mu\nu}}{4(1-D)\ \text{q1}^2.\text{q2}^2.(\text{q1}-\text{p1})^2.(\text{q1}-\text{q2})^2.(\text{q2}-\text{p1})^2}+\frac{D\ \text{p1}^{\mu}\ \text{p1}^{\nu}-\text{p1}^2 g^{\mu\nu}}{2(1-D)\ \text{p1}^4\ \text{q1}^2.(\text{q1}-\text{q2})^2.(\text{q2}-\text{p1})^2}$$

9.78 FeynAmpDenominatorCombine

FeynAmpDenominatorCombine[expr] expands expr with respect to **FeynAmpDenominator** and combines products of **FeynAmpDenominator** in expr into one **FeynAmpDenominator**.

9.78.1 See also

Overview, FeynAmpDenominatorSplit.

9.78.2 Examples

```
FAD[q] FAD[q - p]

ex = FeynAmpDenominatorCombine[%]
```

$$\frac{1}{q^2(q-p)^2}$$

$$\frac{1}{q^2.(q-p)^2}$$

```
ex // FCE // StandardForm

(*FAD[q, -p + q]*)
```

```
ex2 = FeynAmpDenominatorSplit[ex]
```

$$\frac{1}{q^2(q-p)^2}$$

```
ex2 // FCE // StandardForm

(*FAD[q] FAD[-p + q]*)
```

9.79 FeynAmpDenominatorExplicit

FeynAmpDenominatorExplicit[exp] changes each occurrence of **PropagatorDenominator[a, b]** in exp into **1/(SPD[a, a]-b^2)** and replaces **FeynAmpDenominator** by **Identity**.

9.79.1 See also

Overview, FeynAmpDenominator, PropagatorDenominator.

9.79.2 Examples

```
FAD[{q, m}, {q - p, 0}]

FeynAmpDenominatorExplicit[%]

% // FCE // StandardForm
```

$$\frac{1}{(q^2-m^2).(q-p)^2}$$

$$\frac{1}{(q^2-m^2)\,(-2(p\cdot q)+p^2+q^2)}$$

$$\frac{1}{(-m^2+\mathrm{SPD}[q,q])\,(\mathrm{SPD}[p,p]-2\,\mathrm{SPD}[p,q]+\mathrm{SPD}[q,q])}$$

Notice that you should never apply **FeynAmpDenominatorExplicit** to loop integrals. Denominators in a proper loop integral should be written as **FeynAmpDenominator**s. Otherwise, the given integral is assumed to have no denominators and consequently set to zero as being scaleless.

```
TID[FVD[q, mu] FAD[{q, m}, {q - p, 0}], q]
```

$$\frac{\left(m^2+p^2\right) p^{\text{mu}}}{2p^2q^2.\left((q-p)^2-m^2\right)}-\frac{p^{\text{mu}}}{2p^2\left(q^2-m^2\right)}$$

```
TID[FeynAmpDenominatorExplicit[FVD[q, mu] FAD[{q, m}, {q - p, 0}]], q]
```

$$0$$

9.80 FeynAmpDenominatorSimplify

FeynAmpDenominatorSimplify[exp] tries to simplify each **PropagatorDenominator** in a canonical way. **FeynAmpDenominatorSimplify[exp, q1]** simplifies all **FeynAmpDenominator**s in **exp** in a canonical way, including momentum shifts. Scaleless integrals are discarded.

9.80.1 See also

Overview, TID.

9.80.2 Examples

```
FDS
```

FeynAmpDenominatorSimplify

The cornerstone of dimensional regularization is that $\int d^n k\, f(k)/k^4 = 0$

```
FeynAmpDenominatorSimplify[f[k] FAD[k, k], k]
```

$$0$$

This brings some loop integrals into a standard form.

```
FeynAmpDenominatorSimplify[FAD[k - Subscript[p, 1], k - Subscript[p, 2]],
k]
```

$$\frac{1}{k^2.(k-p_1+p_2)^2}$$

```
FeynAmpDenominatorSimplify[FAD[k, k, k - q], k]
```

$$\frac{1}{(k^2)^2.(k-q)^2}$$

```
FeynAmpDenominatorSimplify[f[k] FAD[k, k - q, k - q], k]
```

$$\frac{f(q-k)}{(k^2)^2.(k-q)^2}$$

```
FeynAmpDenominatorSimplify[FAD[k - Subscript[p, 1], k - Subscript[p, 2]]
SPD[k, k], k]

ApartFF[%, {k}]

TID[%, k] // Factor2
```

$$\frac{2\,(k\cdot p_2)+k^2+{p_2}^2}{k^2.(k-p_1+p_2)^2}$$

$$\frac{2\,(k\cdot p_2)+{p_2}^2}{k^2.(k-p_1+p_2)^2}$$

$$\frac{p_1\cdot p_2}{k^2.(k-p_1+p_2)^2}$$

```
FDS[FAD[k - p1, k - p2] SPD[k, OPEDelta]^2, k]
```

$$\frac{(k\cdot\Delta+\Delta\cdot\,\mathrm{p2})^2}{k^2.(k-\mathrm{p1}+\mathrm{p2})^2}$$

9.81 FDS

FDS is shorthand for **FeynAmpDenominatorSimplify**.

9.81.1 See also

Overview, FeynAmpDenominatorSimplify.

9.81.2 Examples

9.82 FeynAmpDenominatorSplit

FeynAmpDenominatorSplit[expr] splits all **FeynAmpDenominator[a, b, ...]** in **expr** into **FeynAmpDenominator[a]*FeynAmpDenominator[b]*...** . **FeynAmpDenominatorSplit[expr, Momentum ->q1]** splits all **FeynAmpDenominator** in expr into two products, one containing **q1** and other momenta, the second being free of **q1**.

9.82.1 See also

Overview, FeynAmpDenominatorCombine.

9.82.2 Examples

```
FAD[q1, q1 - p, q1 - q2, q2, q2 - p]

ex = FeynAmpDenominatorSplit[%]
```

$$\frac{1}{\text{q1}^2.(\text{q1}-p)^2.(\text{q1}-\text{q2})^2.\text{q2}^2.(\text{q2}-p)^2}$$

$$\frac{1}{\text{q1}^2\ \text{q2}^2(\text{q1}-p)^2(\text{q2}-p)^2(\text{q1}-\text{q2})^2}$$

```
ex // FCE // StandardForm

(*FAD[q1] FAD[-p + q1] FAD[q1 - q2] FAD[q2] FAD[-p + q2]*)
```

```
ex = FeynAmpDenominatorSplit[FAD[q1, q1 - p, q1 - q2, q2, q2 - p], Momentum
-> {q1}]
```

$$\frac{1}{\text{q2}^2.(\text{q2}-p)^2\ \text{q1}^2.(\text{q1}-p)^2.(\text{q1}-\text{q2})^2}$$

```
ex // FCE // StandardForm

(*FAD[q2, -p + q2] FAD[q1, -p + q1, q1 - q2]*)
```

```
FeynAmpDenominatorCombine[ex] // FCE // StandardForm

(*FAD[q1, q2, q1 - q2, -p + q1, -p + q2]*)
```

9.83 FromGFAD

FromGFAD[exp] converts all suitable generic propagator denominators into standard and Cartesian propagator denominators.

The options **InitialSubstitutions** and **IntermediateSubstitutions** can be used to help the function handle nontrivial propagators.

For propagators containing symbolic variables it might be necessary to tell the function that those are larger than zero (if applicable), so that expressions such as $\sqrt{\lambda^2}$ can be simplified accordingly.

9.83.1 See also

Overview, GFAD, SFAD, CFAD, FeynAmpDenominatorExplicit.

9.83.2 Examples

```
GFAD[SPD[p1]]

ex = FromGFAD[%]
```

$$\frac{1}{(\mathrm{p1}^2 + i\eta)}$$

$$\frac{1}{(\mathrm{p1}^2 + i\eta)}$$

```
ex // StandardForm

(*FeynAmpDenominator[StandardPropagatorDenominator[Momentum[p1, D], 0, 0,
{1, 1}]]*)
```

```
GFAD[SPD[p1] + 2 SPD[p1, p2]]

ex = FromGFAD[%]
```

$$\frac{1}{(\text{p1}^2 + 2(\text{p1} \cdot \text{p2}) + i\eta)}$$

$$\frac{1}{(\text{p1}^2 + 2(\text{p1} \cdot \text{p2}) + i\eta)}$$

```
ex // StandardForm

(*FeynAmpDenominator[StandardPropagatorDenominator[Momentum[p1, D], 2
Pair[Momentum[p1, D], Momentum[p2, D]], 0, {1, 1}]]*)
```

```
GFAD[{{CSPD[p1] + 2 CSPD[p1, p2] + m^2, -1}, 2}]

ex = FromGFAD[%]
```

$$\frac{1}{(m^2 + \text{p1}^2 + 2(\text{p1} \cdot \text{p2}) - i\eta)^2}$$

$$\frac{1}{(\text{p1}^2 + 2(\text{p1} \cdot \text{p2}) + m^2 - i\eta)^2}$$

```
ex // StandardForm

(*FeynAmpDenominator[CartesianPropagatorDenominator[CartesianMomentum[p1,
-1 + D], 2 CartesianPair[CartesianMomentum[p1, -1 + D],
CartesianMomentum[p2, -1 + D]], m^2, {2, -1}]]*)
```

```
prop = FeynAmpDenominator[GenericPropagatorDenominator[-la
Pair[Momentum[p1, D],
       Momentum[p1, D]] + 2 Pair[Momentum[p1, D], Momentum[q, D]], {1,1}]]
```

$$\frac{1}{(2(\text{p1} \cdot q) - \text{la}\ \text{p1}^2 + i\eta)}$$

```
ex = FromGFAD[prop]
```

$$\frac{1}{(-\text{la}\ \text{p1}^2 + 2(\text{p1} \cdot q) + i\eta)}$$

```
ex // StandardForm
```

$$\text{FeynAmpDenominator}\left[\text{StandardPropagatorDenominator}\left[\sqrt{-\text{la}}\,\text{Momentum}[\text{p1}, D], 2\,\text{Pair}[\text{Momentum}[\text{p1}, D], \text{Momentum}[q, D]], 0, \{1, 1\}\right]\right]$$

```
ex = FromGFAD[prop, PowerExpand -> {la}]
```

$$\frac{1}{(-\text{la}\,\text{p1}^2 + 2(\text{p1}\cdot q) + i\eta)}$$

```
ex // StandardForm
```

$$\text{FeynAmpDenominator}\left[\text{StandardPropagatorDenominator}\left[i\sqrt{\text{la}}\,\text{Momentum}[\text{p1}, D], 2\,\text{Pair}[\text{Momentum}[\text{p1}, D], \text{Momentum}[q, D]], 0, \{1, 1\}\right]\right]$$

```
ex = GFAD[{{-SPD[p1, p1], 1}, 1}]*GFAD[{{SPD[p1, -p1 + 2*p3] - SPD[p3, p3],
1}, 1}]*
    GFAD[{{-SPD[p3, p3], 1}, 1}]*SFAD[{{I*(p1 + q), 0}, {-mb^2, 1}, 1}]*
    SFAD[{{I*(p3 + q), 0}, {-mb^2, 1}, 1}] +  (-2*mg^2*GFAD[{{-SPD[p1, p1],
1}, 2}]*
       GFAD[{{SPD[p1, -p1 + 2*p3] - SPD[p3, p3], 1}, 1}]*GFAD[{{-SPD[p3,
p3], 1}, 1}]*
       SFAD[{{I*(p1 + q), 0}, {-mb^2, 1}, 1}]*SFAD[{{I*(p3 + q), 0},
{-mb^2, 1}, 1}] -
        2*mg^2*GFAD[{{-SPD[p1, p1], 1}, 1}]*GFAD[{{SPD[p1, -p1 + 2*p3] -
        SPD[p3, p3],
            1}, 2}]*GFAD[{{-SPD[p3, p3], 1}, 1}]*SFAD[{{I*(p1 + q), 0},
            {-mb^2, 1}, 1}]*
         SFAD[{{I*(p3 + q), 0}, {-mb^2, 1}, 1}] -    2*mg^2*GFAD[{{-SPD[p1,
p1], 1},
           1}]*GFAD[{{SPD[p1, -p1 + 2*p3] - SPD[p3, p3], 1},
           1}]*GFAD[{{-SPD[p3, p3],
            1}, 2}]*SFAD[{{I*(p1 + q), 0}, {-mb^2, 1}, 1}]*SFAD[{{I*(p3 +
            q), 0},
           {-mb^2, 1}, 1}])/2
```

$$\frac{1}{2}\left(-\frac{2\,\mathrm{mg}^2}{(-\mathrm{p1}^2+i\eta)^2(-\mathrm{p3}^2+i\eta)(-(\mathrm{p1}+q)^2+\mathrm{mb}^2+i\eta)(-(\mathrm{p3}+q)^2+\mathrm{mb}^2+i\eta)(\mathrm{p1}\cdot(2\,\mathrm{p3}-\mathrm{p1})-\mathrm{p3}^2+i\eta)}\right.$$
$$-\frac{2\,\mathrm{mg}^2}{(-\mathrm{p1}^2+i\eta)(-\mathrm{p3}^2+i\eta)(-(\mathrm{p1}+q)^2+\mathrm{mb}^2+i\eta)(-(\mathrm{p3}+q)^2+\mathrm{mb}^2+i\eta)(\mathrm{p1}\cdot(2\,\mathrm{p3}-\mathrm{p1})-\mathrm{p3}^2+i\eta)^2}$$
$$\left.-\frac{2\,\mathrm{mg}^2}{(-\mathrm{p1}^2+i\eta)(-\mathrm{p3}^2+i\eta)^2(-(\mathrm{p1}+q)^2+\mathrm{mb}^2+i\eta)(-(\mathrm{p3}+q)^2+\mathrm{mb}^2+i\eta)(\mathrm{p1}\cdot(2\,\mathrm{p3}-\mathrm{p1})-\mathrm{p3}^2+i\eta)}\right)$$
$$+\frac{1}{(-\mathrm{p1}^2+i\eta)(-\mathrm{p3}^2+i\eta)(-(\mathrm{p1}+q)^2+\mathrm{mb}^2+i\eta)(-(\mathrm{p3}+q)^2+\mathrm{mb}^2+i\eta)(\mathrm{p1}\cdot(2\,\mathrm{p3}-\mathrm{p1})-\mathrm{p3}^2+i\eta)}$$

```
FromGFAD[ex]
```

$$\frac{1}{2}\left(-\frac{2\,\mathrm{mg}^2}{(-\mathrm{p1}^2+i\eta)^2(-\mathrm{p3}^2+i\eta)(-(\mathrm{p1}+q)^2+\mathrm{mb}^2+i\eta)(-(\mathrm{p3}+q)^2+\mathrm{mb}^2+i\eta)(-\mathrm{p3}^2+\mathrm{p1}\cdot(2\,\mathrm{p3}-\mathrm{p1})+i\eta)}\right.$$
$$-\frac{2\,\mathrm{mg}^2}{(-\mathrm{p1}^2+i\eta)(-\mathrm{p3}^2+i\eta)^2(-(\mathrm{p1}+q)^2+\mathrm{mb}^2+i\eta)(-(\mathrm{p3}+q)^2+\mathrm{mb}^2+i\eta)(-\mathrm{p3}^2+\mathrm{p1}\cdot(2\,\mathrm{p3}-\mathrm{p1})+i\eta)}$$
$$\left.-\frac{2\,\mathrm{mg}^2}{(-\mathrm{p1}^2+i\eta)(-\mathrm{p3}^2+i\eta)(-(\mathrm{p1}+q)^2+\mathrm{mb}^2+i\eta)(-(\mathrm{p3}+q)^2+\mathrm{mb}^2+i\eta)(-\mathrm{p3}^2+\mathrm{p1}\cdot(2\,\mathrm{p3}-\mathrm{p1})+i\eta)^2}\right)$$
$$+\frac{1}{(-\mathrm{p1}^2+i\eta)(-\mathrm{p3}^2+i\eta)(-(\mathrm{p1}+q)^2+\mathrm{mb}^2+i\eta)(-(\mathrm{p3}+q)^2+\mathrm{mb}^2+i\eta)(-\mathrm{p3}^2+\mathrm{p1}\cdot(2\,\mathrm{p3}-\mathrm{p1})+i\eta)}$$

9.84 GammaExpand

`GammaExpand[exp]` rewrites **`Gamma[n + m]`** in **`exp`** (where **`n`** has **`Head Integer`**).

9.84.1 See also

Overview, GammaEpsilon.

9.84.2 Examples

```
GammaExpand[Gamma[2 + Epsilon]]
```

$$(\varepsilon+1)\Gamma(\varepsilon+1)$$

```
GammaExpand[Gamma[-3 + Epsilon]]
```

$$\frac{\Gamma(\varepsilon+1)}{(\varepsilon-3)(\varepsilon-2)(\varepsilon-1)\varepsilon}$$

```
GammaExpand[Gamma[1 + Epsilon]]
```

$$\Gamma(\varepsilon+1)$$

9.85 GenPaVe

`GenPaVe[i, j, ..., {{0, m0}, {Momentum[p1], m1}, {Momentum[p2], m2}, ...]` denotes the invariant (and scalar) Passarino-Veltman integrals, i.e. the coefficient functions of the tensor integral decomposition. In contrast to **`PaVe`** which uses the LoopTools convention, masses and external momenta in **`GenPaVe`** are written in the same order as they appear in the original tensor integral, i.e. **`FAD[{q, m0}, {q-p1, m1}, {q-p2, m2}, ...]`**.

9.85.1 See also

Overview, PaVe.

9.85.2 Examples

```
FVD[q, \[Mu]] FVD[q, \[Nu]] FAD[{q, m0}, {q + p1, m1}, {q + p2,
m2}]/(I*Pi^2)

TID[%, q, UsePaVeBasis -> True]

TID[%%, q, UsePaVeBasis -> True, GenPaVe -> True]
```

$$-\frac{iq^{\mu}q^{\nu}}{\pi^2\left(q^2-\text{m0}^2\right).\left((\text{p1}+q)^2-\text{m1}^2\right).\left((\text{p2}+q)^2-\text{m2}^2\right)}$$

$$g^{\mu\nu}\,\text{C}_{00}\left(\text{p1}^2,\text{p2}^2,-2(\text{p1}\cdot\ \text{p2})+\text{p1}^2+\text{p2}^2,\text{m1}^2,\text{m0}^2,\text{m2}^2\right)+\text{p1}^{\mu}\,\text{p1}^{\nu}\,\text{C}_{11}\left(\text{p1}^2,\right.$$
$$-2(\text{p1}\cdot\ \text{p2})+\text{p1}^2+\text{p2}^2,\text{p2}^2,\text{m0}^2,\text{m1}^2,\text{m2}^2\left.\right)+\text{p2}^{\mu}\,\text{p2}^{\nu}\,\text{C}_{11}\left(\text{p2}^2,-2(\text{p1}\cdot\ \text{p2})+\text{p1}^2+\text{p2}^2,\text{p1}^2,\right.$$
$$\left.\text{m0}^2,\text{m2}^2,\text{m1}^2\right)+\left(\text{p1}^{\nu}\,\text{p2}^{\mu}+\text{p1}^{\mu}\,\text{p2}^{\nu}\right)\,\text{C}_{12}\left(\text{p1}^2,-2(\text{p1}\cdot\ \text{p2})+\text{p1}^2+\text{p2}^2,\text{p2}^2,\text{m0}^2,\text{m1}^2,\text{m2}^2\right)$$

$$g^{\mu\nu}\,\text{GenPaVe}\left(\{0,0\},\left(\begin{array}{cc}0&\text{m0}\\\text{p1}&\text{m1}\\\text{p2}&\text{m2}\end{array}\right)\right)+\text{p1}^{\mu}\,\text{p1}^{\nu}\,\text{GenPaVe}\left(\{1,\right.$$
$$\left.1\},\left(\begin{array}{cc}0&\text{m0}\\\text{p1}&\text{m1}\\\text{p2}&\text{m2}\end{array}\right)\right)+\text{p2}^{\mu}\,\text{p2}^{\nu}\,\text{GenPaVe}\left(\{2,2\},\right.$$
$$\left.\left(\begin{array}{cc}0&\text{m0}\\\text{p1}&\text{m1}\\\text{p2}&\text{m2}\end{array}\right)\right)+\left(\text{p1}^{\nu}\,\text{p2}^{\mu}+\text{p1}^{\mu}\,\text{p2}^{\nu}\right)\,\text{GenPaVe}\left(\{1,2\},\left(\begin{array}{cc}0&\text{m0}\\\text{p1}&\text{m1}\\\text{p2}&\text{m2}\end{array}\right)\right)$$

9.86 PaVe

`PaVe[i, j, ..., {p10, p12, ...}, {m1^2, mw^2, ...}]` denotes the invariant (and scalar) Passarino-Veltman integrals, i.e. the coefficient functions of the tensor integral decomposition. Joining **`plist`** and **`mlist`** gives the same conventions as for **`A0`**, **`B0`**, **`C0`**, **`D0`**. Automatic simplifications are performed for the coefficient functions of two-point integrals and for the scalar integrals.

9.86.1 See also

Overview, PaVeReduce.

9.86.2 Examples

Some of the PaVe's reduce to special cases with **`PaVeAutoReduce`**to **`True`**

```
PaVe[0, 0, {pp}, {m^2, M^2}, PaVeAutoReduce -> True]
```

$$\frac{\left(m^2-2mM+M^2-\text{pp}\right)\left(m^2+2mM+M^2-\text{pp}\right)\ \text{B}_0\left(\text{pp},m^2,M^2\right)}{4(1-D)\ \text{pp}}$$
$$-\frac{\text{A}_0\left(m^2\right)\left(m^2-M^2+\text{pp}\right)}{4(1-D)\ \text{pp}}+\frac{\text{A}_0\left(M^2\right)\left(m^2-M^2-\text{pp}\right)}{4(1-D)\ \text{pp}}$$

9.87 GLI

`GLI[id, {indices}]` is a generic loop integral, where the indices denote powers of propagators in the propagator basis (**`FCTopology`**) named **`id`**.

9.87.1 See also

Overview, FCTopology.

9.87.2 Examples

```
GLI["topo1", {1, 0, 0, 1, 1}]
```

$$G^{\text{topo1}}(1,0,0,1,1)$$

9.88 GLIMultiply

`GLIMultiply` is like **`GLI`** but with local multiplication properties.

9.88.1 See also

Overview, GLI.

9.88.2 Examples

```
GLI["topo1", {1, 0, 0, 1, 1}] GLI["topo1", {0, -1, -1, 0, 0}]

% /. GLI -> GLIMultiply /. GLIMultiply -> GLI
```

$$G^{\text{topo1}}(0,-1,-1,0,0)G^{\text{topo1}}(1,0,0,1,1)$$

$$G^{\text{topo1}}(1,-1,-1,1,1)$$

9.89 Hill

`Hill[x, y]` gives the Hill identity with arguments **`x`** and **`y`**. The returned object is **`0`**.

9.89.1 See also

Overview, SimplifyPolyLog.

9.89.2 Examples

```
Hill[a, b]

% /. a :> .123 /. b :> .656 // Chop
```

$$\begin{aligned}&\text{Li}_2\left(\frac{1-a}{1-b}\right)+\text{Li}_2\left(\frac{b}{a}\right)-\text{Li}_2\left(\frac{(1-a)b}{a(1-b)}\right)+\log(a)(\log(1-a)-\log(1-b))\\&+\log\left(\frac{1-a}{1-b}\right)\left(-\log\left(\frac{a-b}{a}\right)+\log\left(\frac{a-b}{1-b}\right)-\log(a)+\log(1-b)\right)\\&-\left(-\log\left(\frac{a-b}{a}\right)+\log\left(\frac{a-b}{a(1-b)}\right)+\log(1-b)\right)\log\left(\frac{(1-a)b}{a(1-b)}\right)+\text{Li}_2(a)-\text{Li}_2(b)-\frac{\pi^2}{6}\end{aligned}$$

$$0$$

```
Hill[x, x y] // PowerExpand // SimplifyPolyLog // Expand

% /. x :> .34 /. y -> .6 // N // Chop
```

$$\zeta(2) - \text{Li}_2(xy) + \text{Li}_2\left(\frac{1-x}{1-xy}\right) - \text{Li}_2\left(\frac{(1-x)y}{1-xy}\right) - \text{Li}_2(1-x) \\ - \text{Li}_2(1-y) - \log(x)\log(1-xy) - \log(1-y)\log(y)$$

$$0$$

9.90 HypergeometricAC

`HypergeometricAC[n][exp]` analytically continues **`Hypergeometric2F1`** functions in **`exp`**. The second argument **`n`** refers to the equation number (n) in chapter 2.10 of "Higher Transcendental Functions" by Erdelyi, Magnus, Oberhettinger, Tricomi. In case of eq. (6) (p.109) the last line is returned for **`HypergeometricAC[6][exp]`**, while the first equality is given by **`HypergeometricAC[61][exp]`**.

(2.10.1) is identical to eq. (9.5.7) of "Special Functions & their Applications" by N.N.Lebedev.

9.90.1 See also

Overview, HypExplicit, HypergeometricIR, HypergeometricSE, ToHypergeometric.

9.90.2 Examples

These are all transformation rules currently built in.

```
HypergeometricAC[1][Hypergeometric2F1[\[Alpha], \[Beta], \[Gamma], z]]
```

$$\frac{\Gamma(\gamma)\Gamma(\alpha+\beta-\gamma)(1-z)^{-\alpha-\beta+\gamma}\,{}_2F_1(\gamma-\alpha,\gamma-\beta;-\alpha-\beta+\gamma+1;1-z)}{\Gamma(\alpha)\Gamma(\beta)} \\ + \frac{\Gamma(\gamma)\Gamma(-\alpha-\beta+\gamma)\,{}_2F_1(\alpha,\beta;\alpha+\beta-\gamma+1;1-z)}{\Gamma(\gamma-\alpha)\Gamma(\gamma-\beta)}$$

```
HypergeometricAC[2][Hypergeometric2F1[\[Alpha], \[Beta], \[Gamma], z]]
```

$$\frac{\Gamma(\gamma)(-z)^{-\alpha}\Gamma(\beta-\alpha)\,{}_2F_1\left(\alpha,\alpha-\gamma+1;\alpha-\beta+1;\frac{1}{z}\right)}{\Gamma(\beta)\Gamma(\gamma-\alpha)} + \frac{\Gamma(\gamma)(-z)^{-\beta}\Gamma(\alpha-\beta)\,{}_2F_1\left(\beta,\beta-\gamma+1;-\alpha+\beta+1;\frac{1}{z}\right)}{\Gamma(\alpha)\Gamma(\gamma-\beta)}$$

```
HypergeometricAC[3][Hypergeometric2F1[\[Alpha], \[Beta], \[Gamma], z]]
```

$$\frac{\Gamma(\gamma)(1-z)^{-\alpha}\Gamma(\beta-\alpha)\,{}_2F_1\left(\alpha,\gamma-\beta;\alpha-\beta+1;\frac{1}{1-z}\right)}{\Gamma(\beta)\Gamma(\gamma-\alpha)}+\frac{\Gamma(\gamma)(1-z)^{-\beta}\Gamma(\alpha-\beta)\,{}_2F_1\left(\beta,\gamma-\alpha;-\alpha+\beta+1;\frac{1}{1-z}\right)}{\Gamma(\alpha)\Gamma(\gamma-\beta)}$$

```
HypergeometricAC[4][Hypergeometric2F1[\[Alpha], \[Beta], \[Gamma], z]]
```

$$\frac{\Gamma(\gamma)z^{-\alpha}\Gamma(-\alpha-\beta+\gamma)\,{}_2F_1\left(\alpha,\alpha-\gamma+1;\alpha+\beta-\gamma+1;-\frac{1-z}{z}\right)}{\Gamma(\gamma-\alpha)\Gamma(\gamma-\beta)}$$
$$+\frac{\Gamma(\gamma)z^{\alpha-\gamma}\Gamma(\alpha+\beta-\gamma)(1-z)^{-\alpha-\beta+\gamma}\,{}_2F_1\left(1-\alpha,\gamma-\alpha;-\alpha-\beta+\gamma+1;-\frac{1-z}{z}\right)}{\Gamma(\alpha)\Gamma(\beta)}$$

```
HypergeometricAC[6][Hypergeometric2F1[\[Alpha], \[Beta], \[Gamma], z]]
```

$$(1-z)^{-\beta}\,{}_2F_1\left(\beta,\gamma-\alpha;\gamma;-\frac{z}{1-z}\right)$$

```
HypergeometricAC[61][Hypergeometric2F1[\[Alpha], \[Beta], \[Gamma], z]]
```

$$(1-z)^{-\alpha}\,{}_2F_1\left(\alpha,\gamma-\beta;\gamma;-\frac{z}{1-z}\right)$$

9.91 HypergeometricIR

`HypergeometricIR[exp, t]` substitutes for all **`Hypergeometric2F1[a, b, c, z]`** in **`exp`** by its Euler integral representation. The factor **`Integratedx[t, 0, 1]`** can be omitted by setting the option **`Integratedx -> False`**.

9.91.1 See also

Overview, HypergeometricAC, HypergeometricSE, ToHypergeometric.

9.91.2 Examples

```
HypergeometricIR[Hypergeometric2F1[a, b, c, z], t]
```

$$\frac{t^{b-1}\Gamma(c)(1-tz)^{-a}(1-t)^{-b+c-1}}{\Gamma(b)\Gamma(c-b)}$$

```
ToHypergeometric[t^b (1 - t)^c (1 + t z)^a, t]

HypergeometricIR[%, t]
```

$$\frac{\Gamma(b+1)\Gamma(c+1)\,{}_2F_1(-a,b+1;b+c+2;-z)}{\Gamma(b+c+2)}$$

$$t^b(1-t)^c(tz+1)^a$$

9.92 HypergeometricSE

HypergeometricSE[exp, nu] expresses Hypergeometric functions by their series expansion in terms of a sum (the **Sum** is omitted and **nu**, running from 0 to ∞, is the summation index).

9.92.1 See also

Overview, HypergeometricIR.

9.92.2 Examples

```
HypergeometricSE[Hypergeometric2F1[a, b, c, z], \[Nu]]
```

$$\frac{\Gamma(c)z^\nu\Gamma(a+\nu)\Gamma(b+\nu)}{\Gamma(a)\Gamma(b)\Gamma(\nu+1)\Gamma(c+\nu)}$$

```
HypergeometricSE[HypergeometricPFQ[{a, b, c}, {d, e}, z], \[Nu]]
```

$$\frac{\Gamma(d)\Gamma(e)z^\nu\Gamma(a+\nu)\Gamma(b+\nu)\Gamma(c+\nu)}{\Gamma(a)\Gamma(b)\Gamma(c)\Gamma(\nu+1)\Gamma(d+\nu)\Gamma(e+\nu)}$$

9.93 HypExplicit

HypExplicit[exp, nu] expresses Hypergeometric functions in exp by their definition in terms of a sum (the **Sum** is omitted and **nu** is the summation index).

9.93.1 See also

Overview, HypergeometricIR.

9.93.2 Examples

```
Hypergeometric2F1[a, b, c, z]

HypExplicit[%, \[Nu]]
```

$$ {}_2F_1(a, b; c; z) $$

$$ \frac{\Gamma(c) z^{\nu} \Gamma(a+\nu) \Gamma(b+\nu)}{\Gamma(a) \Gamma(b) \Gamma(\nu+1) \Gamma(c+\nu)} $$

```
HypergeometricPFQ[{a, b, c}, {d, e}, z]

HypExplicit[%, \[Nu]]
```

$$ {}_3F_2(a, b, c; d, e; z) $$

$$ \frac{\Gamma(d) \Gamma(e) z^{\nu} \Gamma(a+\nu) \Gamma(b+\nu) \Gamma(c+\nu)}{\Gamma(a) \Gamma(b) \Gamma(c) \Gamma(\nu+1) \Gamma(d+\nu) \Gamma(e+\nu)} $$

9.94 HypInt

`HypInt[exp, t]` substitutes all **`Hypergeometric2F1[a, b, c, z]`** in **`exp`** with **`Gamma[c]/(Gamma[b] Gamma[c-b]) Integratedx[t, 0, 1] t^(b-1) (1-t)^(c-b-1) (1-t z)^(-a)`**.

9.94.1 See also

Overview, Series2.

9.94.2 Examples

```
Hypergeometric2F1[a, b, c, z]

HypInt[%, t]
```

$$ {}_2F_1(a,b;c;z) $$

$$ \frac{t^{b-1}\Gamma(c)(1-tz)^{-a}(1-t)^{-b+c-1}\int_0^1 dt}{\Gamma(b)\Gamma(c-b)} $$

9.95 IntegrateByParts

`IntegrateByParts[(1 - t)^(a Epsilon -1)g[t], deriv, t]` does an integration by parts of the definite integral over **`t`** from **`0`** to **`1`**.

9.95.1 See also

Overview, PartialIntegrate, Integrate2.

9.95.2 Examples

```
IntegrateByParts[(1 - t)^(a Epsilon - 1) g[t], (1 - t)^(a Epsilon - 1), t]
```

$$ \frac{(1-t)^{a\varepsilon}g'(t)}{a\varepsilon} + \frac{g(0)}{a\varepsilon} $$

9.96 PartialIntegrate

`PartialIntegrate[exp, ap, t]` does a partial integration of the definite integral **`Integrate[exp, {t, 0, 1}]`**, with **`ap`** the factor that is to be integrated and **`exp/ap`** the factor that is to be differentiated.

9.96.1 See also

Overview, IntegrateByParts, Integrate2.

9.96.2 Examples

```
PartialIntegrate[f[x] g[x], g[x], {x, 0, 1}]
```

$$-(f(x)\int g(x)\,dx/.x\to 0)+(f(x)\int g(x)\,dx/.x\to 1)-\int_0^1 f'(x)(\int g(x)\,dx)\,dx$$

```
f[x_] = Integrate[Log[3 x + 2], x]

g[x_] = D[1/Log[3 x + 2], x]
```

$$\left(x+\frac{2}{3}\right)\log(3x+2)-x$$

$$-\frac{3}{(3x+2)\log^2(3x+2)}$$

```
Integrate[PartialIntegrate[f[x] g[x], f[x], x], {x, 0, 1}] // FullSimplify
```

$$-\frac{1}{\log(5)}$$

```
Integrate[f[x] g[x], {x, 0, 1}] // Simplify
```

$$-\frac{1}{\log(5)}$$

```
Clear[f, g]
```

9.97 NPointTo4Point

`NPointTo4Point[expr, q]` reduces scalar IR finite 5-point functions to scalar 4-point functions according to Eq. 4.52 in arXiv:0709.1075.

9.97.1 See also

Overview, PaVeReduce.

9.97.2 Examples

```
FCClearScalarProducts[]

SPD[p1] = 0;

SPD[p1, p4] = 0;

SPD[p3, p4] = 0;

SPD[p1, p2] = 0;

SPD[p2, p4] = 0;

int = FCI[FAD[{q, m0}, {q + p1, 0}, {q + p2, 0}, {q + p3, 0}, {q + p4, 0}]]
```

$$\frac{1}{\left(q^2-\text{m0}^2\right).(\text{p1}+q)^2.(\text{p2}+q)^2.(\text{p3}+q)^2.(\text{p4}+q)^2}$$

```
NPointTo4Point[int, q, FCE -> True, FCVerbose -> -1]

FCClearScalarProducts[]
```

$$\left(\frac{8\,\text{m0}^2\,\text{p2}^2\,\text{p4}^2(\text{p1}\cdot\,\text{p3})}{\left(q^2-\text{m0}^2\right).(\text{p1}+q)^2.(\text{p2}+q)^2.(\text{p4}+q)^2}\right.$$
$$+\frac{8\,\text{p4}^2(\text{p1}\cdot\,\text{p3})\left(\text{m0}^2(\text{p1}\cdot\,\text{p3})-\text{m0}^2(\text{p2}\cdot\,\text{p3})+\text{p2}^2(\text{p1}\cdot\,\text{p3})\right)}{\left(q^2-\text{m0}^2\right).(\text{p1}+q)^2.(\text{p3}+q)^2.(\text{p4}+q)^2}$$
$$+\frac{8\,\text{p4}^2\left(\text{m0}^2\,\text{p2}^2(\text{p1}\cdot\,\text{p3})-\text{m0}^2(\text{p1}\cdot\,\text{p3})(\text{p2}\cdot\,\text{p3})-\text{m0}^2\,\text{p2}^2\,\text{p3}^2+\text{m0}^2(\text{p2}\cdot\,\text{p3})^2-\text{p2}^2(\text{p1}\cdot\,\text{p3})(\text{p2}\cdot\,\text{p3})+\text{p2}^2\,\text{p}\right.}{\left(q^2-\text{m0}^2\right).(\text{p2}+q)^2.(\text{p3}+q)^2.(\text{p4}+q)^2}$$
$$+\frac{8\,\text{p2}^2\left(\text{m0}^2+\text{p4}^2\right)(\text{p1}\cdot\,\text{p3})^2}{\left(q^2-\text{m0}^2\right).(\text{p1}+q)^2.(\text{p2}+q)^2.(\text{p3}+q)^2}$$
$$-\left(8\left(2\,\text{m0}^2\,\text{p2}^2\,\text{p4}^2(\text{p1}\cdot\text{p3})-2\,\text{m0}^2\,\text{p4}^2(\text{p1}\cdot\text{p3})(\text{p2}\cdot\text{p3})+\text{m0}^2\,\text{p2}^2(\text{p1}\cdot\text{p3})^2+\text{m0}^2\,\text{p4}^2(\text{p1}\cdot\text{p3})^2-\text{m0}^2\,\text{p2}^2\,\text{p3}^2\,\text{p4}^2+\text{m0}\right.\right.$$

9.98 OneLoopSimplify

`OneLoopSimplify[amp, q]` simplifies the one-loop amplitude amp. The second argument denotes the integration momentum.

If the first argument has head **`FeynAmp`** then **`OneLoopSimplify[FeynAmp[name, k, expr], k]`** transforms to **`OneLoopSimplify[expr, k]`**

9.98.1 See also

Overview, TID, TIDL.

9.98.2 Examples

```
SPD[k, r] FAD[{k, m} , k - p] // FCI

OneLoopSimplify[%, k]

OneLoopSimplify[% /. m -> 0, k]
```

$$\frac{k \cdot r}{\left(k^2 - m^2\right).(k-p)^2}$$

$$\frac{\left(m^2 + p^2\right)(p \cdot r)}{2p^2k^2.\left((k-p)^2 - m^2\right)} - \frac{p \cdot r}{2p^2\left(k^2 - m^2\right)}$$

$$\frac{p \cdot r}{2k^2.(k-p)^2}$$

```
FAD[k, k, k - Subscript[p, 1], k - Subscript[p, 2]] FVD[k, \[Mu]] // FCI

OneLoopSimplify[ %, k]

FCE[%] /. SPD[Subscript[p, 1]] -> 0 // FCI
```

$$\frac{k^\mu}{\left(k^2\right)^2.(k-p_1)^2.(k-p_2)^2}$$

$$\begin{aligned}
&\frac{{p_1}^2 p_2^\mu - p_1^\mu\,(p_1 \cdot p_2)}{2\left(k^2\right)^2.(k-p_1)^2\left((p_1 \cdot p_2)^2 - {p_1}^2{p_2}^2\right)} \\
&+ \frac{-{p_1}^2{p_2}^2 p_1^\mu - {p_1}^2{p_2}^2 p_2^\mu + {p_1}^2 p_2^\mu\,(p_1 \cdot p_2) + {p_2}^2 p_1^\mu\,(p_1 \cdot p_2)}{2\left(k^2\right)^2.(k-p_1)^2.(k-p_2)^2\left((p_1 \cdot p_2)^2 - {p_1}^2{p_2}^2\right)} \\
&- \frac{p_2^\mu\,(p_1 \cdot p_2) - {p_2}^2 p_1^\mu}{2\left(k^2\right)^2.(k-p_2)^2\left((p_1 \cdot p_2)^2 - {p_1}^2{p_2}^2\right)} - \frac{{p_1}^2 p_2^\mu + {p_2}^2 p_1^\mu - p_1^\mu\,(p_1 \cdot p_2) - p_2^\mu\,(p_1 \cdot p_2)}{2k^2.(k-p_1)^2.(k-p_2)^2\left((p_1 \cdot p_2)^2 - {p_1}^2{p_2}^2\right)}
\end{aligned}$$

$$\begin{aligned}
&\frac{{p_2}^2 p_1^\mu}{2(p_1 \cdot p_2)\left(k^2\right)^2.(k-p_1)^2.(k-p_2)^2} - \frac{p_1^\mu}{2(p_1 \cdot p_2)\left(k^2\right)^2.(k-p_1)^2} \\
&- \frac{p_2^\mu\,(p_1 \cdot p_2) - {p_2}^2 p_1^\mu}{2\left(k^2\right)^2.(k-p_2)^2(p_1 \cdot p_2)^2} - \frac{{p_2}^2 p_1^\mu - p_1^\mu\,(p_1 \cdot p_2) - p_2^\mu\,(p_1 \cdot p_2)}{2k^2.(k-p_1)^2.(k-p_2)^2(p_1 \cdot p_2)^2}
\end{aligned}$$

```
OneLoopSimplify[FAD[k - Subscript[p, 1], k - Subscript[p, 2]] SPD[k, l]^2,
k]
```

$$-\frac{D(l\cdot p_1)^2+D(l\cdot p_2)^2+2D\,(l\cdot p_2)\,(l\cdot p_1)-l^2{p_1}^2-l^2{p_2}^2-4\,(l\cdot p_2)\,(l\cdot p_1)+2l^2\,(p_1\cdot p_2)}{4(1-D)k^2.(k-p_1+p_2)^2}$$

9.99 ToHypergeometric

`ToHypergeometric[t^b (1 - t)^c (1+tz)^a, t]` returns **`u^a Gamma[b+1] Gamma[c+1]/Gamma[ b+c+2] Hypergeometric2F1[-a, b+1, b+c+2, -z/u]`**. Remember that $\mathrm{Re}(b) > 0$ and $\mathrm{Re}(c-b) > 0$ should hold (need not be set in Mathematica).

9.99.1 See also

Overview, HypergeometricAC, HypergeometricIR, HypergeometricSE.

9.99.2 Examples

```
ToHypergeometric[t^b (1 - t)^c (1 + t z)^a, t]
```

$$\frac{\Gamma(b+1)\Gamma(c+1)\,{}_2F_1(-a,b+1;b+c+2;-z)}{\Gamma(b+c+2)}$$

```
ToHypergeometric[w t^(b - 1) (1 - t)^(c - b - 1) (1 - t z)^-a, t]
```

$$\frac{w\Gamma(b)\Gamma(c-b)\,{}_2F_1(a,b;c;z)}{\Gamma(c)}$$

```
ToHypergeometric[t^b (1 - t)^c (u + t z)^a, t]
```

$$\frac{u^a\Gamma(b+1)\Gamma(c+1)\,{}_2F_1\left(-a,b+1;b+c+2;-\frac{z}{u}\right)}{\Gamma(b+c+2)}$$

```
ToHypergeometric[w t^(b - 1) (1 - t)^(c - b - 1) (u - t z)^-a, t]
```

$$\frac{wu^{-a}\Gamma(b)\Gamma(c-b)\,{}_2F_1\left(a,b;c;\frac{z}{u}\right)}{\Gamma(c)}$$

9.100 PaVeToABCD

PaVeToABCD[expr] converts suitable PaVe functions to direct Passarino-Veltman functions (**A0**, **A00**, **B0**, **B1**, **B00**, **B11**, **C0**, **D0**). **PaVeToABCD** is nearly the inverse of **ToPaVe2**.

9.100.1 See also

Overview, ToPaVe, ToPaVe2, A0, A00, B0, B1, B00, B11, C0, D0.

9.100.2 Examples

```
PaVe[0, {pp}, {m1^2, m2^2}]

ex = PaVeToABCD[%]
```

$$\mathrm{B}_0\left(\mathrm{pp}, \mathrm{m1}^2, \mathrm{m2}^2\right)$$

$$\mathrm{B}_0\left(\mathrm{pp}, \mathrm{m1}^2, \mathrm{m2}^2\right)$$

```
ex // FCI // StandardForm

(*B0[pp, m1^2, m2^2]*)
```

```
PaVe[0, {SPD[p1], 0, SPD[p2]}, {m1^2, m2^2, m3^2}]

ex = PaVeToABCD[%]
```

$$\mathrm{C}_0\left(0, \mathrm{p1}^2, \mathrm{p2}^2, \mathrm{m3}^2, \mathrm{m2}^2, \mathrm{m1}^2\right)$$

$$\mathrm{C}_0\left(0, \mathrm{p1}^2, \mathrm{p2}^2, \mathrm{m3}^2, \mathrm{m2}^2, \mathrm{m1}^2\right)$$

```
ex // FCI // StandardForm

(*C0[0, Pair[Momentum[p1, D], Momentum[p1, D]], Pair[Momentum[p2, D],
Momentum[p2, D]], m3^2, m2^2, m1^2]*)
```

```
PaVe[0, 0, {SPD[p1], 0, SPD[p2]}, {m1^2, m2^2, m3^2}]

ex = PaVeToABCD[%]
```

$$C_{00}\left(0, p1^2, p2^2, m3^2, m2^2, m1^2\right)$$

$$C_{00}\left(0, p1^2, p2^2, m3^2, m2^2, m1^2\right)$$

```
ex // FCI // StandardForm

(*PaVe[0, 0, {0, Pair[Momentum[p1, D], Momentum[p1, D]], Pair[Momentum[p2,
D], Momentum[p2, D]]}, {m3^2, m2^2, m1^2}]*)
```

9.101 PaVeOrder

PaVeOrder[expr] orders the arguments of PaVe functions in expr in a standard way.

PaVeOrder[expr, PaVeOrderList -> { {..., s, u, ...}, {... m1^2, m2^2, ...}, ...}] orders the arguments of **PaVe** functions in **expr** according to the specified ordering lists. The lists may contain only a subsequence of the kinematic variables.

PaVeOrder has knows about symmetries in the arguments of PaVe functions with up to 6 legs.

Available symmetry relations are saved here

```
FileBaseName /@ FileNames["*.sym", FileNameJoin[{$FeynCalcDirectory,
"Tables",
     "PaVeSymmetries"}]]
```

{ScalarFunctions, TensorBFunctions, TensorCFunctions,
TensorDFunctions, TensorEFunctions, TensorFFunctions}

For the time being, these tables contain relations for B-functions up to rank 10, C-functions up to rank 9, D-functions up to rank 8, E-functions (5-point functions) up to rank 7 and F-functions (6-point functions) up to rank 4. If needed, relations for more legs and higher tensor ranks can be calculated using FeynCalc and saved to PaVeSymmetries using template codes provided inside ***.sym** files.

9.101.1 See also

Overview, PaVeReduce.

9.101.2 Examples

```
ClearAll[t, s]
```

Use PaVeOrder to change the ordering of arguments in a **D0** function

```
ex = D0[me2, me2, mw2, mw2, t, s, me2, 0, me2, 0]
```

$$\mathrm{D_0}(\mathrm{me2}, \mathrm{me2}, \mathrm{mw2}, \mathrm{mw2}, t, s, \mathrm{me2}, 0, \mathrm{me2}, 0)$$

```
PaVeOrder[ex, PaVeOrderList -> {me2, me2, 0, 0}]
```

$$\mathrm{D_0}(\mathrm{me2}, s, \mathrm{mw2}, t, \mathrm{mw2}, \mathrm{me2}, \mathrm{me2}, 0, 0, \mathrm{me2})$$

Different orderings are possible

```
PaVeOrder[D0[me2, me2, mw2, mw2, t, s, me2, 0, me2, 0], PaVeOrderList ->
{0, 0, me2, me2}]
```

$$\mathrm{D_0}(s, \mathrm{me2}, t, \mathrm{mw2}, \mathrm{mw2}, \mathrm{me2}, 0, 0, \mathrm{me2}, \mathrm{me2})$$

When applying the function to an amplitude containing multiple PaVe functions, one can specify a list of possible orderings

```
ex = D0[a, b, c, d, e, f, m12, m22, m32, m42] + D0[me2, me2, mw2, mw2, t,
s, me2, 0, me2, 0]
```

$$\mathrm{D_0}(a, b, c, d, e, f, \mathrm{m12}, \mathrm{m22}, \mathrm{m32}, \mathrm{m42}) + \mathrm{D_0}(\mathrm{me2}, \mathrm{me2}, \mathrm{mw2}, \mathrm{mw2}, t, s, \mathrm{me2}, 0, \mathrm{me2}, 0)$$

```
PaVeOrder[ex, PaVeOrderList -> {{me2, me2, 0, 0}, {f, e}}]
```

$$\mathrm{D_0}(a, b, c, d, e, f, \mathrm{m12}, \mathrm{m22}, \mathrm{m32}, \mathrm{m42}) + \mathrm{D_0}(\mathrm{me2}, s, \mathrm{mw2}, t, \mathrm{mw2}, \mathrm{me2}, \mathrm{me2}, 0, 0, \mathrm{me2})$$

PaVeOrder can be useful to show that a particular linear combination of **PaVe** functions yields zero

```
diff = PaVe[0, 0, {p14, p30, p24, p13, p20, p40, p34, p23, p12, p10}, {m4,
m3, m2,m1, m0},
    PaVeAutoOrder -> False] - PaVe[0, 0, {p10, p13, p12, p40, p30, p34,
p20, p24, p14, p23},
    {m3, m0, m1, m4, m2}, PaVeAutoOrder -> False]
```

$$\mathrm{E}_{00}(\mathrm{p14}, \mathrm{p30}, \mathrm{p24}, \mathrm{p13}, \mathrm{p20}, \mathrm{p40}, \mathrm{p34}, \mathrm{p23}, \mathrm{p12}, \mathrm{p10}, \mathrm{m4}, \mathrm{m3}, \mathrm{m2}, \mathrm{m1},$$
$$\mathrm{m0}) - \mathrm{E}_{00}(\mathrm{p10}, \mathrm{p13}, \mathrm{p12}, \mathrm{p40}, \mathrm{p30}, \mathrm{p34}, \mathrm{p20}, \mathrm{p24}, \mathrm{p14}, \mathrm{p23}, \mathrm{m3}, \mathrm{m0}, \mathrm{m1}, \mathrm{m4}, \mathrm{m2})$$

```
diff // PaVeOrder
```

$$0$$

In most cases, such simplifications require not only 1-to-1 relations but also linear relations between **PaVe** functions. For example, here we have a 1-to-1 relation between C_1 and C_2

```
PaVe[2, {p10, p12, p20}, {m1^2, m2^2, m3^2}, PaVeAutoOrder -> False]

PaVeOrder[%]
```

$$\mathrm{C}_2\left(\mathrm{p10}, \mathrm{p12}, \mathrm{p20}, \mathrm{m1}^2, \mathrm{m2}^2, \mathrm{m3}^2\right)$$

$$\mathrm{C}_1\left(\mathrm{p12}, \mathrm{p20}, \mathrm{p10}, \mathrm{m2}^2, \mathrm{m3}^2, \mathrm{m1}^2\right)$$

It seems that **PaVeOrder** cannot rewrite C_1 in such a way, that the mass arguments appear as m_2^2, m_1^2, m_3^2

```
ex = PaVe[1, {p10, p12, p20}, {m1^2, m2^2, m3^2}, PaVeAutoOrder -> False]
```

$$\mathrm{C}_1\left(\mathrm{p10}, \mathrm{p12}, \mathrm{p20}, \mathrm{m1}^2, \mathrm{m2}^2, \mathrm{m3}^2\right)$$

```
PaVeOrder[ex, PaVeOrderList -> {m2, m1, m3}]
```

$$\mathrm{C}_1\left(\mathrm{p10}, \mathrm{p12}, \mathrm{p20}, \mathrm{m1}^2, \mathrm{m2}^2, \mathrm{m3}^2\right)$$

In fact, such a rewriting is possible, but it involves a linear relation between multiple **PaVe** functions. To avoid an accidental expression swell, by default **PaVeOrder** uses only 1-to-1 relations. Setting the option **Sum** to **True** allows the routine to return linear relations too

```
PaVeOrder[ex, PaVeOrderList -> {m2, m1, m3}, Sum -> True]
```

$$-\mathrm{C}_0\left(\mathrm{p10}, \mathrm{p20}, \mathrm{p12}, \mathrm{m2}^2, \mathrm{m1}^2, \mathrm{m3}^2\right) - \mathrm{C}_1\left(\mathrm{p10}, \mathrm{p20}, \mathrm{p12}, \mathrm{m2}^2, \mathrm{m1}^2, \mathrm{m3}^2\right) - \mathrm{C}_2\left(\mathrm{p10}, \mathrm{p20}, \mathrm{p12}, \mathrm{m2}^2, \mathrm{m1}^2, \mathrm{m3}^2\right)$$

When trying to minimize the number of **PaVe** functions in the expression, one often has to try different orderings first

```
diff = (C0[0, SP[p, p], SP[p, p], 0, 0, 0] + 2 PaVe[1, {0, SP[p, p], SP[p,
p]}, {0, 0, 0}] +
    PaVe[1, {SP[p, p], SP[p, p], 0}, {0, 0, 0}])
```

$$\mathrm{C}_0\left(0,\overline{p}^2,\overline{p}^2,0,0,0\right)+2\,\mathrm{C}_1\left(0,\overline{p}^2,\overline{p}^2,0,0,0\right)+\mathrm{C}_1\left(\overline{p}^2,\overline{p}^2,0,0,0,0\right)$$

This ordering doesn't look very helpful

```
PaVeOrder[diff, PaVeOrderList -> {0, SP[p, p], SP[p, p]}, Sum -> True]

% // PaVeOrder
```

$$\mathrm{C}_0\left(0,\overline{p}^2,\overline{p}^2,0,0,0\right)+2\,\mathrm{C}_1\left(0,\overline{p}^2,\overline{p}^2,0,0,0\right)+\mathrm{C}_2\left(0,\overline{p}^2,\overline{p}^2,0,0,0\right)$$

$$\mathrm{C}_0\left(0,\overline{p}^2,\overline{p}^2,0,0,0\right)+2\,\mathrm{C}_1\left(0,\overline{p}^2,\overline{p}^2,0,0,0\right)+\mathrm{C}_1\left(\overline{p}^2,\overline{p}^2,0,0,0,0\right)$$

But this one does the job

```
PaVeOrder[diff, PaVeOrderList -> {SP[p, p], 0, SP[p, p]}, Sum -> True]

% // PaVeOrder
```

$$\mathrm{C}_1\left(\overline{p}^2,0,\overline{p}^2,0,0,0\right)-\mathrm{C}_2\left(\overline{p}^2,0,\overline{p}^2,0,0,0\right)$$

$$0$$

Here are few simpler cases

```
diff = PaVe[0, {0}, {m2^2, m3^2}] + PaVe[1, {0}, {m3^2, m2^2}] + PaVe[1,
{0}, {m2^2, m3^2}]
```

$$\mathrm{B}_0\left(0,\mathrm{m2}^2,\mathrm{m3}^2\right)+\mathrm{B}_1\left(0,\mathrm{m2}^2,\mathrm{m3}^2\right)+\mathrm{B}_1\left(0,\mathrm{m3}^2,\mathrm{m2}^2\right)$$

```
PaVeOrder[diff, PaVeOrderList -> {m2, m3}, Sum -> True]
```

$$0$$

```
diff = PaVe[0, {0}, {m2^2, m3^2}] + 2 PaVe[1, {0}, {m3^2, m2^2}] - PaVe[1,
1, {0}, {m2^2, m3^2}] +
   PaVe[1, 1, {0}, {m3^2, m2^2}]
```

$$B_0\left(0, m2^2, m3^2\right) + 2\, B_1\left(0, m3^2, m2^2\right) - B_{11}\left(0, m2^2, m3^2\right) + B_{11}\left(0, m3^2, m2^2\right)$$

```
PaVeOrder[diff, Sum -> True, PaVeOrderList -> {m3, m2}]
```

$$0$$

```
diff = PaVe[0, {0, 0, 0}, {m2^2, m3^2, m4^2}] + 2 PaVe[1, {0, 0, 0}, {m2^2,
m3^2, m4^2}] +
   2 PaVe[1, {0, 0, 0}, {m3^2, m2^2, m4^2}] + PaVe[1, 1, {0, 0, 0}, {m2^2,
   m3^2, m4^2}] -
   PaVe[1, 1, {0, 0, 0}, {m2^2, m4^2, m3^2}] + PaVe[1, 1, {0, 0, 0}, {m3^2,
m2^2, m4^2}] +
   2 PaVe[1, 2, {0, 0, 0}, {m4^2, m2^2, m3^2}]
```

$$C_0\left(0, 0, 0, m2^2, m3^2, m4^2\right) + 2\, C_1\left(0, 0, 0, m2^2, m3^2, m4^2\right) + 2\, C_1\left(0, 0, 0, m3^2, m2^2, m4^2\right) + C_{11}\left(0, 0, 0, m2^2, m3^2, m4^2\right) - C_{11}\left(0, 0, 0, m2^2, m4^2, m3^2\right) + C_{11}\left(0, 0, 0, m3^2, m2^2, m4^2\right) + 2\, C_{12}\left(0, 0, 0, m4^2, m2^2, m3^2\right)$$

```
PaVeOrder[diff, Sum -> True, PaVeOrderList -> {m3, m2}]
```

$$0$$

9.102 PaVeLimitTo4

`PaVeLimitTo4[expr]` simplifies products of Passarino-Veltman functions and D-dependent prefactors by evaluating the prefactors at $D = 4$ and adding an extra term from the product of $(D - 4)$ and the UV pole of the Passarino-Veltman function.

This is possible because the UV poles of arbitrary Passarino-Veltman functions can be determined via **`PaVeUVPart`**. The result is valid up to 0th order in Epsilon, i.e. it is sufficient for 1-loop calculations.

Warning! This simplification always ignores possible IR poles of Passarino-Veltman functions. Therefore, it can be used only if all IR poles are regulated without using dimensional regularization (e.g. by assigning gluons or photons a fake mass) or if it is known in advance that the given expression is free of IR singularities.

The application of **`PaVeLimitTo4`** is equivalent to using the old **`OneLoop`** routine with the flags **`$LimitTo4`** and **`$LimitTo4IRUnsafe`** set to **`True`**.

9.102.1 See also

Overview, $LimitTo4.

9.102.2 Examples

```
ex = (D - 2)/(D - 3) A0[m^2]

PaVeLimitTo4[ex]
```

$$\frac{(D-2)\,\mathrm{A}_0\left(m^2\right)}{D-3}$$

$$2\,\mathrm{A}_0\left(m^2\right)+2m^2$$

Simplify the 1-loop amplitude for $H \to gg$

```
ex = (-(1/((-2 + D) mH^2 mW sinW)) 2 I (-4 + D) e gs^2 mt^2 \[Pi]^2
B0[mH^2, mt^2, mt^2]
     SD[Glu2, Glu3] (-2 SPD[k1, Polarization[k2, -I, Transversality ->
True]]
        SPD[k2, Polarization[k1, -I, Transversality -> True]] +
       mH^2 SPD[Polarization[k1, -I, Transversality -> True],
         Polarization[k2, -I, Transversality -> True]]) - 1/((-2 +
          D) mH^2 mW sinW) I e gs^2 mt^2 (-2 mH^2 + D mH^2 -
       8 mt^2) \[Pi]^2 C0[0, 0, mH^2, mt^2, mt^2, mt^2] SD[Glu2, Glu3] (-2
       SPD[k1,
         Polarization[k2, -I, Transversality -> True]] SPD[k2,
Polarization[k1,
          -I, Transversality -> True]] + mH^2 SPD[Polarization[k1, -I,
          Transversality -> True], Polarization[k2, -I, Transversality ->
True]]))
```

$$-\frac{2i\pi^2(D-4)e\,\mathrm{gs}^2\,\mathrm{mt}^2\delta^{\mathrm{Glu2\,Glu3}}\,\mathrm{B}_0\left(\mathrm{mH}^2,\mathrm{mt}^2,\mathrm{mt}^2\right)\left(\mathrm{mH}^2\left(\varepsilon^*(\mathrm{k1})\cdot\varepsilon^*(\mathrm{k2})\right)-2\left(\mathrm{k1}\cdot\varepsilon^*(\mathrm{k2})\right)\left(\mathrm{k2}\cdot\varepsilon^*(\mathrm{k1})\right)\right)}{(D-2)\,\mathrm{mH}^2\,\mathrm{mW}\,\mathrm{sinW}}$$

$$-\frac{i\pi^2 e\,\mathrm{gs}^2\,\mathrm{mt}^2\left(D\,\mathrm{mH}^2-2\,\mathrm{mH}^2-8\,\mathrm{mt}^2\right)\delta^{\mathrm{Glu2\,Glu3}}\,\mathrm{C}_0\left(0,0,\mathrm{mH}^2,\mathrm{mt}^2,\mathrm{mt}^2,\mathrm{mt}^2\right)\left(\mathrm{mH}^2\left(\varepsilon^*(\mathrm{k1})\cdot\varepsilon^*(\mathrm{k2})\right)-2\left(\mathrm{k1}\cdot\varepsilon^*(\right.\right.}{(D-2)\,\mathrm{mH}^2\,\mathrm{mW}\,\mathrm{sinW}}$$

```
PaVeLimitTo4[ex]
```

$$\frac{i\pi^2 e\,\mathrm{gs}^2\,\mathrm{mt}^2\left(\mathrm{mH}^2-4\,\mathrm{mt}^2\right)\delta^{\mathrm{Glu2\,Glu3}}\,\mathrm{C}_0\left(0,0,\mathrm{mH}^2,\mathrm{mt}^2,\mathrm{mt}^2,\mathrm{mt}^2\right)\left(2\left(\overline{\mathrm{k1}}\cdot\bar{\varepsilon}^*(\mathrm{k2})\right)\left(\overline{\mathrm{k2}}\cdot\bar{\varepsilon}^*(\mathrm{k1})\right)-\mathrm{mH}^2\left(\bar{\varepsilon}^*(\mathrm{k1})\cdot\bar{\varepsilon}^*\right.\right.}{\mathrm{mH}^2\,\mathrm{mW}\,\mathrm{sinW}}$$

$$-\frac{2i\pi^2 e\,\mathrm{gs}^2\,\mathrm{mt}^2\delta^{\mathrm{Glu2\,Glu3}}\left(2\left(\overline{\mathrm{k1}}\cdot\bar{\varepsilon}^*(\mathrm{k2})\right)\left(\overline{\mathrm{k2}}\cdot\bar{\varepsilon}^*(\mathrm{k1})\right)-\mathrm{mH}^2\left(\bar{\varepsilon}^*(\mathrm{k1})\cdot\bar{\varepsilon}^*(\mathrm{k2})\right)\right)}{\mathrm{mH}^2\,\mathrm{mW}\,\mathrm{sinW}}$$

9.103 PaVeReduce

PaVeReduce[expr] reduces all Passarino-Veltman integrals (i.e. all PaVe's) in expr down to scalar **A0**, **B0**, **C0** and **D0**.

9.103.1 See also

Overview, FRH, PaVeOrder.

9.103.2 Examples

```
PaVeReduce[PaVe[1, 2, {s, m^2, m^2}, {m^2, m^2, M^2}], IsolateNames -> FF]

FRH[%]
```

FF(31)

$$\left(\frac{2(2D-3)M^2}{(D-2)\left(4m^2-s\right)^2}-\frac{M^2 s}{2m^2\left(4m^2-s\right)^2}-\frac{4m^2}{\left(4m^2-s\right)^2}+\frac{s}{\left(4m^2-s\right)^2}\right)\mathrm{B}_0\left(m^2,m^2,M^2\right)+\left(-\frac{2(D-1)M^2}{(D-2)\left(4m^2-s\right)^2}+\frac{4m^2}{(D-2)\left(4m^2-s\right)^2}-\frac{s}{(D-2)\left(4m^2-s\right)^2}\right)\mathrm{B}_0\left(s,m^2,m^2\right)+\left(-\frac{2(D-1)M^4}{(D-2)\left(4m^2-s\right)^2}-\frac{DM^2 s}{(D-2)\left(4m^2-s\right)^2}+\frac{4Dm^2M^2}{(D-2)\left(4m^2-s\right)^2}\right)\mathrm{C}_0\left(m^2,m^2,s,m^2,M^2,m^2\right)-\frac{\mathrm{A}_0\left(M^2\right)}{2m^2\left(4m^2-s\right)}+\frac{\mathrm{A}_0\left(m^2\right)}{2m^2\left(4m^2-s\right)}$$

The reduction results can be saved to a Mathematica file

```
PaVeReduce[PaVe[2, {SmallVariable[me2], mw2, t}, {SmallVariable[me2], 0,
mw2}],
   WriteOutPaVe -> "p"]

TableForm[ReadList["pPaVe1Cmw2tsmame2C0mw2smame2.s", String]]

DeleteFile /@ FileNames["pPaVe1Cmw2tsmame2C0mw2smame2.s"];
```

$$\frac{\mathrm{B}_0(\mathrm{mw2},0,\mathrm{mw2})}{\mathrm{mw2}-t}-\frac{\mathrm{B}_0(t,\mathrm{mw2},\mathrm{me2})}{\mathrm{mw2}-t}$$

```
( PaVe[0, {mw2}, {0, mw2}]/(mw2 - t) - PaVe[0, {t}, {mw2, SmallVariable[me2]}]/
 (mw2 - t)
 )
```

Fortran export is also available

```
se = SmallVariable[ME2];

d122 = PaVeReduce[PaVe[1, 2, 2, {se, MW2, MW2, se, S, T}, {0, se, 0, se}],
Mandelstam -> {S, T, U, 2 MW2}, IsolateNames -> F] // FRH

Write2["fctd122.for", d122res == d122, FormatType -> FortranForm];

TableForm[ReadList["fctd122.for", String]]

DeleteFile /@ FileNames["fctd122.for"]; Clear[d122, se];
```

$$-\frac{D(\text{MW2}-S)T^2\,\text{D}_0(\text{MW2},\text{MW2},\text{ME2},\text{ME2},T,S,\text{ME2},0,\text{ME2},0)S^3}{8(3-D)\left(\text{MW2}^2-SU\right)^3}$$
$$-\frac{D(\text{MW2}-S)^2T\,\text{C}_0(\text{MW2},S,\text{ME2},\text{ME2},0,0)S^2}{4(3-D)\left(\text{MW2}^2-SU\right)^3}+\frac{D(\text{MW2}-S)T^2\,\text{C}_0(T,\text{ME2},\text{ME2},\text{ME2},\text{ME2},0)S^2}{8(3-D)\left(\text{MW2}^2-SU\right)^3}$$
$$-\frac{\left(2\,\text{MW2}^2-DST-2SU\right)\,\text{B}_0(S,0,0)S}{2(2-D)(\text{MW2}-S)\left(\text{MW2}^2-SU\right)^2}+\frac{(\text{MW2}+U)\,\text{A}_0(\text{ME2})}{2\,\text{MW2}(4\,\text{MW2}-T)\left(\text{MW2}^2-SU\right)}$$
$$+\left(\left(-2D\,\text{MW2}^5+16\,\text{MW2}^5-3DS\,\text{MW2}^4+2S\,\text{MW2}^4-3DU\,\text{MW2}^4+10U\,\text{MW2}^4\right.\right.$$
$$-8DS^2\,\text{MW2}^3+4DSU\,\text{MW2}^3-16SU\,\text{MW2}^3+2DS^3\,\text{MW2}^2+4DSU^2\,\text{MW2}^2-8SU^2\,\text{MW2}^2$$
$$\left.\left.-2DS^2U\,\text{MW2}^2+2DS^4\,\text{MW2}+4DS^3U\,\text{MW2}+DS^2U^3-2S^2U^3+DS^3U^2-2S^3U^2\right)\,\text{B}_0(\text{MW2},$$
$$0,\text{ME2})\right)/\left(2(2-D)(\text{MW2}-S)(4\,\text{MW2}-T)^2\left(\text{MW2}^2-SU\right)^2\right)$$
$$-\left(\left(20\,\text{MW2}^4-2DS\,\text{MW2}^3-12S\,\text{MW2}^3-6T\,\text{MW2}^3-2DU\,\text{MW2}^3-4DS^2\,\text{MW2}^2+2ST\,\text{MW2}^2-20SU\,\text{MW2}^2+DS^3\,\text{MW}\right.\right.$$
$$\left.\left.\text{ME2},\text{ME2})\right)/\left(2(2-D)(4\,\text{MW2}-T)^2\left(\text{MW2}^2-SU\right)^2\right)\right.$$
$$-\left(\left(128D\,\text{MW2}^7+48\,\text{MW2}^7-608DS\,\text{MW2}^6-128DT\,\text{MW2}^6-144U\,\text{MW2}^6-4D^2S^2\,\text{MW2}^5+1152DS^2\,\text{MW2}^5+28DT^2\,\text{M}\right.\right.$$
$$\left.\left.\text{MW2},T,\text{ME2},0,\text{ME2})\right)/\left(8(2-D)(3-D)(4\,\text{MW2}-T)^2\left(\text{MW2}^2-SU\right)^3\right)\right.$$

```
d122res = ((MW2 + U)*5.D-1*PaVe(0D0,List(),List(ME2)))/
& (MW2*(MW2**2 - S*U*1D0)*(-(T*1D0) + MW2*4D0)) +
& (5.D-1*(D*S**3*U**2 + D*S**2*U**3 - D*MW2**5*2D0 +
& MW2**4*S*2D0 + D*MW2**2*S**3*2D0 +
& D*MW2*S**4*2D0 - D*MW2**2*S**2*U*2D0 -
& S**3*U**2*2D0 - S**2*U**3*2D0 - D*MW2**4*S*3D0 -
& D*MW2**4*U*3D0 + D*MW2**3*S*U*4D0 +
& D*MW2*S**3*U*4D0 + D*MW2**2*S*U**2*4D0 -
& D*MW2**3*S**2*8D0 - MW2**2*S*U**2*8D0 +
& MW2**4*U*1.D1 + MW2**5*1.6D1 - MW2**3*S*U*1.6D1)*
& PaVe(0D0,List(MW2),List(0D0,ME2)))/
& ((MW2 - S*1D0)*(MW2**2 - S*U*1D0)**2*
& (-(D*1D0) + 2D0)*(-(T*1D0) + MW2*4D0)**2) -
& (S*5.D-1*(-(D*S*T*1D0) + MW2**2*2D0 - S*U*2D0)*
& PaVe(0D0,List(S),List(0D0,0D0)))/
& ((MW2 - S*1D0)*(MW2**2 - S*U*1D0)**2*
& (-(D*1D0) + 2D0)) -
& (5.D-1*(D*MW2*S**3 + D*S**4 + D*S**2*U**2 -
& D*MW2**3*S*2D0 + MW2**2*S*T*2D0 -
& D*MW2**3*U*2D0 + D*S**3*U*2D0 - S**2*T*U*2D0 +
& D*MW2*S*U**2*3D0 - D*MW2**2*S**2*4D0 -
& MW2**3*T*6D0 + MW2*S*T*U*6D0 - MW2**3*S*1.2D1 +
& MW2*S**2*U*1.2D1 + MW2**4*2.D1 - MW2**2*S*U*2.D1)
& *PaVe(0D0,List(T),List(ME2,ME2)))/
& ((MW2**2 - S*U*1D0)**2*(-(D*1D0) + 2D0)*
& (-(T*1D0) + MW2*4D0)**2) -
& (1.25D-1*(-(D**2*MW2*S**6*1D0) - D**2*S**7*1D0 -
& D**2*S**3*U**4*1D0 + D*S**3*T**4*2D0 -
& D**2*MW2**3*S**3*U*2D0 - D**2*MW2*S**5*U*2D0 -
& D**2*MW2**5*S**2*4D0 - D**2*S**6*U*4D0 -
& D**2*MW2**5*U**2*4D0 - D**2*S**4*U**3*4D0 +
& D**2*MW2**3*S**4*6D0 - D**2*S**5*U**2*6D0 -
& D**2*MW2*S**3*U**3*6D0 -
& D**2*MW2*S**2*U**4*7D0 + D**2*MW2**2*S**5*8D0 -
& D**2*MW2**5*S*U*8D0 +
& D**2*MW2**2*S**3*U**2*8D0 +
& D**2*MW2**3*S*U**3*1.D1 -
& D**2*MW2**4*S**3*1.6D1 +
& D**2*MW2**2*S**4*U*1.6D1 +
& D**2*MW2**3*S**2*U**2*1.8D1 +
& D*MW2**5*T**2*2.8D1 + D*MW2*S**2*T**4*3.8D1 +
& D*MW2*S**5*T*4.D1 + MW2**7*4.8D1 +
& MW2**5*T*U*4.8D1 + MW2**3*S**2*U**2*4.8D1 +
& MW2*S**2*T*U**3*4.8D1 + D*MW2**3*S*T**3*6.4D1 -
& D*MW2**2*S**5*9.6D1 - MW2**5*S*U*9.6D1 -
& MW2**3*S*T*U**2*9.6D1 + D*MW2*S**3*T**3*1.D2 +
& D*MW2*S**4*T**2*1.2D2 + D*MW2**7*1.28D2 -
& D*MW2**6*T*1.28D2 - MW2**6*U*1.44D2 -
& MW2**2*S**2*U**3*1.44D2 + MW2**4*S*U**2*2.88D2 -
& D*MW2**2*S**2*T**3*3.24D2 -
& D*MW2**4*S*T**2*4.2D2 + D*MW2**3*S**4*5.12D2 -
& D*MW2**2*S**4*T*5.28D2 - D*MW2**6*S*6.08D2 -
& D*MW2**2*S**3*T**2*7.16D2 +
& D*MW2**5*S*T*8.88D2 - D**3*MW2**4*S**3*1.088D3 +
& D*MW2**3*S**2*T**2*1.116D3 +
& D*MW2**5*S**2*1.152D3 +
& D*MW2**3*S**3*T*1.568D3 - D*MW2**4*S**2*T*1.84D3)
& *PaVe(0D0,List(MW2,MW2,T),List(ME2,0D0,ME2)))/
```

9.104 PaVeUVPart

`PaVeUVPart[expr]` replaces all occurring Passarino-Veltman functions by their explicit values, where only the UV divergent part is preserved, while possible IR divergences and the finite part are discarded. The function uses the algorithm from arXiv:hep-ph/0609282 by G. Sulyok. This allows to treat Passarino-Veltman of arbitrary rank and multiplicity

9.104.1 See also

Overview, PaVe, PaVeReduce.

9.104.2 Examples

```
PaVeUVPart[A0[m^2]]
```

$$-\frac{2m^2}{D-4}$$

```
PaVeUVPart[x + y B0[SPD[p, p], 0, M^2]]
```

$$\frac{Dx-4x-2y}{D-4}$$

```
PaVe[0, 0, {p10, p12, p20}, {m1^2, m2^2, m3^2}]

PaVeUVPart[%]
```

$$\mathrm{C}_{00}\left(\mathrm{p10},\mathrm{p12},\mathrm{p20},\mathrm{m1}^2,\mathrm{m2}^2,\mathrm{m3}^2\right)$$

$$-\frac{1}{2(D-4)}$$

```
PaVe[0, 0, 0, 0, 0, 0, {p10, p12, p23, 0, p20, p13}, {m1^2, m2^2, m3^2,
m4^2}]

PaVeUVPart[%]
```

$$\mathrm{D}_{000000}\left(0,\mathrm{p10},\mathrm{p12},\mathrm{p23},\mathrm{p13},\mathrm{p20},\mathrm{m4}^2,\mathrm{m1}^2,\mathrm{m2}^2,\mathrm{m3}^2\right)$$

$$\frac{-5\,\mathrm{m1}^2-5\,\mathrm{m2}^2-5\,\mathrm{m3}^2-5\,\mathrm{m4}^2+\mathrm{p10}+\mathrm{p12}+\mathrm{p13}+\mathrm{p20}+\mathrm{p23}}{480(D-4)}$$

```
int = FVD[k + p, rho] FVD[k + p, si] FAD[k, {k + p, 0, 2}]

TID[int, k, UsePaVeBasis -> True]

% // PaVeUVPart[#, FCE -> True] &
```

$$\frac{(k+p)^{\text{rho}}(k+p)^{\text{si}}}{k^2.(k+p)^4}$$

$$i\pi^2 g^{\text{rho si}}\, \text{C}_{00}\left(0, p^2, p^2, 0, 0, 0\right) + i\pi^2 p^{\text{rho}} p^{\text{si}}\, \text{C}_{11}\left(p^2, p^2, 0, 0, 0, 0\right)$$

$$-\frac{i\pi^2 g^{\text{rho si}}}{2(D-4)}$$

9.105 SimplifyDeltaFunction

SimplifyDeltaFunction[exp, x] simplifies **f[x]*DeltaFunction[1-x]** to **Limit[f[x], x->1] DeltaFunction[1-x]** and applies a list of transformation rules for **DeltaFunctionPrime[1-x]*x^(OPEm-1)*f[x]** where **x^(OPEm-1)** is suppressed in **exp**.

9.105.1 See also

Overview, DeltaFunction, DeltaFunctionPrime.

9.105.2 Examples

```
g[x] DeltaFunction[1 - x]

SimplifyDeltaFunction[ %, x]
```

$$g(x)\delta(1-x)$$

$$\delta(1-x)\lim_{x\to 1} g(x)$$

```
g[x] DeltaFunctionPrime[1 - x]
SimplifyDeltaFunction[ %, x]
x Log[x] DeltaFunctionPrime[1 - x]
SimplifyDeltaFunction[ %, x]
```

$$g(x)\delta'(1-x)$$

$$\delta(1-x)\lim_{x\to 1} g'(x) + \delta'(1-x)\lim_{x\to 1} g(x)$$

$$x\log(x)\delta'(1-x)$$

$$\delta(1-x)$$

```
PolyLog[2, 1 - x] DeltaFunctionPrime[1 - x]
SimplifyDeltaFunction[ %, x]
```

$$\text{Li}_2(1-x)\delta'(1-x)$$

$$-\delta(1-x)$$

```
Log[x] PolyLog[2, 1 - x] DeltaFunctionPrime[1 - x]
SimplifyDeltaFunction[ %, x]
```

$$\text{Li}_2(1-x)\log(x)\delta'(1-x)$$

$$0$$

```
PolyLog[3, 1 - x] DeltaFunctionPrime[1 - x]
SimplifyDeltaFunction[ %, x]
```

$$\text{Li}_3(1-x)\delta'(1-x)$$

$$-\delta(1-x)$$

9.106 Sn

Sn is $\pi^{n/2}/(2\pi)^n$.

9.106.1 See also

Overview

9.106.2 Examples

9.107 TarcerToFC

`TarcerToFC[expr, {q1, q2}]` translates loop integrals in the TARCER-notation to the FeynCalc notation.

See **`TFI`** for details on the convention.

As in the case of **`ToTFI`**, the $\frac{1}{\pi^D}$ and $\frac{1}{\pi^{D/2}}$ prefactors are implicit, i.e. **`TarcerToFC`** doesn't add them.

To recover momenta from scalar products use the option **`ScalarProduct`** e.g. as in **`TarcerToFC[TBI[D, pp^2, {{1, 0}, {1, 0}}], {q1, q2}, ScalarProduct -> {{pp^2, p1}}]`**

9.107.1 See also

Overview, ToFI.

9.107.2 Examples

```
Tarcer`TFI[D, Pair[Momentum[p, D], Momentum[p, D]], {0, 0, 3, 2, 0},
  {{4, 0}, {2, 0}, {1, 0}, {0, 0}, {1, 0}}]
```

$$\text{Tarcer`TFI}\left[D, p^2, \{0,0,3,2,0\}, \begin{pmatrix} 4 & 0 \\ 2 & 0 \\ 1 & 0 \\ 0 & 0 \\ 1 & 0 \end{pmatrix}\right]$$

```
TarcerToFC[%, {q1, q2}]
```

$$\frac{(p\cdot \text{q1})^3 (p\cdot \text{q2})^2}{\left(\text{q1}^2\right)^4 . \left(\text{q2}^2\right)^2 . (\text{q1}-p)^2 . (\text{q1}-\text{q2})^2}$$

```
a1 Tarcer`TBI[D, pp^2, {{1, 0}, {1, 0}}] + b1 Tarcer`TBI[D, mm1, {{1, 0},
{1, 0}}]
```

$$\text{a1 TarcerTBI}\left(D, \text{pp}^2, \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}\right) + \text{b1 TarcerTBI}\left(D, \text{mm1}, \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}\right)$$

```
TarcerToFC[%, {q1, q2}, ScalarProduct -> {{pp^2, p1}, {mm1, p1}}, FCE ->
True]
```

$$\frac{\text{a1}}{\text{q1}^2.(\text{q1}-\text{p1})^2} + \frac{\text{b1}}{\text{q1}^2.(\text{q1}-\text{p1})^2}$$

9.108 TFIOrder

`TFIOrder[exp]` orders the arguments of some **`TFI`** functions in exp in a standard way.

9.108.1 See also

Overview, TarcerToFC.

9.108.2 Examples

```
Tarcer`TFI[D, p^2, {{1, M2}, {1, M1}, {1, M3}, {1, M4}, {1, M5}}]

TFIOrder[%]
```

$$\text{TarcerTFI}\left(D, p^2, \begin{pmatrix} 1 & \text{M2} \\ 1 & \text{M1} \\ 1 & \text{M3} \\ 1 & \text{M4} \\ 1 & \text{M5} \end{pmatrix}\right)$$

$$\text{TarcerTFI}\left(D, p^2, \begin{pmatrix} 1 & \text{M1} \\ 1 & \text{M2} \\ 1 & \text{M4} \\ 1 & \text{M3} \\ 1 & \text{M5} \end{pmatrix}\right)$$

```
((2*m2^4*m3^2 + m2^2*(-((-2 + D)*m1^2) + (-6 + D)*m3^2)*m4^2 + m4^2*(2*(-3
+
              D)*m1^4 + m3^2*(2*(-3 + D)*m3^2 - (-4 + D)*m4^2) +
              m1^2*(-4*(-3 + D)*m3^2 + (-2 +
                 D)*m4^2)) + (-(m2^2*(4*m3^2 + (-6 + D)*m4^2)) + m4^2*((-6 +
                 D)*m1^2 - (-2 +
                 D)*m3^2 + (-4 + D)*m4^2))*SPD[p, p] + (2*m3^2 - (-4 +
                 D)*m4^2)*SPD[p,
           p]^2)*(Tarcer`TFI[D, SPD[p, p], {{1, m1}, {1, m2}, {1, m3}, {1,
           m4}, {1, m3}}] -
        Tarcer`TFI[D, SPD[p, p], {{1, m3}, {1, m4}, {1, m1}, {1, m2}, {1,
m3}}]))/(4*(m2^4*
         m3^2 - m2^2*(m1^2 + m3^2)*m4^2 + m4^2*(m1^4 + m3^4 + m1^2*(-2*m3^2
+ m4^2)) -
        ((m1^2 + m3^2)*m4^2 + m2^2*(2*m3^2 - m4^2))*SPD[p, p] + m3^2*SPD[p,
p]^2))

TFIOrder[%]
```

$$\left(\begin{array}{l}\left(p^2\left(\text{m4}^2\left((D-6)\,\text{m1}^2-(D-2)\,\text{m3}^2+(D-4)\,\text{m4}^2\right)-\text{m2}^2\left((D-6)\,\text{m4}^2+4\,\text{m3}^2\right)\right)\right.\\ +\text{m2}^2\,\text{m4}^2\left((D-6)\,\text{m3}^2-(D-2)\,\text{m1}^2\right)\\ +\text{m4}^2\left(2(D-3)\,\text{m1}^4+\text{m1}^2\left((D-2)\,\text{m4}^2-4(D-3)\,\text{m3}^2\right)+\text{m3}^2\left(2(D-3)\,\text{m3}^2-(D-4)\,\text{m4}^2\right)\right)\\ \left.+p^4\left(2\,\text{m3}^2-(D-4)\,\text{m4}^2\right)+2\,\text{m2}^4\,\text{m3}^2\right)\left(\text{TarcerTFI}\left(D,p^2,\begin{pmatrix}1&\text{m1}\\1&\text{m2}\\1&\text{m3}\\1&\text{m4}\\1&\text{m3}\end{pmatrix}\right)-\text{TarcerTFI}\left(D,p^2,\right.\right.\\ \left.\left.\left.\begin{pmatrix}1&\text{m3}\\1&\text{m4}\\1&\text{m1}\\1&\text{m2}\\1&\text{m3}\end{pmatrix}\right)\right)\Big/\left(4\left(-p^2\left(\text{m4}^2\left(\text{m1}^2+\text{m3}^2\right)+\text{m2}^2\left(2\,\text{m3}^2-\text{m4}^2\right)\right)-\text{m2}^2\,\text{m4}^2\left(\text{m1}^2+\text{m3}^2\right)+\text{m4}^2\left(\text{m1}^4+\text{m1}^2\left(\text{m4}^2\right.\right.\right.\right.\end{array}$$

$$0$$

9.109 TID

`TID[amp, q]` performs tensor decomposition of 1-loop integrals with loop momentum **`q`**.

9.109.1 See also

Overview, OneLoopSimplify, TIDL, PaVeLimitTo4.

9.109.2 Examples

```
FCClearScalarProducts[];
```

```
int = FAD[{k, m}, k - Subscript[p, 1], k - Subscript[p, 2]] FVD[k, \[Mu]]
// FCI
```

$$\frac{k^{\mu}}{\left(k^2-m^2\right).(k-p_1)^2.(k-p_2)^2}$$

By default, all tensor integrals are reduced to the Passarino-Veltman scalar integrals A_0, B_0, C_0, D_0 etc.

```
TID[int, k]
```

$$\frac{{p_1}^2 p_2^{\mu}-p_1^{\mu}\left(p_1\cdot p_2\right)}{2\left(\left(p_1\cdot p_2\right)^2-{p_1}^2{p_2}^2\right)k^2.\left((k+p_1)^2-m^2\right)}$$
$$-\frac{{p_2}^2\left(m^2+{p_1}^2\right)p_1^{\mu}+{p_1}^2\left(m^2+{p_2}^2\right)p_2^{\mu}+\left(m^2+{p_1}^2\right)\left(-p_2^{\mu}\right)\left(p_1\cdot p_2\right)-\left(m^2+{p_2}^2\right)p_1^{\mu}\left(p_1\cdot p_2\right)}{2\left(\left(p_1\cdot p_2\right)^2-{p_1}^2{p_2}^2\right)\left(k^2-m^2\right).(k-p_2)^2.(k-p_1)^2}$$
$$-\frac{p_2^{\mu}\left(p_1\cdot p_2\right)-{p_2}^2p_1^{\mu}}{2\left(\left(p_1\cdot p_2\right)^2-{p_1}^2{p_2}^2\right)k^2.\left((k+p_2)^2-m^2\right)}-\frac{{p_1}^2p_2^{\mu}+{p_2}^2p_1^{\mu}-p_1^{\mu}\left(p_1\cdot p_2\right)-p_2^{\mu}\left(p_1\cdot p_2\right)}{2k^2.(k-p_1+p_2)^2\left(\left(p_1\cdot p_2\right)^2-{p_1}^2{p_2}^2\right)}$$

Scalar integrals can be converted to the Passarino-Veltman notation via the option **ToPaVe**

```
TID[int, k, ToPaVe -> True]
```

$$\frac{i\pi^2\left({p_1}^2p_2^{\mu}-p_1^{\mu}\left(p_1\cdot p_2\right)\right)\ \mathrm{B}_0\left({p_1}^2,0,m^2\right)}{2\left(\left(p_1\cdot p_2\right)^2-{p_1}^2{p_2}^2\right)}-\frac{i\pi^2\left(p_2^{\mu}\left(p_1\cdot p_2\right)-{p_2}^2p_1^{\mu}\right)\ \mathrm{B}_0\left({p_2}^2,0,m^2\right)}{2\left(\left(p_1\cdot p_2\right)^2-{p_1}^2{p_2}^2\right)}$$
$$-\frac{i\pi^2\left({p_1}^2p_2^{\mu}+{p_2}^2p_1^{\mu}-p_1^{\mu}\left(p_1\cdot p_2\right)-p_2^{\mu}\left(p_1\cdot p_2\right)\right)\ \mathrm{B}_0\left({p_1}^2-2\left(p_1\cdot p_2\right)+{p_2}^2,0,0\right)}{2\left(\left(p_1\cdot p_2\right)^2-{p_1}^2{p_2}^2\right)}$$
$$-\frac{i\pi^2\left({p_2}^2\left(m^2+{p_1}^2\right)p_1^{\mu}+{p_1}^2\left(m^2+{p_2}^2\right)p_2^{\mu}+\left(m^2+{p_1}^2\right)\left(-p_2^{\mu}\right)\left(p_1\cdot p_2\right)-\left(m^2+{p_2}^2\right)p_1^{\mu}\left(p_1\cdot p_2\right)\right)\ \mathrm{C}_0\left({p_1}^2,p_2\right.}{2\left(\left(p_1\cdot p_2\right)^2-{p_1}^2{p_2}^2\right)}$$

We can force the reduction algorithm to use Passarino-Veltman coefficient functions via the option **UsePaVeBasis**

```
TID[int, k, UsePaVeBasis -> True]
```

$$-i\pi^2 {p_1}^{\mu}\, \mathrm{C}_1\left({p_1}^2, {p_1}^2 + {p_2}^2 - 2\,(p_1 \cdot p_2), {p_2}^2, m^2, 0, 0\right) - i\pi^2 {p_2}^{\mu}\, \mathrm{C}_1\left({p_2}^2, {p_1}^2 + {p_2}^2 - 2\,(p_1 \cdot p_2), {p_1}^2, m^2, 0, 0\right)$$

Very often the integral can be simplified via partial fractioning even before performing the loop reduction. In this case the output will contain a mixture of **FAD** symbols and Passarino-Veltman functions

```
TID[SPD[p, q] FAD[q, {q - p, m}] FVD[q, mu], q, UsePaVeBasis -> True]
```

$$\frac{p^{\mathrm{mu}}}{2\,(q^2 - m^2)} + \frac{1}{2} i\pi^2 \left(m^2 - p^2\right) p^{\mathrm{mu}} \left(\frac{\left(m^2 - p^2\right)\, \mathrm{B}_0\left(p^2, 0, m^2\right)}{2p^2} - \frac{\mathrm{A}_0\left(m^2\right)}{2p^2}\right)$$

This can be avoided by setting both **UsePaVeBasis** and **ToPaVe** to **True**

```
TID[SPD[p, q] FAD[q, {q - p, m}] FVD[q, mu], q, UsePaVeBasis -> True,
ToPaVe -> True]
```

$$\frac{1}{2} i\pi^2 \left(m^2 - p^2\right) p^{\mathrm{mu}} \left(\frac{\left(m^2 - p^2\right)\, \mathrm{B}_0\left(p^2, 0, m^2\right)}{2p^2} - \frac{\mathrm{A}_0\left(m^2\right)}{2p^2}\right) + \frac{1}{2} i\pi^2 p^{\mathrm{mu}}\, \mathrm{A}_0\left(m^2\right)$$

Alternatively, we may set **ToPaVe** to **Automatic** which will automatically invoke the **ToPaVe** function if the final result contains even a single Passarino-Veltman function

```
TID[SPD[p, q] FAD[q, {q - p, m}] FVD[q, mu], q, ToPaVe -> Automatic]
```

$$\frac{\left(m^2 - p^2\right)^2 p^{\mathrm{mu}}}{4p^2 q^2 .\left((q-p)^2 - m^2\right)} - \frac{\left(m^2 - 3p^2\right) p^{\mathrm{mu}}}{4p^2\left(q^2 - m^2\right)}$$

```
TID[SPD[p, q] FAD[q, {q - p, m}] FVD[q, mu], q, UsePaVeBasis -> True,
ToPaVe -> Automatic]
```

$$\frac{1}{2} i\pi^2 \left(m^2 - p^2\right) p^{\mathrm{mu}} \left(\frac{\left(m^2 - p^2\right)\, \mathrm{B}_0\left(p^2, 0, m^2\right)}{2p^2} - \frac{\mathrm{A}_0\left(m^2\right)}{2p^2}\right) + \frac{1}{2} i\pi^2 p^{\mathrm{mu}}\, \mathrm{A}_0\left(m^2\right)$$

The basis of Passarino-Veltman coefficient functions is used automatically if there are zero Gram determinants

```
FCClearScalarProducts[];

SPD[Subscript[p, 1], Subscript[p, 1]] = 0;

SPD[Subscript[p, 2], Subscript[p, 2]] = 0;

SPD[Subscript[p, 1], Subscript[p, 2]] = 0;

TID[FAD[{k, m}, k - Subscript[p, 1], k - Subscript[p, 2]] FVD[k, \[Mu]] //
FCI, k]

FCClearScalarProducts[];
```

$$-i\pi^2 \left(p_1{}^{\mu} + p_2{}^{\mu}\right) \, \mathrm{C}_1\left(0,0,0,0,0,m^2\right)$$

In FeynCalc, Passarino-Veltman coefficient functions are defined in the same way as in LoopTools. If one wants to use a different definition, it is useful to activate the option GenPaVe

```
FCClearScalarProducts[];

SPD[Subscript[p, 1], Subscript[p, 1]] = 0;

SPD[Subscript[p, 2], Subscript[p, 2]] = 0;

SPD[Subscript[p, 1], Subscript[p, 2]] = 0;

TID[FAD[{k, m}, k - Subscript[p, 1], k - Subscript[p, 2]] FVD[k, \[Mu]] //
FCI, k, GenPaVe -> True]

FCClearScalarProducts[];
```

$$-i\pi^2 p_1{}^{\mu} \, \mathrm{GenPaVe}\left(\{1\}, \begin{pmatrix} 0 & m \\ p_1 & 0 \\ p_2 & 0 \end{pmatrix}\right) - i\pi^2 p_2{}^{\mu} \, \mathrm{GenPaVe}\left(\{2\}, \begin{pmatrix} 0 & m \\ p_1 & 0 \\ p_2 & 0 \end{pmatrix}\right)$$

To simplify manifestly IR-finite 1-loop results written in terms of Passarino-Veltman functions, we may employ the option **PaVeLimitTo4** (must be used together with **ToPaVe**). The result is valid up to 0th order in **Epsilon**, i.e. sufficient for 1-loop calculations.

```
FCClearScalarProducts[];

int = (D - 1) (D - 2)/(D - 3) FVD[p, mu] FVD[p, nu] FAD[p, p - q]
```

$$\frac{(D-2)(D-1)p^{\text{mu}}p^{\text{nu}}}{(D-3)p^2.(p-q)^2}$$

```
TID[int, p, ToPaVe -> True]
```

$$\frac{i\pi^2(2-D)\,\text{B}_0\left(q^2,0,0\right)\left(Dq^{\text{mu}}q^{\text{nu}}-q^2g^{\text{mu nu}}\right)}{4(3-D)}$$

```
TID[int, p, ToPaVe -> True, PaVeLimitTo4 -> True]
```

$$\frac{1}{2}i\pi^2\,\text{B}_0\left(\overline{q}^2,0,0\right)\left(4\overline{q}^{\text{mu}}\overline{q}^{\text{nu}}-\overline{q}^2\bar{g}^{\text{mu nu}}\right)+\frac{1}{2}i\pi^2\left(2\overline{q}^{\text{mu}}\overline{q}^{\text{nu}}-\overline{q}^2\bar{g}^{\text{mu nu}}\right)$$

Sometimes one would like to have external momenta multiplied by symbolic parameters in the propagators. In this case one should first declare the corresponding variables to be of **FCVariable** type

```
DataType[a, FCVariable] = True;
DataType[b, FCVariable] = True;
```

```
ExpandScalarProduct[SP[P, Q] /. P -> a P1 + b P2]

StandardForm[%]
```

$$a\left(\overline{\text{P1}}\cdot\overline{Q}\right)+b\left(\overline{\text{P2}}\cdot\overline{Q}\right)$$

```
(*a Pair[Momentum[P1], Momentum[Q]] + b Pair[Momentum[P2], Momentum[Q]]*)
```

9.110 TIDL

TIDL is a database of tensorial reduction formulas.

9.110.1 See also

Overview, TID.

9.110.2 Examples

```
TIDL[{q, mu}, {p}]
```

$$\frac{p^{\text{mu}}(p\cdot q)}{p^2}$$

```
TIDL[{q, mu}, {p1, p2}]
```

$$\frac{\text{p1}^{\text{mu}}\left((\text{p1}\cdot\ \text{p2})(\text{p2}\cdot q)-\text{p2}^2(\text{p1}\cdot q)\right)}{(\text{p1}\cdot\ \text{p2})^2-\text{p1}^2\,\text{p2}^2}\ -\ \frac{\text{p2}^{\text{mu}}\left(\text{p1}^2(\text{p2}\cdot q)-(\text{p1}\cdot\ \text{p2})(\text{p1}\cdot q)\right)}{(\text{p1}\cdot\ \text{p2})^2-\text{p1}^2\,\text{p2}^2}$$

```
TIDL[{{q1, mu}, {q2, nu}}, {p}]
```

$$\frac{g^{\text{mu nu}}\left((p\cdot\ \text{q1})(p\cdot\ \text{q2})-p^2(\text{q1}\cdot\ \text{q2})\right)}{(1-D)p^2}\ -\ \frac{p^{\text{mu}}p^{\text{nu}}\left(D(p\cdot\ \text{q1})(p\cdot\ \text{q2})-p^2(\text{q1}\cdot\ \text{q2})\right)}{(1-D)p^4}$$

```
TIDL[{{q1, mu}, {q2, nu}}, {}]
```

$$\frac{g^{\text{mu nu}}(\text{q1}\cdot\ \text{q2})}{D}$$

9.111 ToDistribution

`ToDistribution[exp, x]` replaces **`(1-x)^(a Epsilon - 1)`** in **`exp`** by **`1/(a Epsilon) DeltaFunction[1-x] + 1/(1-x) + a Epsilon Log[1-x]/(1-x) + 1/2 a^2 Epsilon^2 Log[1-x]^2/(1-x)]`** and **`(1-x)^(a Epsilon - 2)`** in **`exp`** by **`-1/(a Epsilon) DeltaFunctionPrime[1-x] + 1/(1-x)^2 + (a Epsilon) Log[1-x]/(1-x)^2 + a^2 Epsilon^2/2 Log[1-x]^2/(1-x)^2 + a^3 Epsilon^3/6 Log[1-x]^3/(1-x)^2`**.

9.111.1 See also

Overview, PlusDistribution.

9.111.2 Examples

```
ToDistribution[(1 - x)^(Epsilon - 1), x, PlusDistribution -> pd]
```

$$\frac{1}{6}\varepsilon^3\ \text{pd}\left(\frac{\log^3(1-x)}{1-x}\right)+\frac{1}{2}\varepsilon^2\ \text{pd}\left(\frac{\log^2(1-x)}{1-x}\right)+\varepsilon\ \text{pd}\left(\frac{\log(1-x)}{1-x}\right)+\text{pd}\left(\frac{1}{1-x}\right)+\frac{\delta(1-x)}{\varepsilon}$$

```
ToDistribution[(1 - x)^(Epsilon - 2), x, PlusDistribution -> Identity]
```

$$-\frac{\delta'(1-x)}{\varepsilon}+\frac{\varepsilon^3\log^3(1-x)}{6(1-x)^2}+\frac{\varepsilon^2\log^2(1-x)}{2(1-x)^2}+\frac{\varepsilon\log(1-x)}{(1-x)^2}+\frac{1}{(1-x)^2}$$

```
Series2[Integrate[(1 - x)^(Epsilon - 2), {x, 0, 1}, GenerateConditions ->
False], Epsilon, 3]
```

$$-\varepsilon^3-\varepsilon^2-\varepsilon-1$$

```
Integrate2[ToDistribution[(1 - x)^(Epsilon - 2), x], {x, 0, 1}]
```

$$-\varepsilon^3-\varepsilon^2-\varepsilon-1$$

9.112 ToFI

ToFI[expr, {q1, q2}, {p}] translates all non-tensorial loop integrals in expr into **TFI** notation from TARCER.

ToFI[expr, {q}, {p}] introduces **TBI B0**-like integrals.

ToFI can be extended to more external particles and more loops if needed.

9.112.1 See also

Overview, TarcerToFC.

9.112.2 Examples

9.113 ToTFI

ToTFI[expr, q1, q2, p] translates FeynCalc 2-loop self energy type integrals into the **TFI** notation, which can be used as input for the function **TarcerRecurse** from the TARCER package.

See the TARCER documentation on TFI for details on the conventions.

9.113.1 See also

Overview, TarcerToFC.

9.113.2 Examples

9.114 ToGFAD

ToGFAD[exp] converts all occurring propagator types (**FAD**, **SFAD**, **CFAD**) to **GFAD**s. This is mainly useful when doing expansions in kinematic invariants, where e.g. scalar products may not be appear explicitly when using **FAD**- or **SFAD**-notation.

ToGFAD is the inverse operation to FromGFAD.

9.114.1 See also

Overview, GFAD, SFAD, CFAD, FeynAmpDenominatorExplicit, FromGFAD

9.114.2 Examples

```
ToGFAD[FAD[p]]
```

$$\frac{1}{(p^2 + i\eta)}$$

```
ToGFAD[FAD[p]] // StandardForm

(*FeynAmpDenominator[GenericPropagatorDenominator[Pair[Momentum[p, D],
Momentum[p, D]], {1, 1}]]*)
```

```
ToGFAD[SFAD[{p + q, m^2}]]
```

$$\frac{1}{(-m^2 + p^2 + 2(p \cdot q) + q^2 + i\eta)}$$

```
ToGFAD[SFAD[{p + q, m^2}]] // StandardForm

(*FeynAmpDenominator[GenericPropagatorDenominator[-m^2 + Pair[Momentum[p,
D], Momentum[p, D]] + 2 Pair[Momentum[p, D], Momentum[q, D]] +
Pair[Momentum[q, D], Momentum[q, D]], {1, 1}]]*)
```

```
ToGFAD[SFAD[{p + q, m^2}], FinalSubstitutions -> {SPD[q] -> 0}]
```

$$\frac{1}{(-m^2 + p^2 + 2(p \cdot q) + i\eta)}$$

```
ToGFAD[SFAD[{p + q, m^2}], FinalSubstitutions -> {SPD[q] -> 0}] //
StandardForm

(*FeynAmpDenominator[GenericPropagatorDenominator[-m^2 + Pair[Momentum[p,
D], Momentum[p, D]] + 2 Pair[Momentum[p, D], Momentum[q, D]], {1, 1}]]*)
```

9.115 ToPaVe

`ToPaVe[exp, q]` converts all scalar 1-loop integrals in **`exp`** that depend on the momentum **`q`** to scalar Passarino Veltman functions **`A0`**, **`B0`**, **`C0`**, **`D0`** etc.

9.115.1 See also

Overview, PaVeToABCD, ToPaVe2, A0, A00, B0, B1, B00, B11, C0, D0.

9.115.2 Examples

```
FAD[{q, m1}]

ToPaVe[%, q]
```

$$\frac{1}{q^2 - \text{m1}^2}$$

$$i\pi^2 \, \text{A}_0\left(\text{m1}^2\right)$$

```
FAD[{q, m1}, {q + p1, m2}]

ToPaVe[%, q]
```

$$\frac{1}{\left(q^2 - \text{m1}^2\right) . \left((\text{p1} + q)^2 - \text{m2}^2\right)}$$

$$i\pi^2 \, \text{B}_0\left(\text{p1}^2, \text{m1}^2, \text{m2}^2\right)$$

```
% // StandardForm

(*I \[Pi]^2 B0[Pair[Momentum[p1, D], Momentum[p1, D]], m1^2, m2^2]*)
```

```
FAD[{q, m1}, {q + p1, m2}, {q + p2, m3}, {q + p3, m4}, {q + p4, m5}]

ToPaVe[%, q]
```

$$\frac{1}{(q^2-\text{m1}^2).((\text{p1}+q)^2-\text{m2}^2).((\text{p2}+q)^2-\text{m3}^2).((\text{p3}+q)^2-\text{m4}^2).((\text{p4}+q)^2-\text{m5}^2)}$$

$$i\pi^2\,\text{E}_0\left(\text{p1}^2,\text{p2}^2,-2(\text{p2}\cdot\text{p3})+\text{p2}^2+\text{p3}^2,-2(\text{p3}\cdot\text{p4})+\text{p3}^2+\text{p4}^2,\right.$$
$$-2(\text{p1}\cdot\text{p4})+\text{p1}^2+\text{p4}^2,-2(\text{p1}\cdot\text{p2})+\text{p1}^2+\text{p2}^2,\text{p3}^2,-2(\text{p2}\cdot\text{p4})+\text{p2}^2+\text{p4}^2,$$
$$\left.-2(\text{p1}\cdot\text{p3})+\text{p1}^2+\text{p3}^2,\text{p4}^2,\text{m2}^2,\text{m1}^2,\text{m3}^2,\text{m4}^2,\text{m5}^2\right)$$

By default, **ToPaVe** has the option **PaVeToABCD** set to **True**. This means that some of the **PaVe** functions are automatically converted to direct Passarino-Veltman functions (**A0**, **A00**, **B0**, **B1**, **B00**, **B11**, **C0**, **D0**). This also has consequences for **TID**

```
TID[FVD[q, mu] FAD[{q, m1}, {q + p}], q, ToPaVe -> True]
```

$$\frac{i\pi^2 p^{\text{mu}}\,\text{A}_0\left(\text{m1}^2\right)}{2p^2}-\frac{i\pi^2\left(\text{m1}^2+p^2\right)p^{\text{mu}}\,\text{B}_0\left(p^2,0,\text{m1}^2\right)}{2p^2}$$

```
% // StandardForm
```

$$\frac{i\pi^2\ \text{A0}\left[\text{m1}^2\right]\ \text{Pair}[\text{LorentzIndex}[\text{mu},D],\text{Momentum}[p,D]]}{2\ \text{Pair}[\text{Momentum}[p,D],\text{Momentum}[p,D]]}-\left(i\pi^2\ \text{B0}\left[\text{Pair}[\text{Momentum}[p,D],\right.\right.$$
$$\left.\left.\text{Momentum}[p,D]],0,\text{m1}^2\right]\ \text{Pair}[\text{LorentzIndex}[\text{mu},D],\text{Momentum}[p,D]]\left(\text{m1}^2+\text{Pair}[\text{Momentum}[p,\right.\right.$$
$$\left.\left.D],\text{Momentum}[p,D]]\right)\right)\Big/\left(2\ \text{Pair}[\text{Momentum}[p,D],\text{Momentum}[p,D]]\right)$$

If you want to avoid direct functions in the output of **TID** and other functions that employ **ToPaVe**, you need to set the option **PaVeToABCD** to **False** globally.

```
SetOptions[ToPaVe, PaVeToABCD -> False];
```

```
TID[FVD[q, mu] FAD[{q, m1}, {q + p}], q, ToPaVe -> True]
```

$$\frac{i\pi^2 p^{\text{mu}}\,\text{A}_0\left(\text{m1}^2\right)}{2p^2}-\frac{i\pi^2\left(\text{m1}^2+p^2\right)p^{\text{mu}}\,\text{B}_0\left(p^2,0,\text{m1}^2\right)}{2p^2}$$

```
% // StandardForm
```

$$\frac{i\pi^2\ \text{Pair}[\text{LorentzIndex}[\text{mu}, D], \text{Momentum}[p, D]]\ \text{PaVe}\left[0, \{\}, \{\text{m1}^2\}\right]}{2\ \text{Pair}[\text{Momentum}[p, D], \text{Momentum}[p, D]]} - \left(i\pi^2\ \text{Pair}[\text{LorentzIndex}[\text{mu}, D], \text{Momentum}[p, D]]\ \left(\text{m1}^2 + \text{Pair}[\text{Momentum}[p, D], \text{Momentum}[p, D]]\right)\ \text{PaVe}\left[0, \{\text{Pair}[\text{Momentum}[p, D], \text{Momentum}[p, D]]\}, \{0, \text{m1}^2\}\right]\right) / \left(2\ \text{Pair}[\text{Momentum}[p, D], \text{Momentum}[p, D]]\right)$$

9.116 ToPaVe2

`ToPaVe2[expr]` converts all the direct Passarino-Veltman functions (**`A0`**, **`A00`**, **`B0`**, **`B1`**, **`B00`**, **`B11`**, **`C0`**, **`D0`**) to **`PaVe`**-functions.

9.116.1 See also

Overview, ToPaVe.

9.116.2 Examples

```
ToPaVe2[A0[m^2]]
```

$$\text{A}_0\left(m^2\right)$$

```
ToPaVe2[A0[m^2]] // FCI // StandardForm

(*PaVe[0, {}, {m^2}]*)
```

```
ToPaVe2[B11[pp, m^2, M^2, BReduce -> False]]
```

$$\text{B}_{11}\left(\text{pp}, m^2, M^2\right)$$

```
ToPaVe2[B11[pp, m^2, M^2, BReduce -> False]] // FCI // StandardForm

(*PaVe[1, 1, {pp}, {m^2, M^2}]*)
```

9.117 ToSFAD

ToSFAD[exp] converts all propagator denominators written as **FAD** or **FeynAmpDenmonina-tor[..., PropagatorDenominator[...], ...]** to **SFAD** or **FeynAmpDenmoninator[..., StandardPropagatorDenominator[...], ...]** respectively.

9.117.1 See also

Overview, FAD, SFAD.

9.117.2 Examples

```
ToSFAD[FAD[p]]
```

$$\frac{1}{(p^2+i\eta)}$$

```
ToSFAD[FAD[p]] // StandardForm

(*FeynAmpDenominator[StandardPropagatorDenominator[Momentum[p, D], 0, 0,
{1, 1}]]*)
```

```
ToSFAD[FAD[{p, m}]]
```

$$\frac{1}{(p^2-m^2+i\eta)}$$

```
ToSFAD[FAD[{p, m}]] // StandardForm

(*FeynAmpDenominator[StandardPropagatorDenominator[Momentum[p, D], 0, -m^2,
{1, 1}]]*)
```

```
ToSFAD[FAD[{p + q, m, 2}]]
```

$$\frac{1}{((p+q)^2-m^2+i\eta)^2}$$

```
ToSFAD[FAD[{p + q, m, 2}]] // StandardForm

(*FeynAmpDenominator[StandardPropagatorDenominator[Momentum[p + q, D], 0,
-m^2, {1, 1}], StandardPropagatorDenominator[Momentum[p + q, D], 0, -m^2,
{1, 1}]]*)
```

9.118 TrickIntegrate

`TrickIntegrate[(1 - t)^(a * Epsilon - 1) g[t], t]` does an integration trick for the definite integral of $((1-t)^{a\ \text{Epsilon}-1}$ g[t]) from **`0`** to **`1`**, yielding **`g[1]/a/Epsilon + Hold[Integrate][(1-t)^{ a Epsilon-1} (g[t]-g[1]), {t, 0, 1}]`**

`TrickIntegrate[t^(a Epsilon-1) g[t], t]` gives $\frac{g[0]}{a\ \text{Epsilon}}$+ Hold[Integrate][$t^{a\ \text{Epsilon}-1}$ (g[t]-g[0]),{t,0,1}], provided g[1] and g[0] exist.

9.118.1 See also

Overview, Epsilon.

9.118.2 Examples

```
TrickIntegrate[(1 - t)^(a Epsilon - 1) g[t], t]
```

$$\text{Hold}[\text{Integrate}]\left[g(t)(1-t)^{a\varepsilon-1}, \{t, 0, 1\}\right]$$

```
TrickIntegrate[t^(a Epsilon - 1) g[t], t]
```

$$\text{Hold}[\text{Integrate}]\left[g(t)t^{a\varepsilon-1}, \{t, 0, 1\}\right]$$

10 Export and import

10.1 FCGVToSymbol

FCGVToSymbol[exp] converts objects of type **FCGV["sth"]** in **exp** to symbols using **ToExpression["sth"]**.

The option **StringReplace** can be used to specify string replacement rules that will take care of special characters (e.g. **^** or **_**) that cannot appear in valid Mathematica expressions. **SMPToSymbol** is useful when exporting FeynCalc expressions to other tools, e.g. FORM.

10.1.1 See also

Overview, FCGV, SMPToSymbol.

10.1.2 Examples

```
FCGV["a"] // FCGVToSymbol
```

```
% // InputForm
```

$$a$$

```
a
```

```
FCGV["$MU"] // FCGVToSymbol
```

```
% // InputForm
```

$MU

```
$MU
```

10.2 FCLoopGLIToSymbol

FCLoopGLIToSymbol[exp] converts **GLI**s to symbols.

The option **Head** determines the prefix of the symbol and can be set to **FCTopology** (default) or **GLI**

The option **Character** specifies the separator between to prefix and the indices.

10.2.1 See also

Overview, GLI, SMPToSymbol, FCGVToSymbol.

10.2.2 Examples

```
FCLoopGLIToSymbol[GLI[topo1, {1, 1, 1, 1, 1}]]
```

topo1X11111

```
FCLoopGLIToSymbol[GLI[topo1, {1, 1, 1, 1, 1}], Head -> GLI]
```

GLIXtopo1X11111

```
FCLoopGLIToSymbol[GLI[topo1, {1, 1, 1, 1, 1}], Character -> "$"]
```

topo1$11111

10.3 FCToTeXReorder

FCToTeXReorder[exp, {{v1, v2, ... }, {a1, a2, ... }, {b1, b2, ... }}] is an auxiliary function that helps to bring the given Mathematica expression **exp** into a form suitable for being inserted into a LaTeX document.

To override the built-in ordering of **Plus** and **Times**, the expression is converted into a nested list made of elements of the form **{a, b, ... , Plus}** or **{a, b, ... , Times}** for a sum or a product respectively.

Then, the option **SortBy** allows to specify two sorting functions that will be used to reorder the terms in both groups.

Most importantly, **FCToTeXReorder** can be applied to the output of a previous function call. This allows for arbitrarily deep nesting.

Finally, you can check if the final result satisfies your expectations by using **FCToTeXPreviewTermOrder**.

10.3.1 See also

Overview, FCToTeXPreviewTermOrder.

10.3.2 Examples

```
exp = (-13629 - 4452*L1 + 24*L2 + 380*NH + 75*L1*NH + 130*NL + 150*L1*NL +
      130*NV + 150*L1*NV + 20*Sqrt[3]*Pi - 75*Sqrt[3]*NH*Pi + 360*Pi^2 +
      66300*z +
      20628*L1*z + 648*L2*z + 450*NL*z + 900*NV*z + 72*Pi^2*z +
      2592*z*Log[z])/81;
```

```
aux1 = FCToTeXReorder[exp, {{z}, {Log, L1, L2}, {Log, L1, L2}}]
```

$$\left\{\left\{\frac{2}{27}z\left(75\ \mathrm{NL}+150\ \mathrm{NV}+12\pi^2+11050\right),\{8z,\mathrm{L2},\mathrm{Times}\},\{32z,\log(z),\mathrm{Times}\},\left\{\frac{764z}{3},\mathrm{L1},\right.\right.\right.$$
$$\left.\left.\left.\mathrm{Times}\right\},\mathrm{Plus}\right\},\left\{\frac{1}{81}\left(-75\sqrt{3}\pi\ \mathrm{NH}+380\ \mathrm{NH}+130\ \mathrm{NL}+130\ \mathrm{NV}+360\pi^2+20\sqrt{3}\pi-13629\right),\right.$$
$$\left.\left\{\frac{8}{27},\mathrm{L2},\mathrm{Times}\right\},\left\{\frac{1}{27}(25\ \mathrm{NH}+50\ \mathrm{NL}+50\ \mathrm{NV}-1484),\mathrm{L1},\mathrm{Times}\right\},\mathrm{Plus}\right\},\mathrm{Plus}\right\}$$

```
aux1 // FCToTeXPreviewTermOrder
```

$$\left(\frac{2}{27}\left(11050+75\ \mathrm{NL}+150\ \mathrm{NV}+12\pi^2\right)z+8z\ \mathrm{L2}+32z\log(z)+\frac{764z\ \mathrm{L1}}{3}\right)$$
$$+\left(\frac{1}{81}\left(-13629+380\ \mathrm{NH}+130\ \mathrm{NL}+130\ \mathrm{NV}+20\sqrt{3}\pi-75\sqrt{3}\ \mathrm{NH}\pi+360\pi^2\right)\right.$$
$$\left.+\frac{8\ \mathrm{L2}}{27}+\frac{1}{27}(-1484+25\ \mathrm{NH}+50\ \mathrm{NL}+50\ \mathrm{NV})\ \mathrm{L1}\right)$$

```
aux1 // InputForm
```

```
{{(2*(11050 + 75*NL + 150*NV + 12*Pi^2)*z)/27, {8*z, L2, Times},
  {32*z, Log[z], Times}, {(764*z)/3, L1, Times}, Plus},
 {(-13629 + 380*NH + 130*NL + 130*NV + 20*Sqrt[3]*Pi -
    75*Sqrt[3]*NH*Pi + 360*Pi^2)/81, {8/27, L2, Times},
  {(-1484 + 25*NH + 50*NL + 50*NV)/27, L1, Times}, Plus}, Plus}
```

```
res = FCToTeXReorder[aux1, {{L1, L2}, {NH, NV, NL}, {NH, NV, NL}}]
```

$$\left\{\left\{\left\{\frac{4}{27}\left(5525+6\pi^2\right)z,\left\{\frac{50z}{9},\text{NL},\text{Times}\right\},\left\{\frac{100z}{9},\text{NV},\text{Times}\right\},\text{Plus}\right\},\{8z,\text{L2},\text{Times}\},\{32z,\right.\right.\right.$$
$$\left.\left.\left.\log(z),\text{Times}\},\left\{\frac{764z}{3},\text{L1},\text{Times}\right\},\text{Plus}\right\},\left\{\left\{\frac{1}{81}\left(-13629+20\sqrt{3}\pi+360\pi^2\right),\left\{\frac{130}{81},\text{NL},\text{Times}\right\},\right.\right.\right.$$
$$\left.\left.\left.\left\{\frac{130}{81},\text{NV},\text{Times}\right\},\left\{\frac{5}{81}\left(76-15\sqrt{3}\pi\right),\text{NH},\text{Times}\right\},\text{Plus}\right\},\left\{\frac{8}{27},\text{L2},\text{Times}\right\},\left\{\left\{-\frac{1484}{27},\right.\right.\right.\right.$$
$$\left.\left.\left.\left.\left\{\frac{25}{27},\text{NH},\text{Times}\right\},\left\{\frac{50}{27},\text{NL},\text{Times}\right\},\left\{\frac{50}{27},\text{NV},\text{Times}\right\},\text{Plus}\right\},\text{L1},\text{Times}\right\},\text{Plus}\right\},\text{Plus}\right\}$$

```
res // FCToTeXPreviewTermOrder
```

$$\left(\left(\frac{4}{27}\left(5525+6\pi^2\right)z+\frac{50z\ \text{NL}}{9}+\frac{100z\ \text{NV}}{9}\right)+8z\ \text{L2}+32z\log(z)+\frac{764z\ \text{L1}}{3}\right)$$
$$+\left(\left(\frac{1}{81}\left(-13629+20\sqrt{3}\pi+360\pi^2\right)+\frac{130\ \text{NL}}{81}+\frac{130\ \text{NV}}{81}+\frac{5}{81}\left(76-15\sqrt{3}\pi\right)\ \text{NH}\right)\right.$$
$$\left.+\frac{8\ \text{L2}}{27}+\left(-\frac{1484}{27}+\frac{25\ \text{NH}}{27}+\frac{50\ \text{NL}}{27}+\frac{50\ \text{NV}}{27}\right)\ \text{L1}\right)$$

```
exp = ((L2*(-5 + nc)*(1 + nc)*(-32*nc - 32*nc^2))/nc^3 + (L1*(1 +
nc)*(672*nc + 256*nc^2 +
          32*nc^3 - 40*nc^2*NH - 80*nc^2*NL - 80*nc^2*NV))/(3*nc^3) + ((1 +
          nc)*(14544*nc +
          7872*nc^2 - 1440*nc^3 - 1216*nc^2*NH - 416*nc^2*NL - 416*nc^2*NV
          - 192*Sqrt[3]*nc*Pi +
          240*Sqrt[3]*nc^2*NH*Pi - 384*nc^3*Pi^2 -
          1440*nc^2*NV*z))/(36*nc^3) +
       ((1 + nc)*(14544*nc + 7872*nc^2 - 1440*nc^3 - 1216*nc^2*NH -
416*nc^2*NL -
           416*nc^2*NV - 192*Sqrt[3]*nc*Pi + 240*Sqrt[3]*nc^2*NH*Pi -
           384*nc^3*Pi^2 +
           11520*nc*z + 15984*nc^2*z + 3312*nc^3*z - 1440*nc^2*NL*z -
           2880*nc^2*NV*z -
           768*nc^3*Pi^2*z))/(36*nc^3))/2
```

$$\frac{1}{2}\left(\frac{\text{L1}(\text{nc}+1)\left(32\ \text{nc}^3-40\ \text{nc}^2\ \text{NH}-80\ \text{nc}^2\ \text{NL}-80\ \text{nc}^2\ \text{NV}+256\ \text{nc}^2+672\ \text{nc}\right)}{3\ \text{nc}^3}\right.$$
$$+\frac{\text{L2}(\text{nc}-5)(\text{nc}+1)\left(-32\ \text{nc}^2-32\ \text{nc}\right)}{\text{nc}^3}+\frac{1}{36\ \text{nc}^3}(\text{nc}+1)\left(-384\pi^2\ \text{nc}^3-1440\ \text{nc}^3-1216\ \text{nc}^2\ \text{NH}\right.$$
$$\left.+240\sqrt{3}\pi\ \text{nc}^2\ \text{NH}-416\ \text{nc}^2\ \text{NL}-1440\ \text{nc}^2\ \text{NV}z-416\ \text{nc}^2\ \text{NV}+7872\ \text{nc}^2-192\sqrt{3}\pi\ \text{nc}+14544\ \text{nc}\right)$$
$$+\frac{1}{36\ \text{nc}^3}(\text{nc}+1)\left(-768\pi^2\ \text{nc}^3z+3312\ \text{nc}^3z-384\pi^2\ \text{nc}^3-1440\ \text{nc}^3-1216\ \text{nc}^2\ \text{NH}\right.$$
$$+240\sqrt{3}\pi\ \text{nc}^2\ \text{NH}-1440\ \text{nc}^2\ \text{NL}z-416\ \text{nc}^2\ \text{NL}-2880\ \text{nc}^2\ \text{NV}z$$
$$\left.\left.-416\ \text{nc}^2\ \text{NV}+15984\ \text{nc}^2z+7872\ \text{nc}^2+11520\ \text{nc}z-192\sqrt{3}\pi\ \text{nc}+14544\ \text{nc}\right)\right)$$

Split into pieces that depend on **L1**, **L2** and those then don' t . Then collect terms in the first group w.r.t **L1, L2** . Collect terms in the second group w.r.t. **z** . Use **ExpandAll** as the factoring function in both groups . Sort the resulting terms in the first group such, that terms containing **L1** come first, then those with **L2** and finally all the rest . Put terms that depend on **z** in the second group first .

```
out1 = FCToTeXReorder[exp, {{L1, L2}, {L1, L2}, {z}}, Split -> True,
Factoring -> {Function[x,
      ExpandAll[x]], Function[x, ExpandAll[x]]}, SortBy -> {Function[x,
      Which[! FreeQ2[x, {L1}], 1,
       ! FreeQ2[x, {L2}], 2, True, 30]], Function[x, Which[! FreeQ2[x,
{z}], 1, True, 3]]}]
```

$$\left\{\left\{\left\{\frac{112}{\text{nc}^2}-\frac{20\ \text{NH}}{3\ \text{nc}}-\frac{40\ \text{NL}}{3\ \text{nc}}-\frac{40\ \text{NV}}{3\ \text{nc}}+\frac{16\ \text{nc}}{3}+\frac{464}{3\ \text{nc}}-\frac{20\ \text{NH}}{3}-\frac{40\ \text{NL}}{3}-\frac{40\ \text{NV}}{3}+48,\right.\right.\right.$$
$$\left.\left.\left.\text{L1}, \text{Times}\right\}, \left\{\frac{80}{\text{nc}^2}-16\ \text{nc}+\frac{144}{\text{nc}}+48, \text{L2}, \text{Times}\right\}, \text{Plus}\right\},$$
$$\left\{\left\{\frac{160}{\text{nc}^2}-\frac{20\ \text{NL}}{\text{nc}}-\frac{60\ \text{NV}}{\text{nc}}-\frac{32\pi^2\ \text{nc}}{3}+46\ \text{nc}+\frac{382}{\text{nc}}-20\ \text{NL}-60\ \text{NV}-\frac{32\pi^2}{3}+268, z,\right.\right.$$
$$\left.\left.\text{Times}\right\}, -\frac{16\pi}{\sqrt{3}\ \text{nc}^2}+\frac{404}{\text{nc}^2}-\frac{304\ \text{NH}}{9\ \text{nc}}+\frac{20\pi\ \text{NH}}{\sqrt{3}\ \text{nc}}-\frac{104\ \text{NL}}{9\ \text{nc}}-\frac{104\ \text{NV}}{9\ \text{nc}}-\frac{32\pi^2\ \text{nc}}{3}-40\ \text{nc}\right.$$
$$\left.-\frac{16\pi}{\sqrt{3}\ \text{nc}}+\frac{1868}{3\ \text{nc}}-\frac{304\ \text{NH}}{9}+\frac{20\pi\ \text{NH}}{\sqrt{3}}-\frac{104\ \text{NL}}{9}-\frac{104\ \text{NV}}{9}-\frac{32\pi^2}{3}+\frac{536}{3}, \text{Plus}\right\}, \text{Plus}\Big\}$$

Now work with the innermost brackets and put terms that contain **z** first . All the other terms should be sorted, such that **NH**, **NV** and **NL** terms appear in this order.

```
out2 = FCToTeXReorder[out1, {{}, {}, {}}, Split -> False, Factoring ->
{Function[x, ExpandAll[x]],
     Function[x, ExpandAll[x]]}, SortBy -> {Function[x, Which[! FreeQ2[x,
     {z}], 1, ! FreeQ2[x, {NH}],
       2, ! FreeQ2[x, {NV}], 3, ! FreeQ2[x, {NL}], 4, True, 5]],
       Function[x, Which[! FreeQ2[x, {z}],
       1, ! FreeQ2[x, {NH}], 2, ! FreeQ2[x, {NV}], 3, ! FreeQ2[x, {NL}], 4,
       True, 5]]}]
```

$$\left\{\left\{\left\{\left\{-\frac{20\ \text{NH}}{3}, -\frac{20\ \text{NH}}{3\ \text{nc}}, -\frac{40\ \text{NV}}{3}, -\frac{40\ \text{NV}}{3\ \text{nc}}, -\frac{40\ \text{NL}}{3}, -\frac{40\ \text{NL}}{3\ \text{nc}}, 48, \frac{112}{\text{nc}^2}, \frac{464}{3\ \text{nc}}, \frac{16\ \text{nc}}{3},\right.\right.\right.\right.$$
$$\left.\left.\left.\left.\text{Plus}\right\}, \text{L1}, \text{Times}\right\}, \left\{\left\{48, \frac{80}{\text{nc}^2}, \frac{144}{\text{nc}}, -16\ \text{nc}, \text{Plus}\right\}, \text{L2}, \text{Times}\right\}, \text{Plus}\right\}, \left\{\left\{\left\{-60\ \text{NV},\right.\right.\right.$$
$$\left.\left.\left.-\frac{60\ \text{NV}}{\text{nc}}, -20\ \text{NL}, -\frac{20\ \text{NL}}{\text{nc}}, 268, \frac{160}{\text{nc}^2}, \frac{382}{\text{nc}}, 46\ \text{nc}, -\frac{32\pi^2}{3}, -\frac{32\pi^2\ \text{nc}}{3}, \text{Plus}\right\}, z, \text{Times}\right\},$$
$$\left\{-\frac{304\ \text{NH}}{9}, -\frac{304\ \text{NH}}{9\ \text{nc}}, \frac{20\pi\ \text{NH}}{\sqrt{3}}, \frac{20\pi\ \text{NH}}{\sqrt{3}\ \text{nc}}, -\frac{104\ \text{NV}}{9}, -\frac{104\ \text{NV}}{9\ \text{nc}}, -\frac{104\ \text{NL}}{9}, -\frac{104\ \text{NL}}{9\ \text{nc}},\right.$$
$$\left.\frac{536}{3}, \frac{404}{\text{nc}^2}, \frac{1868}{3\ \text{nc}}, -40\ \text{nc}, -\frac{16\pi}{\sqrt{3}\ \text{nc}^2}, -\frac{16\pi}{\sqrt{3}\ \text{nc}}, -\frac{32\pi^2}{3}, -\frac{32\pi^2\ \text{nc}}{3}, \text{Plus}\right\}, \text{Plus}\Big\}, \text{Plus}\Big\}$$

```
FCToTeXPreviewTermOrder[out2]
```

$$\left(\left(-\frac{20\ \mathrm{NH}}{3}-\frac{20\ \mathrm{NH}}{3\ \mathrm{nc}}-\frac{40\ \mathrm{NV}}{3}-\frac{40\ \mathrm{NV}}{3\ \mathrm{nc}}-\frac{40\ \mathrm{NL}}{3}-\frac{40\ \mathrm{NL}}{3\ \mathrm{nc}}+48+\frac{112}{\mathrm{nc}^2}+\frac{464}{3\ \mathrm{nc}}+\frac{16\ \mathrm{nc}}{3}\right)\ \mathrm{L1}\right.$$
$$\left.+\left(48+\frac{80}{\mathrm{nc}^2}+\frac{144}{\mathrm{nc}}-16\ \mathrm{nc}\right)\ \mathrm{L2}\right)$$
$$+\left(\left(-60\ \mathrm{NV}-\frac{60\ \mathrm{NV}}{\mathrm{nc}}-20\ \mathrm{NL}-\frac{20\ \mathrm{NL}}{\mathrm{nc}}+268+\frac{160}{\mathrm{nc}^2}+\frac{382}{\mathrm{nc}}+46\ \mathrm{nc}-\frac{32\pi^2}{3}-\frac{32\ \mathrm{nc}\pi^2}{3}\right)z\right.$$
$$+\left(-\frac{304\ \mathrm{NH}}{9}-\frac{304\ \mathrm{NH}}{9\ \mathrm{nc}}+\frac{20\ \mathrm{NH}\pi}{\sqrt{3}}+\frac{20\ \mathrm{NH}\pi}{\sqrt{3}\ \mathrm{nc}}-\frac{104\ \mathrm{NV}}{9}-\frac{104\ \mathrm{NV}}{9\ \mathrm{nc}}-\frac{104\ \mathrm{NL}}{9}\right.$$
$$\left.\left.-\frac{104\ \mathrm{NL}}{9\ \mathrm{nc}}+\frac{536}{3}+\frac{404}{\mathrm{nc}^2}+\frac{1868}{3\ \mathrm{nc}}-40\ \mathrm{nc}-\frac{16\pi}{\sqrt{3}\ \mathrm{nc}^2}-\frac{16\pi}{\sqrt{3}\ \mathrm{nc}}-\frac{32\pi^2}{3}-\frac{32\ \mathrm{nc}\pi^2}{3}\right)\right)$$

10.4 FCToTeXPreviewTermOrder

`FCToTeXPreviewTermOrder[exp]` displays the output of **`FCToTeXReorder`** using the built-in Plus and Times but preserving the original ordering.

Use **`ReleaseHold`** or **`FRH`** to allow Mathematica return to its original ordering.

Notice that the output of **`FCToTeXPreviewTermOrder`** is not suitable for algebraic manipulations but should be understood as an intermediate expression form created to serve as an input for **`TeXForm`**

10.4.1 See also

Overview, FCToTeXReorder.

10.4.2 Examples

```
ex = {(2*z*(11050 + 3438*L1 + 108*L2 + 75*NL + 150*NV + 12*Pi^2 +
432*Log[z]))/27,
   (-13629 - 4452*L1 + 24*L2 + 380*NH + 75*L1*NH + 130*NL + 150*L1*NL +
130*NV +
      150*L1*NV + 20*Sqrt[3]*Pi - 75*Sqrt[3]*NH*Pi + 360*Pi^2)/81, Plus}
```

$$\left\{\frac{2}{27}z\left(3438\ \mathrm{L1}+108\ \mathrm{L2}+75\ \mathrm{NL}+150\ \mathrm{NV}+432\log(z)+12\pi^2+11050\right),\right.$$
$$\frac{1}{81}\left(75\ \mathrm{L1\ NH}+150\ \mathrm{L1\ NL}+150\ \mathrm{L1\ NV}-4452\ \mathrm{L1}+24\ \mathrm{L2}+380\ \mathrm{NH}\right.$$
$$\left.\left.-75\sqrt{3}\pi\ \mathrm{NH}+130\ \mathrm{NL}+130\ \mathrm{NV}+360\pi^2+20\sqrt{3}\pi-13629\right),\mathrm{Plus}\right\}$$

```
FCToTeXPreviewTermOrder[ex]
```

$$\frac{2}{27}z\left(11050+3438\,\mathrm{L1}+108\,\mathrm{L2}+75\,\mathrm{NL}+150\,\mathrm{NV}+12\pi^2+432\log(z)\right)+\frac{1}{81}\left(-13629-4452\,\mathrm{L1}+24\,\mathrm{L2}\right.$$
$$\left.+380\,\mathrm{NH}+75\,\mathrm{L1}\,\mathrm{NH}+130\,\mathrm{NL}+150\,\mathrm{L1}\,\mathrm{NL}+130\,\mathrm{NV}+150\,\mathrm{L1}\,\mathrm{NV}+20\sqrt{3}\pi-75\sqrt{3}\,\mathrm{NH}\pi+360\pi^2\right)$$

10.5 FeynCalc2FORM

`FeynCalc2FORM[exp]` displays **`exp`** in **`FORM`** syntax.

`FeynCalc2FORM[file, x]` writes **`x`** in FORM syntax to a file.

`FeynCalc2FORM[file, x == y]` writes $x = y$ to a file in FORM syntax.

The capabilities of this function are very limited, so you should not expect it to easily handle large and complicated expressions.

10.5.1 See also

Overview, FeynCalc2FORM.

10.5.2 Examples

```
FORM2FeynCalc
```

FORM2FeynCalc

```
MT[\[Mu], \[Nu]] FV[p, \[Rho]] y^2/d

FeynCalc2FORM[%];
```

$$\frac{y^2 \overline{p}^{\rho} \bar{g}^{\mu\nu}}{d}$$

(y^2*d_(mu,nu)*p(ro))/d

```
LC[\[Alpha], \[Beta], \[Delta], \[Rho]]

FeynCalc2FORM[%];
```

$$\bar{\epsilon}^{\alpha\beta\delta\rho}$$

(-i_)*e_(al,be,de,ro)

```
DiracTrace[GA[\[Mu], \[Nu], \[Rho], \[Sigma]]]

FeynCalc2FORM[%];
```

$$\text{tr}\left(\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\sigma}\right)$$

g_(0,mu)*g_(0,nu)*g_(0,ro)*g_(0,si)

```
DiracTrace[GA[\[Mu], \[Nu]]] DiracTrace[GA[\[Mu], \[Rho]]]

FeynCalc2FORM[%];
```

$$\text{tr}\left(\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}\right)\ \text{tr}\left(\bar{\gamma}^{\mu}.\bar{\gamma}^{\rho}\right)$$

g_(0,mu)*g_(0,nu)*g_(1,mu)*g_(1,ro)

```
t = DiracSimplify[DiracTrace[GA[\[Mu], \[Nu], \[Rho], \[Sigma]] . GS[p,
q]]]
```

$$4\overline{p}^{\nu}\overline{q}^{\mu}\bar{g}^{\rho\sigma}-4\overline{p}^{\mu}\overline{q}^{\nu}\bar{g}^{\rho\sigma}-4\overline{p}^{\rho}\overline{q}^{\mu}\bar{g}^{\nu\sigma}+4\overline{p}^{\rho}\overline{q}^{\nu}\bar{g}^{\mu\sigma}+4\overline{p}^{\mu}\overline{q}^{\rho}\bar{g}^{\nu\sigma}-4\overline{p}^{\nu}\overline{q}^{\rho}\bar{g}^{\mu\sigma}+4\overline{p}^{\sigma}\overline{q}^{\mu}\bar{g}^{\nu\rho}-4\overline{p}^{\sigma}\overline{q}^{\nu}\bar{g}^{\mu\rho}$$
$$+4\overline{p}^{\sigma}\overline{q}^{\rho}\bar{g}^{\mu\nu}-4\overline{p}^{\mu}\overline{q}^{\sigma}\bar{g}^{\nu\rho}+4\overline{p}^{\nu}\overline{q}^{\sigma}\bar{g}^{\mu\rho}-4\overline{p}^{\rho}\overline{q}^{\sigma}\bar{g}^{\mu\nu}+4\bar{g}^{\mu\nu}\bar{g}^{\rho\sigma}\left(\overline{p}\cdot\overline{q}\right)+4\bar{g}^{\mu\sigma}\bar{g}^{\nu\rho}\left(\overline{p}\cdot\overline{q}\right)-4\bar{g}^{\mu\rho}\bar{g}^{\nu\sigma}\left(\overline{p}\cdot\overline{q}\right)$$

```
FeynCalc2FORM["fc2ftest.f", L == t];
```

```
TableForm[ReadList[If[$OperatingSystem === "MacOS", ":", ""] <>
"fc2ftest.f", String]]
```

```
Indices \\[Mu],\\[Nu],\\[Rho],\\[Sigma];
Vectors OPEDelta,p,q;
write statistics;
Local L = (
4*d_(mu,si)*d_(nu,ro)*q.p-4*d_(mu,ro)*d_(nu,si)*q.p+4*d_(mu,nu)*d_(ro,si)*q.p+
4*d_(ro,si)*p(nu)*q(mu)-4*d_(nu,si)*p(ro)*q(mu)+4*d_(nu,ro)*p(si)*q(mu)-
4*d_(ro,si)*p(mu)*q(nu)+4*d_(mu,si)*p(ro)*q(nu)-4*d_(mu,ro)*p(si)*q(nu)+
4*d_(nu,si)*p(mu)*q(ro)-4*d_(mu,si)*p(nu)*q(ro)+4*d_(mu,nu)*p(si)*q(ro)-
4*d_(nu,ro)*p(mu)*q(si)+4*d_(mu,ro)*p(nu)*q(si)-4*d_(mu,nu)*p(ro)*q(si) );

print;
.end
```

```
If[FileNames["fc2ftest.f"] =!= {}, DeleteFile["fc2ftest.f"]];
```

```
Clear[t];
```

10.6 FeynCalcToLaTeX

FeynCalcToLaTeX[exp] generates LaTeX with line-breaking for **exp**.

FeynCalcToLaTeX[expr, 500] generates LaTeX for **exp** where **500** is the Window width setting for the Mathematica frontend. Increasing its value will generate less line breaks.

NB: This function appears to be broken in the recent Mathematica versions, most likely it will be rewritten from scratch in a future version of FeynCalc.

10.6.1 See also

Overview, Write2.

10.6.2 Examples

10.7 FORM2FeynCalc

FORM2FeynCalc[exp] translates the FORM expression **exp** into FeynCalc notation.

FORM2FeynCalc[file] translates the FORM expressions in **file** into FeynCalc notation.

FORM2FeynCalc[file, x1, x2, ...] reads in a file in FORM-format and translates the assignments for the variables $a, b, \ldots$ into FeynCalc syntax.

If the option **Set** is **True**, the variables **x1**, **x2** are assigned to the right hand sides defined in the FORM-file.The capabilities of this function are very limited, so that you should not expect it to easily handle large and complicated expressions.

10.7.1 See also

Overview, FeynCalc2FORM.

10.7.2 Examples

```
FORM2FeynCalc["p.q + 2*x m^2"]
```

$$\overline{p} \cdot \overline{q} + 2m^2.x$$

```
FORM2FeynCalc["p.q + 2*x m^2"] // StandardForm

(*2 x . m^2 + SP[p, q]*)
```

Functions are automatically converted in a proper way, but bracketed expressions need to be substituted explicitly.

```
FORM2FeynCalc["x +f(z)+ log(x)^2+[li2(1-x)]", Replace -> {"[li2(1-x)]" ->
"PolyLog[2,1-x]"}]
```

$$f(z)+\text{Li}_2(1-x)+x+\log^2(x)$$

```
FORM2FeynCalc["x +f(z)+ log(x)^2+[li2(1-x)]", Replace -> {"[li2(1-x)]" ->
"PolyLog[2,1-x]"}] // StandardForm

(*x + f[z] + Log[x]^2 + PolyLog[2, 1 - x]*)
```

```
FORM2FeynCalc["x + [(1)]*y -[(-1)^m]"]
```

$$-\text{Hold}\left[(-1)^m\right]+\text{Hold}[1].y+x$$

```
ReleaseHold[FORM2FeynCalc["x + [(1)]*y -[(-1)^m]"]]
```

$$-(-1)^m+x+1.y$$

```
FORM2FeynCalc["p(mu)*q(nu)+d_(mu,nu)"]
```

$$\bar{g}^{\text{mu nu}}+p(\text{mu}).q(\text{nu})$$

```
FORM2FeynCalc["p(mu)*q(nu)+d_(mu,nu)"] // StandardForm

(*p[mu] . q[nu] + MT[mu, nu]*)
```

```
FORM2FeynCalc["p(mu)*q(nu)+d_(mu,nu)", Replace -> {mu -> \[Mu], nu ->
\[Nu]}]
```

$$\bar{g}^{\mu\nu}+p(\mu).q(\nu)$$

```
FORM2FeynCalc["i_*az*bz*aM^2*D1*[(1)]*b_G1 * (
4*eperp(mu,nu)*avec.bvec*blam )"]
```

$$(4i).\text{az.bz.aM}^2.\text{D1.Hold}[1].\text{b\$G1.eperp}(\text{mu},\text{nu}).\left(\overline{\text{avec}}\cdot\overline{\text{bvec}}\right).\text{blam}$$

10.8 StringChomp

`StringChomp[str]` chops initial and final white space of the string **`str`**.

10.8.1 See also

Overview, Write2.

10.8.2 Examples

```
StringChomp[" abc "]

% // InputForm
```

abc

```
"abc"
```

10.9 SMPToSymbol

`SMPToSymbol[exp]` converts objects of type **`SMP["sth"]`** in **`exp`** to symbols using **`ToExpression["sth"]`**.

The option **`StringReplace`** can be used to specify string replacement rules that will take care of special characters (e.g. **`^`** or **`_`**) that cannot appear in valid Mathematica expressions. **`SMPToSymbol`** is useful when exporting FeynCalc expressions to other tools, e.g. FORM.

10.9.1 See also

Overview, SMP, FCGVToSymbol.

10.9.2 Examples

```
SP[p] - SMP["m_e"]^2

SMPToSymbol[%]
```

$$\overline{p}^2 - m_e^2$$

$$\overline{p}^2 - \text{me}^2$$

10.10 Write2

Write2[file, val1 = expr1, val2 = expr2, ...] writes the settings **val1 = expr1, val2 = expr2** in sequence followed by a newline, to the specified output file. Setting the option **FormatType** of **Write2** to **FortranForm** results in Fortran syntax output.

10.10.1 See also

Overview, Isolate, PaVeReduce.

10.10.2 Examples

```
FullForm[$FortranContinuationCharacter]
```

$$\&$$

```
t = Collect[((a - c)^2 + (a - b)^2)^2, a, Factor]
```

$$4a^4 - 8a^3(b+c) + 8a^2\left(b^2+bc+c^2\right) - 4a(b+c)\left(b^2+c^2\right) + \left(b^2+c^2\right)^2$$

This writes the assignment r=t to a file.

```
tempfilename = ToString[$SessionID] <> ".s";

Write2[tempfilename, r = t];
```

This shows the contents of the file.

```
TableForm[ReadList[If[$OperatingSystem === "MacOS", ":", ""] <>
tempfilename, String]]
```

```
r = ( 4*a^4 - 8*a^3*(b + c) - 4*a*(b + c)*(b^2 + c^2) +
 (b^2 + c^2)^2 + 8*a^2*(b^2 + b*c + c^2)
 );
```

```
DeleteFile[If[$OperatingSystem === "MacOS", ":", ""] <> tempfilename]
```

```
t2 = x + Isolate[t, a, IsolateNames -> w]
```

$$4a^4 - 8a^3w(19) + 8a^2w(21) - 4aw(19)w(20) + w(20)^2 + x$$

```
Write2[tempfilename, r = t2];
```

```
TableForm[ReadList[If[$OperatingSystem === "MacOS", ":", ""] <>
tempfilename, String]]
```

```
w[19] = (b + c
 );
w[20] = (b^2 + c^2
 );
w[21] = (b^2 + b*c + c^2
 );
r = ( 4*a^4 + x - 8*a^3*HoldForm[w[19]] - 4*a*HoldForm[w[19]]*
 HoldForm[w[20]] + HoldForm[w[20]]^2 + 8*a^2*HoldForm[w[21]]
 );
```

```
DeleteFile[If[$OperatingSystem === "MacOS", ":", ""] <> tempfilename]
```

This is how to write out the expression **t2** in Fortran format.

```
Write2[tempfilename, r = t2, FormatType -> FortranForm];
```

```
TableForm[ReadList[If[$OperatingSystem === "MacOS", ":", ""] <>
tempfilename, String]]
```

```
      w(19)= b + c
      w(20)= b**2 + c**2
      w(21)= b**2 + b*c + c**2
      r = x + a**4*4D0 - a**3*8D0*w(19) - a*4D0*w(19)*w(20) +
     & w(20)**2 + a**2*8D0*w(21)
```

```
DeleteFile[If[$OperatingSystem === "MacOS", ":", ""] <> tempfilename];

Clear[w, t, t2, r, tempfilename];
```

11 Feynman rules and amplitudes

11.1 BackgroundGluonVertex

BackgroundGluonVertex[{p, mu, a}, {q, nu, b}, {k, la, c}] yields the 3-gluon vertex in the background field gauge, where the first set of arguments corresponds to the external background field. **BackgroundGluonVertex[{p, mu, a}, {q, nu, b}, {k, la, c}, {s, si, d}]** yields the 4-gluon vertex, with **{p, mu , a}** and **{k, la, c}** denoting the external background fields.

The gauge, dimension and the name of the coupling constant are determined by the options **Gauge**, **Dimension** and **CouplingConstant**.

The Feynman rules are taken from L. Abbot NPB 185 (1981), 189-203; except that all momenta are incoming. Note that Abbot's coupling constant convention is consistent with the default setting of **GluonVertex**.

11.1.1 See also

Overview

11.1.2 Examples

```
BackgroundGluonVertex[{p, \[Mu], a}, {q, \[Nu], b}, {k, \[Lambda], c}]
```

$$g_s f^{abc}\left(g^{\mu\nu}(-k+p-q)^{\lambda}+g^{\lambda\mu}(k-p+q)^{\nu}+g^{\lambda\nu}(q-k)^{\mu}\right)$$

```
BackgroundGluonVertex[{p, \[Mu], a}, {q, \[Nu], b}, {k, \[Lambda], c}, {s,
\[Sigma], d}]
```

$$-ig_s^2\left(f^{ad\text{FCGV(u19)}}f^{bc\text{FCGV(u19)}}\left(g^{\lambda\sigma}g^{\mu\nu}-g^{\lambda\nu}g^{\mu\sigma}-g^{\lambda\mu}g^{\nu\sigma}\right)\right.$$
$$\left.+f^{ac\text{FCGV(u19)}}f^{bd\text{FCGV(u19)}}\left(g^{\lambda\sigma}g^{\mu\nu}-g^{\lambda\nu}g^{\mu\sigma}\right)+f^{ab\text{FCGV(u19)}}f^{cd\text{FCGV(u19)}}\left(g^{\lambda\sigma}g^{\mu\nu}-g^{\lambda\nu}g^{\mu\sigma}+g^{\lambda\mu}g^{\nu\sigma}\right)\right)$$

```
BackgroundGluonVertex[{p, \[Mu], a}, {q, \[Nu], b}, {k, \[Lambda], c},Gauge
-> \[Alpha]]
```

$$g_s f^{abc}\left(g^{\mu\nu}\left(-\frac{k}{\alpha}+p-q\right)^{\lambda}+g^{\lambda\mu}\left(k-p+\frac{q}{\alpha}\right)^{\nu}+g^{\lambda\nu}(q-k)^{\mu}\right)$$

```
BackgroundGluonVertex[{p, \[Mu], a}, {q, \[Nu], b}, {k, \[Lambda], c}, {s,
\[Sigma], d}, Gauge -> \[Alpha]]
```

$$-ig_s^2 \left(f^{adFCGV(u20)} f^{bcFCGV(u20)} \left(-\frac{g^{\lambda\nu} g^{\mu\sigma}}{\alpha} + g^{\lambda\sigma} g^{\mu\nu} - g^{\lambda\mu} g^{\nu\sigma} \right) + f^{abFCGV(u20)} f^{cdFCGV(u20)} \left(\frac{g^{\lambda\sigma} g^{\mu\nu}}{\alpha} - g^{\lambda\nu} g^{\mu\sigma} + g^{\lambda\mu} g^{\nu\sigma} \right) + f^{acFCGV(u20)} f^{bdFCGV(u20)} \left(g^{\lambda\sigma} g^{\mu\nu} - g^{\lambda\nu} g^{\mu\sigma} \right) \right)$$

11.2 ComplexConjugate

`ComplexConjugate[exp]` returns the complex conjugate of **`exp`**, where the input expression must be a proper matrix element. All Dirac matrices are assumed to be inside closed Dirac spinor chains. If this is not the case, the result will be inconsistent. Denominators may not contain explicit i's.

11.2.1 See also

Overview, FCRenameDummyIndices, FermionSpinSum, DiracGamma.

11.2.2 Examples

ComplexConjugate is meant to be applied to amplitudes, i.e. given a matrix element $\mathcal{M}$, it will return $\mathcal{M}^*$.

```
amp = (Spinor[Momentum[k1], SMP["m_e"], 1] . GA[\[Mu]] .
Spinor[Momentum[p2], SMP["m_e"], 1]*
     Spinor[Momentum[k2], SMP["m_e"], 1] . GA[\[Nu]] . Spinor[Momentum[p1],
SMP["m_e"], 1]*
     FAD[k1 - p2, Dimension -> 4]*SMP["e"]^2 - Spinor[Momentum[k1],
SMP["m_e"],
       1] . GA[\[Mu]] . Spinor[Momentum[p1], SMP["m_e"],
       1]*Spinor[Momentum[k2],
       SMP["m_e"], 1] . GA[\[Nu]] . Spinor[Momentum[p2], SMP["m_e"],
1]*FAD[k2 - p2,
      Dimension -> 4]*SMP["e"]^2)
```

$$\frac{e^2 \left(\varphi(\overline{k1}, m_e)\right) . \bar{\gamma}^{\mu} . \left(\varphi(\overline{p2}, m_e)\right) \left(\varphi(\overline{k2}, m_e)\right) . \bar{\gamma}^{\nu} . \left(\varphi(\overline{p1}, m_e)\right)}{(\overline{k1} - \overline{p2})^2} - \frac{e^2 \left(\varphi(\overline{k1}, m_e)\right) . \bar{\gamma}^{\mu} . \left(\varphi(\overline{p1}, m_e)\right) \left(\varphi(\overline{k2}, m_e)\right) . \bar{\gamma}^{\nu} . \left(\varphi(\overline{p2}, m_e)\right)}{(\overline{k2} - \overline{p2})^2}$$

```
ComplexConjugate[amp]
```

$$\frac{e^2\,(\varphi(\overline{p2},m_e))\,.\bar{\gamma}^{\mu}.\left(\varphi(\overline{k1},m_e)\right)\,(\varphi(\overline{p1},m_e))\,.\bar{\gamma}^{\nu}.\left(\varphi(\overline{k2},m_e)\right)}{(\overline{k1}-\overline{p2})^2}$$
$$-\frac{e^2\,(\varphi(\overline{p1},m_e))\,.\bar{\gamma}^{\mu}.\left(\varphi(\overline{k1},m_e)\right)\,(\varphi(\overline{p2},m_e))\,.\bar{\gamma}^{\nu}.\left(\varphi(\overline{k2},m_e)\right)}{(\overline{k2}-\overline{p2})^2}$$

Although one can also apply the function to standalone Dirac matrices, it should be understood that the result is not equivalent to the complex conjugation of such matrices.

```
GA[\[Mu]]

ComplexConjugate[%]
```

$$\bar{\gamma}^{\mu}$$

$$\bar{\gamma}^{\mu}$$

```
GA[5]

ComplexConjugate[%]
```

$$\bar{\gamma}^{5}$$

$$-\bar{\gamma}^{5}$$

```
(GS[Polarization[k1, -I, Transversality -> True]] . (GS[k1 - p2] +
SMP["m_e"]) .
    GS[Polarization[k2, -I, Transversality -> True]])

ComplexConjugate[%]
```

$$(\bar{\gamma}\cdot\bar{\varepsilon}^{*}(k1))\,.\left(\bar{\gamma}\cdot\left(\overline{k1}-\overline{p2}\right)+m_e\right).(\bar{\gamma}\cdot\bar{\varepsilon}^{*}(k2))$$

$$(\bar{\gamma}\cdot\bar{\varepsilon}(k2))\,.\left(\bar{\gamma}\cdot\left(\overline{k1}-\overline{p2}\right)+m_e\right).(\bar{\gamma}\cdot\bar{\varepsilon}(k1))$$

```
SUNTrace[SUNT[a, b, c]]

ComplexConjugate[%]
```

$$\text{tr}(T^a.T^b.T^c)$$

$$\text{tr}(T^c.T^b.T^a)$$

Since FeynCalc 9.3 **ComplexConjugate** will automatically rename dummy indices.

```
PolarizationVector[p1, \[Mu]] PolarizationVector[p2, \[Nu]] MT[\[Mu],
\[Nu]]

ComplexConjugate[%]
```

$$\bar{g}^{\mu\nu}\bar{\varepsilon}^{\mu}(\text{p1})\bar{\varepsilon}^{\nu}(\text{p2})$$

$$\bar{g}^{\$\text{AL}(\$19)\$\text{AL}(\$20)}\bar{\varepsilon}^{*\$\text{AL}(\$19)}(\text{p1})\bar{\varepsilon}^{*\$\text{AL}(\$20)}(\text{p2})$$

```
GA[\[Mu], \[Nu]] LC[\[Mu], \[Nu]][p1, p2]

ComplexConjugate[%]
```

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}\bar{\epsilon}^{\mu\nu\overline{\text{p1}}\,\overline{\text{p2}}}$$

$$\bar{\gamma}^{\$\text{AL}(\$21)}.\bar{\gamma}^{\$\text{AL}(\$22)}\bar{\epsilon}^{\$\text{AL}(\$22)\$\text{AL}(\$21)\overline{\text{p1}}\,\overline{\text{p2}}}$$

This behavior can be disabled by setting the option **FCRenameDummyIndices** to **False**.

```
ComplexConjugate[GA[\[Mu], \[Nu]] LC[\[Mu], \[Nu]][p1, p2],
FCRenameDummyIndices -> False]
```

$$\bar{\gamma}^{\nu}.\bar{\gamma}^{\mu}\bar{\epsilon}^{\mu\nu\overline{\text{p1}}\,\overline{\text{p2}}}$$

If particular variables must be replaced with their conjugate values, use the option **Conjugate**.

```
GA[\[Mu]] . (c1 GA[6] + c2 GA[7]) . GA[\[Nu]]

ComplexConjugate[%]
```

$$\bar{\gamma}^{\mu}.\left(\text{c1}\,\bar{\gamma}^{6}+\text{c2}\,\bar{\gamma}^{7}\right).\bar{\gamma}^{\nu}$$

$$\bar{\gamma}^{\nu}.\left(\text{c1}\,\bar{\gamma}^{7}+\text{c2}\,\bar{\gamma}^{6}\right).\bar{\gamma}^{\mu}$$

```
ex = ComplexConjugate[GA[\[Mu]] . (c1 GA[6] + c2 GA[7]) . GA[\[Nu]],
Conjugate -> {c1, c2}]
```

$$\bar{\gamma}^{\nu}.\left(\bar{\gamma}^{7}\,\text{c1}^{*}+\bar{\gamma}^{6}\,\text{c2}^{*}\right).\bar{\gamma}^{\mu}$$

```
ex // StandardForm

(*DiracGamma[LorentzIndex[\[Nu]]] . (Conjugate[c2] DiracGamma[6] +
Conjugate[c1] DiracGamma[7]) . DiracGamma[LorentzIndex[\[Mu]]]*)
```

It may happen that one needs to deal with amplitudes with amputated spinors, i.e. with open Dirac or Pauli indices. If the amplitude contains only a single chain of Dirac/Pauli matrices, everything remains unambiguous and the missing spinors are understood

```
GA[\[Mu], \[Nu], \[Rho], 5] CSI[i, j]

ComplexConjugate[%]
```

$$\overline{\sigma}^{i}.\overline{\sigma}^{j}\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{5}$$

$$-\overline{\sigma}^{j}.\overline{\sigma}^{i}\bar{\gamma}^{5}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\mu}$$

However, when there are at least two spinor chains of the same type involved, such expressions do not make sense anymore. In these cases one should introduce explicit spinor indices to avoid ambiguities

```
DCHN[GA[\[Mu], \[Nu], \[Rho], 5], i, j] DCHN[GA[\[Mu], \[Nu], \[Rho], 5],
k, l]

ComplexConjugate[%]
```

$$\left(\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{5}\right)_{ij}\left(\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{5}\right)_{kl}$$

$$\left(\bar{\gamma}^{5}.\bar{\gamma}^{\text{\$AL(\$23)}}.\bar{\gamma}^{\text{\$AL(\$24)}}.\bar{\gamma}^{\text{\$AL(\$25)}}\right)_{ji}\left(\bar{\gamma}^{5}.\bar{\gamma}^{\text{\$AL(\$23)}}.\bar{\gamma}^{\text{\$AL(\$24)}}.\bar{\gamma}^{\text{\$AL(\$25)}}\right)_{lk}$$

```
PCHN[CSI[i, j, k], a, b] PCHN[CSI[i, j, k], c, d]

ComplexConjugate[%]
```

$$\left(\overline{\sigma}^{i}.\overline{\sigma}^{j}.\overline{\sigma}^{k}\right)_{ab}\left(\overline{\sigma}^{i}.\overline{\sigma}^{j}.\overline{\sigma}^{k}\right)_{cd}$$

$$\left(\overline{\sigma}^{\text{\$AL(\$26)}}.\overline{\sigma}^{\text{\$AL(\$27)}}.\overline{\sigma}^{\text{\$AL(\$28)}}\right)_{ba}\left(\overline{\sigma}^{\text{\$AL(\$26)}}.\overline{\sigma}^{\text{\$AL(\$27)}}.\overline{\sigma}^{\text{\$AL(\$28)}}\right)_{dc}$$

The function does not apply **Conjugate** to symbols that do not depend on **I** and are unrelated to Dirac/Pauli/Color matrices. One can specify symbols that need to be explicitly conjugated using the **Conjugate** option

```
cc SpinorU[p1] . GA[mu] . SpinorV[p2]

ComplexConjugate[%]
```

$$\text{cc}u(\text{p1}).\bar{\gamma}^{\text{mu}}.v(\text{p2})$$

$$\text{cc}\left(\varphi(-\overline{\text{p2}})\right).\bar{\gamma}^{\text{mu}}.\left(\varphi(\overline{\text{p1}})\right)$$

```
cc SpinorU[p1] . GA[mu] . SpinorV[p2]

ComplexConjugate[%, Conjugate -> {cc}]
```

$$\text{cc}u(\text{p1}).\bar{\gamma}^{\text{mu}}.v(\text{p2})$$

$$\text{cc}^{*}\left(\varphi(-\overline{\text{p2}})\right).\bar{\gamma}^{\text{mu}}.\left(\varphi(\overline{\text{p1}})\right)$$

11.3 CovariantD

CovariantD[mu] is a generic covariant derivative with Lorentz index μ.

CovariantD[x, mu] is a generic covariant derivative with respect to x^μ.

CovariantD[mu, a, b] is a covariant derivative for a bosonic field that acts on **QuantumField[f, { }, {a, b}]**, where **f** is some field name and **a** and **b** are two $SU(N)$ indices in the adjoint representation.

CovariantD[OPEDelta, a, b] is a short form for **CovariantD[mu, a, b] FV[OPEDelta, mu]**.

CovariantD[{OPEDelta, a, b}, {n}] yields the product of **n** operators, where **n** is an integer.

CovariantD[OPEDelta, a, b, {m, n}] gives the expanded form of **CovariantD[OPEDelta, a, b]^m** up to order g^n for the gluon, where n is an integer and g the coupling constant indicated by the setting of the option **CouplingConstant**.

CovariantD[OPEDelta, {m, n}] gives the expanded form of **CovariantD[OPEDelta]^m** up to order g^n of the fermionic field. To obtain the explicit expression for a particular covariant derivative, the option **Explicit** must be set to **True**.

11.3.1 See also

Overview

11.3.2 Examples

```
CovariantD[\[Mu]]
```

$$D_\mu$$

```
CovariantD[\[Mu], a, b]
```

$$D_\mu^{ab}$$

```
CovariantD[\[Mu], Explicit -> True]
```

$$\vec{\partial}_\mu - i g_s T^{\text{c19}} . A_\mu^{\text{c19}}$$

The first argument of **CovariantD** is interpreted as type **LorentzIndex**, except for **OPEDelta**, which is type **Momentum**.

```
CovariantD[OPEDelta]
```

$$D_{\Delta}$$

```
CovariantD[OPEDelta, a, b]
```

$$D_{\Delta}^{ab}$$

```
CovariantD[OPEDelta, a, b, Explicit -> True]
```

$$\delta^{ab}\vec{\partial}_{\Delta} - g_s A_{\Delta}^{\text{c20}} f^{ab\text{c20}}$$

```
CovariantD[OPEDelta, Explicit -> True]
```

$$\vec{\partial}_{\Delta} - i g_s T^{\text{c21}}.A_{\Delta}^{\text{c21}}$$

```
CovariantD[OPEDelta, a, b, {2}]
```

$$\left(\delta^{ac22}\vec{\partial}_{\Delta} - g_s A_{\Delta}^{\text{e23}} f^{ac22\ \text{e23}}\right).\left(\delta^{bc22}\vec{\partial}_{\Delta} - g_s A_{\Delta}^{\text{e24}} f^{\text{c22}b\text{e24}}\right)$$

This gives $m * \vec{\partial}_{\Delta}$, the partial derivative $\vec{\partial}_{\mu}$ contracted with Δ^{μ}

```
CovariantD[OPEDelta, a, b, {OPEm, 0}]
```

$$\delta^{ab}\left(\vec{\partial}_{\Delta}\right)^{m}$$

The expansion up to first order in the coupling constant g_s (the sum is the **FeynCalcOPESum**)

```
CovariantD[OPEDelta, a, b, {OPEm, 1}]
```

$$\delta^{ab}\left(\vec{\partial}_{\Delta}\right)^{m} - g_s\left(\sum_{i=0}^{-1+m}\left(\vec{\partial}_{\Delta}\right)^{i}.A_{\Delta}^{\text{c34}_1}.\left(\vec{\partial}_{\Delta}\right)^{-1-i+m} f^{ab\text{c34}_1}\right)$$

The expansion up to second order in the g_s

```
CovariantD[OPEDelta, a, b, {OPEm, 2}]
```

$$-g_s\left(\sum_{i=0}^{-1+m}\left(\vec{\partial}_\Delta\right)^{i}.A_\Delta^{\text{c42}_1}.\left(\vec{\partial}_\Delta\right)^{-1-i+m}f^{abc42_1}\right)$$
$$-g_s^2\left(\sum_{j=0}^{-2+m}\left(\sum_{i=0}^{j}\left(\vec{\partial}_\Delta\right)^{i}.A_\Delta^{\text{c46}_1}.\left(\vec{\partial}_\Delta\right)^{-i+j}.A_\Delta^{\text{c46}_2}.\left(\vec{\partial}_\Delta\right)^{-2-j+m}f^{ac46_1e45_1}f^{bc46_2e45_1}\right)\right)+\delta^{ab}\left(\vec{\partial}_\Delta\right)^{m}$$

```
CovariantD[OPEDelta, a, b]^OPEm
```

$$\left(D_\Delta^{ab}\right)^{m}$$

```
CovariantD[OPEDelta, {OPEm, 2}]
```

$$-ig_s\left(\sum_{i=0}^{-1+m}\left(\vec{\partial}_\Delta\right)^{i}.A_\Delta^{\text{c55}_1}.\left(\vec{\partial}_\Delta\right)^{-1-i+m}T^{\text{c55}_1}\right)$$
$$-g_s^2\left(\sum_{j=0}^{-2+m}\left(\sum_{i=0}^{j}T^{\text{c59}_1}.T^{\text{c59}_2}\left(\vec{\partial}_\Delta\right)^{i}.A_\Delta^{\text{c59}_1}.\left(\vec{\partial}_\Delta\right)^{-i+j}.A_\Delta^{\text{c59}_2}.\left(\vec{\partial}_\Delta\right)^{-2-j+m}\right)\right)+\left(\vec{\partial}_\Delta\right)^{m}$$

```
CovariantD[OPEDelta, Explicit -> True] // StandardForm

(*RightPartialD[Momentum[OPEDelta]] - I SUNT[SUNIndex[c62]] .
QuantumField[GaugeField, Momentum[OPEDelta], SUNIndex[c62]] SMP["g_s"]*)
```

```
CovariantD[\[Mu], a, b, Explicit -> True] // StandardForm

(*RightPartialD[LorentzIndex[\[Mu]]] SUNDelta[a, b] -
QuantumField[GaugeField, LorentzIndex[\[Mu]], SUNIndex[c63]] SMP["g_s"]
SUNF[a, b, c63]*)
```

`CovariantFieldDerivative[f[x], x, {li1, li2, ...}]` is a covariant derivative of **`f[x]`** with respect to space-time variables **`x`** and with Lorentz indices **`li1, li2, ...`**. **`CovariantFieldDerivative`** has only typesetting definitions by default. The user is must supply his/her own definition of the actual function.

11.3.3 See also

Overview, CovariantD, ExpandPartialD, FieldDerivative.

11.3.4 Examples

```
CovariantFieldDerivative[QuantumField[A, {\[Mu]}][x], x, {\[Mu]}]
```

$$\mathcal{D}_\mu \left(A_\mu(x)\right)$$

11.4 CDr

CDr is the shorthand notation for **CovariantFieldDerivative**.

11.4.1 See also

Overview, CovariantFieldDerivative.

11.4.2 Examples

11.5 DoPolarizationSums

DoPolarizationSums[exp, k, ...] acts on an expression **exp** that must contain a polarization vector $\varepsilon(k)$ and its complex conjugate (e.g. **exp** can be a matrix element squared).

Depending on the arguments of the function, it will perform a sum over the polarization of $\varepsilon(k)$ and its c.c.

- **DoPolarizationSums[exp, k]** sums over the three physical polarizations of an external massive vector boson with the 4-momentum **k** and the mass k^2.
- **DoPolarizationSums[exp, k, 0]** replaces the polarization sum of an external massless vector boson with the momentum **k** by $-g^{\mu\nu}$. This corresponds to the summation over all 4 polarizations, including the unphysical ones.
- **DoPolarizationSums[exp, k, n]** sums over physical (transverse) polarizations of an external massless vector boson with the momentum **k**, where **n** is an auxiliary 4-vector from the gauge-dependent polarization sum formula.

Cf. **PolarizationSum** for more examples and explanations on different polarizations.

DoPolarizationSums also work with D-dimensional amplitudes.

11.5.1 See also

Overview, Polarization, PolarizationSum, NumberOfPolarizations, VirtualBoson, Uncontract.

11.5.2 Examples

The standard formula for massless vector bosons is valid for all types of the corresponding particles, including gluons.

```
FCClearScalarProducts[]

SP[p] = 0;

Pair[LorentzIndex[\[Mu]], Momentum[Polarization[p, -I]]]
Pair[LorentzIndex[\[Nu]],
   Momentum[Polarization[p, I]]]
```

$$\bar{\varepsilon}^{*\mu}(p)\bar{\varepsilon}^{\nu}(p)$$

```
DoPolarizationSums[%, p, n]
```

$$-\frac{\overline{n}^2\overline{p}^{\mu}\overline{p}^{\nu}}{(\overline{n}\cdot\overline{p})^2}-\bar{g}^{\mu\nu}+\frac{\overline{n}^{\nu}\overline{p}^{\mu}}{\overline{n}\cdot\overline{p}}+\frac{\overline{n}^{\mu}\overline{p}^{\nu}}{\overline{n}\cdot\overline{p}}$$

In QED the gauge invariance ensures the cancellation of the unphysical polarizations so that for photons one can also employ the simpler replacement with the metric tensor.

```
FCClearScalarProducts[]

SP[p] = 0;

Pair[LorentzIndex[\[Mu]], Momentum[Polarization[p, -I]]]
Pair[LorentzIndex[\[Nu]],
   Momentum[Polarization[p, I]]]
```

$$\bar{\varepsilon}^{*\mu}(p)\bar{\varepsilon}^{\nu}(p)$$

```
DoPolarizationSums[%, p, 0]
```

$$-\bar{g}^{\mu\nu}$$

You can also use this trick in QCD, provided that the unphysical degrees of freedom are subtracted using ghosts at a later stage.

Notice that in this case you should not make the polarization vectors transverse using the **`Transversality`** option.

Furthermore, the averaging over the polarizations of the initial gluons must be done on the physical amplitude squared, i.e. after the ghost contributions have been subtracted.

Massive vector bosons (e.g. W or Z) have 3 degrees of freedom and require no auxiliary vector.

```
FCClearScalarProducts[]

SP[p] = m^2;

Pair[LorentzIndex[\[Mu]], Momentum[Polarization[p, -I]]]
Pair[LorentzIndex[\[Nu]],
   Momentum[Polarization[p, I]]]
```

$$\bar{\varepsilon}^{*\mu}(p)\bar{\varepsilon}^{\nu}(p)$$

```
DoPolarizationSums[%, p]
```

$$\frac{\overline{p}^{\mu}\overline{p}^{\nu}}{m^2} - \bar{g}^{\mu\nu}$$

A more realistic example of summing over the polarizations of the photons in $e^+e^{\to}\gamma\gamma$

```
ClearAll[s, t, u];

FCClearScalarProducts[];

SP[k1] = 0;

SP[k2] = 0;

amp = (-((Spinor[Momentum[p1], 0, 1] . GS[Polarization[k1, I,
              Transversality -> True]] . GS[k2 - p2] . GS[Polarization[k2,
I,
              Transversality -> True]] . Spinor[-Momentum[p2], 0,
1]*SMP["e"]^2)/t) -
     (Spinor[Momentum[p1], 0, 1] . GS[Polarization[k2, I,
           Transversality -> True]] . GS[k1 - p2] . GS[Polarization[k1, I,
           Transversality -> True]] . Spinor[-Momentum[p2], 0,
1]*SMP["e"]^2)/u)*
   (-((Spinor[-Momentum[p2], 0, 1] . GS[Polarization[k1, -I,
              Transversality -> True]] . GS[k1 - p2] . GS[Polarization[k2,
-I,
              Transversality -> True]] . Spinor[Momentum[p1], 0, 1]*
           SMP["e"]^2)/u) - (Spinor[-Momentum[p2], 0, 1] .
GS[Polarization[k2, -I,
           Transversality -> True]] . GS[k2 - p2] . GS[Polarization[k1, -I,
           Transversality -> True]] . Spinor[Momentum[p1], 0,
1]*SMP["e"]^2)/t)
```

$$\left(-\frac{e^2\,(\varphi(\overline{p1}))\,.\,(\bar{\gamma}\cdot\bar{\varepsilon}(k1))\,.\,\left(\bar{\gamma}\cdot\left(\overline{k2}-\overline{p2}\right)\right)\,.\,(\bar{\gamma}\cdot\bar{\varepsilon}(k2))\,.\,(\varphi(-\overline{p2}))}{t}\right.$$
$$\left.-\frac{e^2\,(\varphi(\overline{p1}))\,.\,(\bar{\gamma}\cdot\bar{\varepsilon}(k2))\,.\,\left(\bar{\gamma}\cdot\left(\overline{k1}-\overline{p2}\right)\right)\,.\,(\bar{\gamma}\cdot\bar{\varepsilon}(k1))\,.\,(\varphi(-\overline{p2}))}{u}\right)\left(-\frac{e^2\,(\varphi(-\overline{p2}))\,.\,(\bar{\gamma}\cdot\bar{\varepsilon}^*(k2))\,.\,\left(\bar{\gamma}\cdot\left(\overline{k2}-\overline{p2}\right)\right)\,.\,(\bar{\gamma}\right.}{t}$$
$$\left.-\frac{e^2\,(\varphi(-\overline{p2}))\,.\,(\bar{\gamma}\cdot\bar{\varepsilon}^*(k1))\,.\,\left(\bar{\gamma}\cdot\left(\overline{k1}-\overline{p2}\right)\right)\,.\,(\bar{\gamma}\cdot\bar{\varepsilon}^*(k2))\,.\,(\varphi(\overline{p1}))}{u}\right)$$

```
amp // DoPolarizationSums[#, k1, 0] & // DoPolarizationSums[#, k2, 0] &
```

$$\frac{e^4\,(\varphi(-\overline{p2}))\,.\bar{\gamma}^{\$MU(\$27)}.\left(\bar{\gamma}\cdot\left(\overline{k1}-\overline{p2}\right)\right).\bar{\gamma}^{\$MU(\$29)}.\,(\varphi(\overline{p1}))\,(\varphi(\overline{p1}))\,.\bar{\gamma}^{\$MU(\$27)}.\left(\bar{\gamma}\cdot\left(\overline{k2}-\overline{p2}\right)\right).\bar{\gamma}^{\$MU(\$29)}.\,(\varphi(-\overline{p2}))}{tu}$$
$$+\frac{e^4\,(\varphi(\overline{p1}))\,.\bar{\gamma}^{\$MU(\$29)}.\left(\bar{\gamma}\cdot\left(\overline{k1}-\overline{p2}\right)\right).\bar{\gamma}^{\$MU(\$27)}.\,(\varphi(-\overline{p2}))\,(\varphi(-\overline{p2}))\,.\bar{\gamma}^{\$MU(\$29)}.\left(\bar{\gamma}\cdot\left(\overline{k2}-\overline{p2}\right)\right).\bar{\gamma}^{\$MU(\$27)}.\,(\varphi(\overline{p1}))}{tu}$$
$$+\frac{e^4\,(\varphi(\overline{p1}))\,.\bar{\gamma}^{\$MU(\$29)}.\left(\bar{\gamma}\cdot\left(\overline{k1}-\overline{p2}\right)\right).\bar{\gamma}^{\$MU(\$27)}.\,(\varphi(-\overline{p2}))\,(\varphi(-\overline{p2}))\,.\bar{\gamma}^{\$MU(\$27)}.\left(\bar{\gamma}\cdot\left(\overline{k1}-\overline{p2}\right)\right).\bar{\gamma}^{\$MU(\$29)}.\,(\varphi(\overline{p1}))}{u^2}$$
$$+\frac{e^4\,(\varphi(\overline{p1}))\,.\bar{\gamma}^{\$MU(\$27)}.\left(\bar{\gamma}\cdot\left(\overline{k2}-\overline{p2}\right)\right).\bar{\gamma}^{\$MU(\$29)}.\,(\varphi(-\overline{p2}))\,(\varphi(-\overline{p2}))\,.\bar{\gamma}^{\$MU(\$29)}.\left(\bar{\gamma}\cdot\left(\overline{k2}-\overline{p2}\right)\right).\bar{\gamma}^{\$MU(\$27)}.\,(\varphi(\overline{p1}))}{t^2}$$

This is a small piece of the matrix element squared for $ggtoQ\bar{Q}$.

The proper summation over the polarizations of the gluons requires a choice of two auxiliary vectors (unless we subtract the unphysical contributions using ghosts).

It is customary to take the 4-momentum of another gluon as the auxiliary vector in the summation formula.

The option **`ExtraFactor`** is used to average over the polarizations of the initial gluons.

```
ClearAll[s, t, u];

FCClearScalarProducts[];

SP[p1] = 0;

SP[p2] = 0;
```

```
amp = 1/(s^2 SUNN (1 - SUNN^2) u^2) 2 SMP["g_s"]^4 SP[k1,
    Polarization[p2, -I, Transversality -> True]] SP[k1, Polarization[p2,
     I, Transversality -> True]] (2 s^2 SP[k1, Polarization[p1, I,
        Transversality -> True]] SP[k2, Polarization[p1, -I,
        Transversality -> True]] + 2 s SUNN^2 t SP[k1, Polarization[p1, I,
        Transversality -> True]] SP[k2, Polarization[p1, -I,
        Transversality -> True]] + s SUNN^2 u SP[k1, Polarization[p1, I,
        Transversality -> True]] SP[k2, Polarization[p1, -I,
        Transversality -> True]] + 2 s^2 SP[k1, Polarization[p1, -I,
        Transversality -> True]] SP[k2, Polarization[p1, I,
        Transversality -> True]] + 2 s SUNN^2 t SP[k1, Polarization[p1, -I,
        Transversality -> True]] SP[k2, Polarization[p1, I,
        Transversality -> True]] + s SUNN^2 u SP[k1, Polarization[p1, -I,
        Transversality -> True]] SP[k2, Polarization[p1, I,
        Transversality -> True]] + 2 SUNN^2 u^2 SP[k2,
       Polarization[p1, -I, Transversality -> True]] SP[k2,
       Polarization[p1, I, Transversality -> True]])
```

$$\frac{1}{N\left(1-N^2\right)s^2u^2}2g_s^4\left(\overline{\text{k1}}\cdot\bar{\varepsilon}^*(\text{p2})\right)\left(\overline{\text{k1}}\cdot\bar{\varepsilon}(\text{p2})\right)\left(2N^2st\left(\overline{\text{k1}}\cdot\bar{\varepsilon}(\text{p1})\right)\left(\overline{\text{k2}}\cdot\bar{\varepsilon}^*(\text{p1})\right)\right.$$
$$+2N^2st\left(\overline{\text{k1}}\cdot\bar{\varepsilon}^*(\text{p1})\right)\left(\overline{\text{k2}}\cdot\bar{\varepsilon}(\text{p1})\right)+N^2su\left(\overline{\text{k1}}\cdot\bar{\varepsilon}(\text{p1})\right)\left(\overline{\text{k2}}\cdot\bar{\varepsilon}^*(\text{p1})\right)+N^2su\left(\overline{\text{k1}}\cdot\bar{\varepsilon}^*(\text{p1})\right)\left(\overline{\text{k2}}\cdot\bar{\varepsilon}(\text{p1})\right)$$
$$\left.+2s^2\left(\overline{\text{k1}}\cdot\bar{\varepsilon}(\text{p1})\right)\left(\overline{\text{k2}}\cdot\bar{\varepsilon}^*(\text{p1})\right)+2s^2\left(\overline{\text{k1}}\cdot\bar{\varepsilon}^*(\text{p1})\right)\left(\overline{\text{k2}}\cdot\bar{\varepsilon}(\text{p1})\right)+2N^2u^2\left(\overline{\text{k2}}\cdot\bar{\varepsilon}^*(\text{p1})\right)\left(\overline{\text{k2}}\cdot\bar{\varepsilon}(\text{p1})\right)\right)$$

```
amp // DoPolarizationSums[#, p1, p2, ExtraFactor -> 1/2] & //
   DoPolarizationSums[#, p2, p1, ExtraFactor -> 1/2] & // Simplify
```

$$-\frac{1}{N\left(N^2-1\right)s^2u^2(\overline{\text{p1}}\cdot\overline{\text{p2}})^2}g_s^4\left(2\left(\overline{\text{k1}}\cdot\overline{\text{p1}}\right)\left(\overline{\text{k1}}\cdot\overline{\text{p2}}\right)\right.$$
$$-\overline{\text{k1}}^2\left(\overline{\text{p1}}\cdot\overline{\text{p2}}\right)\Big)\left(s\left(\overline{\text{k1}}\cdot\overline{\text{p2}}\right)\left(\overline{\text{k2}}\cdot\overline{\text{p1}}\right)\left(N^2(2t+u)+2s\right)\right.$$
$$+s\left(\overline{\text{k1}}\cdot\overline{\text{p1}}\right)\left(\overline{\text{k2}}\cdot\overline{\text{p2}}\right)\left(N^2(2t+u)+2s\right)-2N^2st\left(\overline{\text{k1}}\cdot\overline{\text{k2}}\right)\left(\overline{\text{p1}}\cdot\overline{\text{p2}}\right)-N^2su\left(\overline{\text{k1}}\cdot\overline{\text{k2}}\right)\left(\overline{\text{p1}}\cdot\overline{\text{p2}}\right)$$
$$\left.-2s^2\left(\overline{\text{k1}}\cdot\overline{\text{k2}}\right)\left(\overline{\text{p1}}\cdot\overline{\text{p2}}\right)+2N^2u^2\left(\overline{\text{k2}}\cdot\overline{\text{p1}}\right)\left(\overline{\text{k2}}\cdot\overline{\text{p2}}\right)-N^2u^2\overline{\text{k2}}^2\left(\overline{\text{p1}}\cdot\overline{\text{p2}}\right)\right)$$

We can also do the same calculation in D-dimensions

```
ClearAll[s, t, u];

FCClearScalarProducts[];

SPD[p1] = 0;

SPD[p2] = 0;
```

```
ChangeDimension[amp, D] // DoPolarizationSums[#, p1, p2, ExtraFactor ->
1/2] & //
   DoPolarizationSums[#, p2, p1, ExtraFactor -> 1/2] & // Simplify
```

$$-\frac{1}{N\left(N^2-1\right)s^2u^2(\mathrm{p1}\cdot\ \mathrm{p2})^2}g_s^4\left(2(\mathrm{k1}\cdot\ \mathrm{p1})(\mathrm{k1}\cdot\ \mathrm{p2})-\mathrm{k1}^2(\mathrm{p1}\cdot\ \mathrm{p2})\right)\left(s(\mathrm{k1}\cdot\ \mathrm{p2})(\mathrm{k2}\cdot\ \mathrm{p1})\left(N^2(2t+u)+2s\right)\right.$$
$$+s(\mathrm{k1}\cdot\ \mathrm{p1})(\mathrm{k2}\cdot\ \mathrm{p2})\left(N^2(2t+u)+2s\right)-2N^2st(\mathrm{k1}\cdot\ \mathrm{k2})(\mathrm{p1}\cdot\ \mathrm{p2})-N^2su(\mathrm{k1}\cdot\ \mathrm{k2})(\mathrm{p1}\cdot\ \mathrm{p2})$$
$$\left.-2s^2(\mathrm{k1}\cdot\ \mathrm{k2})(\mathrm{p1}\cdot\ \mathrm{p2})+2N^2u^2(\mathrm{k2}\cdot\ \mathrm{p1})(\mathrm{k2}\cdot\ \mathrm{p2})-\mathrm{k2}^2N^2u^2(\mathrm{p1}\cdot\ \mathrm{p2})\right)$$

DoPolarizationSums will complain if you try to sum over the polarizations of a massless vector boson that is not on-shell

```
PolarizationVector[p, mu] ComplexConjugate[PolarizationVector[p, mu]]

DoPolarizationSums[%, p, 0]
```

$$\bar{\varepsilon}^{*\mathrm{mu}}(p)\bar{\varepsilon}^{\mathrm{mu}}(p)$$

PolarizationSum: Warning! You are inserting a polarization sum for massless vector bosons, but the momentum of the external boson p is not on- shell. Please put it on- shell via ScalarProduct[p,p]=0

$$-4$$

The obvious solution to remove this warning is to put the boson on-shell

```
FCClearScalarProducts[]

ScalarProduct[p, p] = 0

PolarizationVector[p, mu] ComplexConjugate[PolarizationVector[p, mu]]

DoPolarizationSums[%, p, 0]
```

$$0$$

$$\bar{\varepsilon}^{*\mathrm{mu}}(p)\bar{\varepsilon}^{\mathrm{mu}}(p)$$

$$-4$$

However, if you have a massless virtual boson in the final state that by definition cannot be on-shell, (e.g. in the process $q\bar{q}\to g\gamma^*$), you can tell this to the function by setting the option **VirtualBoson** to **True**.

```
FCClearScalarProducts[]

PolarizationVector[p, mu] ComplexConjugate[PolarizationVector[p, mu]]

DoPolarizationSums[%, p, 0, VirtualBoson -> True]
```

$$\bar{\varepsilon}^{*\mathrm{mu}}(p)\bar{\varepsilon}^{\mathrm{mu}}(p)$$

$$-4$$

It may happen that your expression is not directly proportional to a pair of polarization vectors. In this case terms that are free of polarization vectors will be multiplied by the suitable number of polarizations. This behavior is controlled by the option **NumberOfPolarizations**. The default value **Automatic** means that the function will automatically figure out the correct number of polarizations.

Here we have 2 physical polarizations (massless vector boson)

```
FCClearScalarProducts[];

ScalarProduct[p, p] = 0;

PolarizationVector[p, mu] ComplexConjugate[PolarizationVector[p, mu]] + xyz


DoPolarizationSums[%, p, n]
```

$$\bar{\varepsilon}^{*\mathrm{mu}}(p)\bar{\varepsilon}^{\mathrm{mu}}(p) + \mathrm{xyz}$$

DoPolarizationSums: The input expression contains terms free of polarization vectors. Those will be multiplied wit

$$2\ \mathrm{xyz} - 2$$

In D dimensions the number of polarizations becomes $D - 2$

```
FCClearScalarProducts[];

ScalarProduct[p, p] = 0;

ChangeDimension[PolarizationVector[p, mu]*
     ComplexConjugate[PolarizationVector[p, mu]] + xyz, D]

DoPolarizationSums[%, p, n]
```

$$\varepsilon^{*\mathrm{mu}}(p)\varepsilon^{\mathrm{mu}}(p) + \mathrm{xyz}$$

DoPolarizationSums: The input expression contains terms free of polarization vectors. Those will be multiplied wit
-2.

$$(D-2)\ \mathrm{xyz} - D + 2$$

A massive vector boson has 3 physical polarizations in 4 dimensions

```
FCClearScalarProducts[];

ScalarProduct[p, p] = M^2;

PolarizationVector[p, mu] ComplexConjugate[PolarizationVector[p, mu]] + xyz


DoPolarizationSums[%, p]
```

$$\bar{\varepsilon}^{*\mathrm{mu}}(p)\bar{\varepsilon}^{\mathrm{mu}}(p) + \mathrm{xyz}$$

DoPolarizationSums: The input expression contains terms free of polarization vectors. Those will be multiplied wit

$$3\ \mathrm{xyz} - 3$$

or $D-1$ physical polarizations in D dimensions

```
FCClearScalarProducts[];

ScalarProduct[p, p] = M^2;

ChangeDimension[PolarizationVector[p, mu]*
     ComplexConjugate[PolarizationVector[p, mu]] + xyz, D]

DoPolarizationSums[%, p]
```

$$\varepsilon^{*\mathrm{mu}}(p)\varepsilon^{\mathrm{mu}}(p) + \mathrm{xyz}$$

DoPolarizationSums: The input expression contains terms free of polarization vectors. Those will be multiplied wit
-1.

$$(D-1)\ \text{xyz} - D + 1$$

In the case of a standalone expression that contains no polarization vectors whatsoever, the function has no way to determine the correct number of polarizations.

```
DoPolarizationSums[xyz, p, n]
```

DoPolarizationSums: The option NumberOfPolarizations is set to Automatic but since the input expression contains no polarization vectors, the function cannot determine the number of polarizations to sum over in the expression automatically. Please set NumberOfPolarizations to the appropriate value e.g. 2 or D-2 etc. by hand.

$Aborted

Here additional user input is needed

```
DoPolarizationSums[xyz, p, NumberOfPolarizations -> 2]
```

DoPolarizationSums: The input expression contains terms free of polarization vectors. Those will be multiplied wit

$$2\ \text{xyz}$$

11.6 PolarizationSum

`PolarizationSum[mu, nu, ... ]` represents the sum over a polarization vector and its complex conjugate with two free indices. Depending on its arguments the function returns different polarization sums for massive or massless vector bosons.

- **`PolarizationSum[nu, nu, k]`** returns $-g^{\mu\nu} + \frac{k^\mu k^\nu}{k^2}$, i.e. the sum over the 3 physical polarizations of a massive on-shell vector boson with $m = k^2$.
- **`PolarizationSum[mu, nu]`** or **`PolarizationSum[mu, nu, k, 0]`** gives $-g^{\mu\nu}$. This corresponds to the summation over all 4 polarizations of a massless vector boson, 2 of which are unphysical if the particle is on-shell.
- **`PolarizationSum[mu, nu, k, n]`** yields $-g^{\mu\nu} + \frac{k^\mu n^\nu + k^\nu n^\mu}{k\cdot n} - \frac{n^2 k^\mu k^\nu}{(k\cdot n)^2}$ which is the so-called axial-gauge polarization sum that picks up only the two physical polarizations of a massless vector boson. Here n is an auxiliary vector that must satisfy $n \cdot k \neq 0$. The physical results will not depend on n, yet in practice it is often convenient to identify n with one of the 4-vectors already present in the calculation. For example, in a final state with multiple gluons denoted by their momenta k_i, the vector n for the i-th gluon could be a k_j with $j \neq i$. Notice that when using this polarization sum in a QCD calculation, one doesn't have to consider diagrams with ghosts in the final states.

To obtain a D-dimensional polarization sum use the option **`Dimension`**.

If you need to calculate a polarization sum depending on a 4-momentum that is not on-shell, use the option **`VirtualBoson`**.

11.6.1 See also

Overview, Polarization, DoPolarizationSums, Uncontract.

11.6.2 Examples

```
PolarizationSum[\[Mu], \[Nu]]
```

$$-\bar{g}^{\mu\nu}$$

```
PolarizationSum[\[Mu], \[Nu], k]
```

$$\frac{\overline{k}^{\mu}\overline{k}^{\nu}}{\overline{k}^2} - \bar{g}^{\mu\nu}$$

```
PolarizationSum[\[Mu], \[Nu], k, Dimension -> D]
```

$$\frac{k^{\mu}k^{\nu}}{k^2} - g^{\mu\nu}$$

```
FCClearScalarProducts[]; SP[k] = 0;

PolarizationSum[\[Mu], \[Nu], k, n]
```

$$-\frac{\overline{n}^2\overline{k}^{\mu}\overline{k}^{\nu}}{(\overline{k}\cdot\overline{n})^2} - \bar{g}^{\mu\nu} + \frac{\overline{k}^{\nu}\overline{n}^{\mu}}{\overline{k}\cdot\overline{n}} + \frac{\overline{k}^{\mu}\overline{n}^{\nu}}{\overline{k}\cdot\overline{n}}$$

```
FCClearScalarProducts[]

PolarizationSum[\[Mu], \[Nu], k, 0, Dimension -> D]
```

```
PolarizationSum: Warning! You are inserting a polarization sum for massless vector bosons, but the momentum of
the external boson k is not on-shell. Please put it on-shell via ScalarProduct[k,k]=0
```

$$-g^{\mu\nu}$$

```
FCClearScalarProducts[]

PolarizationSum[\[Mu], \[Nu], k, 0, Dimension -> D, VirtualBoson -> True]
```

$$-g^{\mu\nu}$$

11.7 ExpandPartialD

ExpandPartialD[exp] expands noncommutative products of **QuantumField**s and partial differentiation operators in **exp** and applies the Leibniz rule.

By default the function assumes that there are no expressions outside of **exp** on which the derivatives inside **exp** could act. If this is not the case, please set the options **LeftPartialD** or **RIghtPartialD** to **True**.

11.7.1 See also

Overview, ExplicitPartialD, LeftPartialD, LeftRightPartialD, PartialDRelations, RightPartialD, LeftRightNablaID, LeftRightNablaID2, LeftNablaID, RightNablaID.

11.7.2 Examples

```
RightPartialD[\[Mu]] . QuantumField[A, LorentzIndex[\[Mu]]] .
QuantumField[A, LorentzIndex[\[Nu]]]

ExpandPartialD[%]
```

$$\vec{\partial}_\mu.A_\mu.A_\nu$$

$$A_\mu.\left(\partial_\mu A_\nu\right)+\left(\partial_\mu A_\mu\right).A_\nu$$

```
RightNablaD[i] . QuantumField[A, LorentzIndex[\[Mu]]] . QuantumField[A,
LorentzIndex[\[Nu]]]

ExpandPartialD[%]
```

$$\vec{\nabla}^i.A_\mu.A_\nu$$

$$-A_\mu.\left(\partial_i A_\nu\right)-\left(\partial_i A_\mu\right).A_\nu$$

```
LeftRightPartialD[\[Mu]] . QuantumField[A, LorentzIndex[\[Nu]]]

ExpandPartialD[%]
```

$$\overleftrightarrow{\partial}_\mu.A_\nu$$

$$\frac{1}{2}\left(\left(\partial_\mu A_\nu\right)-\overleftarrow{\partial}_\mu.A_\nu\right)$$

```
LeftRightNablaD[i] . QuantumField[A, LorentzIndex[\[Nu]]]

ExpandPartialD[%]
```

$$\overleftrightarrow{\nabla}_i.A_\nu$$

$$\frac{1}{2}\left(\overleftarrow{\partial}_i.A_\nu - (\partial_i A_\nu)\right)$$

```
QuantumField[A, LorentzIndex[\[Mu]]] . (LeftRightPartialD[OPEDelta]^2) .
QuantumField[A,
    LorentzIndex[\[Rho]]]

ExpandPartialD[%]
```

$$A_\mu.\overleftrightarrow{\partial}^2_\Delta.A_\rho$$

$$\frac{1}{4}\left(A_\mu.\left(\partial_\Delta\partial_\Delta A_\rho\right) - 2\left(\partial_\Delta A_\mu\right).\left(\partial_\Delta A_\rho\right) + \left(\partial_\Delta\partial_\Delta A_\mu\right).A_\rho\right)$$

```
8 LeftRightPartialD[OPEDelta]^3
```

$$8\overleftrightarrow{\partial}^3_\Delta$$

```
ExplicitPartialD[%]
```

$$\left(\vec{\partial}_\Delta - \overleftarrow{\partial}_\Delta\right)^3$$

```
ExpandPartialD[%]
```

$$-\overleftarrow{\partial}_\Delta.\overleftarrow{\partial}_\Delta.\overleftarrow{\partial}_\Delta + 3\overleftarrow{\partial}_\Delta.\overleftarrow{\partial}_\Delta.\vec{\partial}_\Delta - 3\overleftarrow{\partial}_\Delta.\vec{\partial}_\Delta.\vec{\partial}_\Delta + \vec{\partial}_\Delta.\vec{\partial}_\Delta.\vec{\partial}_\Delta$$

```
LC[\[Mu], \[Nu], \[Rho], \[Tau]] RightPartialD[\[Alpha], \[Mu], \[Beta],
\[Nu]]

ExpandPartialD[%]
```

$$\bar{\epsilon}^{\mu\nu\rho\tau}\vec{\partial}_{\alpha}.\vec{\partial}_{\mu}.\vec{\partial}_{\beta}.\vec{\partial}_{\nu}$$

$$0$$

```
CLC[i, j, k] RightNablaD[i, j, k]

ExpandPartialD[%]
```

$$\bar{\epsilon}^{ijk}\vec{\nabla}^{i}.\vec{\nabla}^{j}.\vec{\nabla}^{k}$$

$$0$$

```
RightPartialD[CartesianIndex[i]] . QuantumField[S, x]

% // ExpandPartialD
```

$$\vec{\partial}_{i}.S^{x}$$

$$(\partial_{i}S^{x})$$

```
RightPartialD[{CartesianIndex[i], x}] . QuantumField[S, x]

% // ExpandPartialD
```

$$\vec{\partial}_{\{i,x\}}.S^{x}$$

$$\left(\partial_{\{i,x\}}S^{x}\right)$$

By default the derivative won't act on anything outside of the input expression. But it can be made to by setting the option **`RightPartialD`** to **`True`**

```
ExpandPartialD[RightPartialD[\[Mu]] . QuantumField[A, LorentzIndex[\[Mu]]]
. QuantumField[A, LorentzIndex[\[Nu]]]]
```

$$A_{\mu}.\left(\partial_{\mu}A_{\nu}\right)+\left(\partial_{\mu}A_{\mu}\right).A_{\nu}$$

```
ExpandPartialD[RightPartialD[\[Mu]] . QuantumField[A, LorentzIndex[\[Mu]]]
. QuantumField[A, LorentzIndex[\[Nu]]], RightPartialD -> True]
```

$$A_\mu.\left(\partial_\mu A_\nu\right)+\left(\partial_\mu A_\mu\right).A_\nu+A_\mu.A_\nu.\vec{\partial}_\mu$$

The same applies also to **LeftPartialD**

```
ExpandPartialD[QuantumField[A, LorentzIndex[\[Nu]]] . LeftNablaD[i]]
```

$$-\left(\partial_i A_\nu\right)$$

```
ExpandPartialD[QuantumField[A, LorentzIndex[\[Nu]]] . LeftNablaD[i],
LeftPartialD -> True]
```

$$-\left(\partial_i A_\nu\right)-\overleftarrow{\partial}_i.A_\nu$$

11.8 ExplicitPartialD

ExplicitPartialD[exp] inserts the definitions for **LeftRightPartialD**, **LeftRightPartialD2**, **LeftRightNablaD**, **LeftRightNablaD2**, **LeftNablaD** and **RightNablaD**

11.8.1 See also

Overview, ExpandPartialD, LeftRightPartialD, LeftRightPartialD2, LeftRightNablalD, LeftRightNablalD2, LeftNablalD, RightNablalD.

11.8.2 Examples

```
LeftRightPartialD[\[Mu]]

ExplicitPartialD[%]
```

$$\overleftrightarrow{\partial}_\mu$$

$$\frac{1}{2}\left(\vec{\partial}_\mu-\overleftarrow{\partial}_\mu\right)$$

```
LeftRightPartialD2[\[Mu]]

ExplicitPartialD[%]
```

$$\overleftrightarrow{\partial}_{\mu}$$

$$\overleftarrow{\partial}_{\mu} + \vec{\partial}_{\mu}$$

```
LeftRightPartialD[OPEDelta]

ExplicitPartialD[%]
```

$$\overleftrightarrow{\partial}_{\Delta}$$

$$\frac{1}{2}\left(\vec{\partial}_{\Delta} - \overleftarrow{\partial}_{\Delta}\right)$$

```
16 LeftRightPartialD[OPEDelta]^4

ExplicitPartialD[%]
```

$$16\,\overleftrightarrow{\partial}_{\Delta}^{4}$$

$$\left(\vec{\partial}_{\Delta} - \overleftarrow{\partial}_{\Delta}\right)^{4}$$

Notice that by definition $\nabla^i = \partial_i = -\partial^i$, where the last equality depends on the metric signature.

```
LeftNablaD[i]

ExplicitPartialD[%]
```

$$\overleftarrow{\nabla}^{i}$$

$$-\overleftarrow{\partial}_{i}$$

```
RightNablaD[i]

ExplicitPartialD[%]
```

$$\vec{\nabla}^i$$

$$-\vec{\partial}_i$$

```
LeftRightNablaD[i]

ExplicitPartialD[%]
```

$$\overleftrightarrow{\nabla}_i$$

$$\frac{1}{2}\overleftarrow{\partial}_i - \vec{\partial}_i$$

```
LeftRightNablaD2[\[Mu]]

ExplicitPartialD[%]
```

$$\overleftrightarrow{\nabla}_\mu$$

$$-\overleftarrow{\partial}_\mu - \vec{\partial}_\mu$$

11.9 FAPatch

FAPatch[] is an auxiliary function that patches FeynArts to be compatible with FeynCalc. If an unpatched copy of FeynArts is present in $FeynArtsDirectory, evaluating FAPatch[] will start the patching process.

11.9.1 See also

Overview, PatchModelsOnly, FAModelsDirectory.

11.9.2 Examples

Setting the option **Quiet** to **True** will suppress the **ChoiceDialog** asking whether you really want to patch FeynArts.

```
(*FAPatch[Quiet->True]*)
```

If you just want to patch some new models (e.g. generated with FeynRules), while your FeynArts version is already patched, use the option **PatchModelsOnly**.

```
(*FAPatch[PatchModelsOnly->True]*)
```

The model files do not necessarily have to be located inside **FileNameJoin[{$FeynArtsDirectory, "Models"}]**. A custom location can be specified via the option **FAModelsDirectory** as in

```
(*FAPatch[PatchModelsOnly->True,FAModelsDirectory->
FileNameJoin[{ParentDirectory@NotebookDirectory[],"FeynArts","MyModel"}]]*)
```

11.10 FCAttachTypesettingRule

FCAttachTypesettingRule[expr, ...] attaches a specific **TraditionalForm** typesetting rule to **expr**. It doesn't change any properties of expr apart from adding a **FormatValue** with a **MakeBoxes** rule.

Following choices are possible:

- **FCAttachTypesettingRule[expr_, str]**
- **FCAttachTypesettingRules[expr, {SubscriptBox, var, sub}]**
- **FCAttachTypesettingRules[expr, {SuperscriptBox, var, sup}]**
- **FCAttachTypesettingRules[expr, {SubsuperscriptBox, var, sub, sup}]**

Use **FCRemoveTypesettingRules** to remove all typesetting rules attached to expr.

11.10.1 See also

Overview, FCRemoveTypesettingRules.

11.10.2 Examples

```
FCRemoveTypesettingRules[mu]
```

```
FCAttachTypesettingRule[mu, "\[Mu]"]

mu
```

$$\mu$$

```
mc["d_ss"]
```

$$\text{mc}(\text{d\_ss})$$

```
FCAttachTypesettingRule[mc["d_ss"], {SubscriptBox, "d", "ss"}]

mc["d_ss"]
```

$$d_{\text{ss}}$$

```
m12
```

$$\text{m12}$$

```
FCAttachTypesettingRule[m12, {SubsuperscriptBox, m, 1, 2}]

m12
```

$$m_1^2$$

```
{p1, p2, p3, p4}

MapThread[FCAttachTypesettingRule[#1, {SubscriptBox, "p", #2}] &, {{p1, p2,
p3, p4}, Range[4]}];

{p1, p2, p3, p4}
```

$$\{\text{p1}, \text{p2}, \text{p3}, \text{p4}\}$$

$$\{p_1, p_2, p_3, p_4\}$$

```
FCRemoveTypesettingRules[mu]

FCRemoveTypesettingRules[mc["d_ss"]]

FCRemoveTypesettingRules[m12]

FCRemoveTypesettingRules /@ {p1, p2, p3, p4};
```

11.11 FCRemoveTypesettingRules

`FCRemoveTypesettingRules[expr]` removes all typesetting rules attached to **`expr`**. Effectively it sets the **`FormatValues`** of **`expr`** to an empty list.

11.11.1 See also

Overview, FCAttachTypesettingRule.

11.11.2 Examples

```
ST1
```

ST1

```
FCAttachTypesettingRule[ST1, {SubscriptBox, "S", "T,1"}]
```

```
ST1
```

$S_{T,1}$

```
FCRemoveTypesettingRules[ST1]

ST1
```

ST1

11.12 FCFAConvert

FCFAConvert[exp] converts a FeynArts amplitude to FeynCalc.

For examples on using **FCFAConvert** please examine the example calculations shipped with FeynCalc.

11.12.1 See also

Overview.

11.12.2 Examples

11.13 FCPrepareFAAmp

FCPrepareFAAmp[exp] is an auxiliary function for a partial conversion of a FeynArts amplitude to FeynCalc.

11.13.1 See also

Overview

11.13.2 Examples

```
FCClearScalarProducts[]

FeynArts`FAFeynAmpDenominator[FeynArts`FAPropagatorDenominator[Momentum[P,
D], MW Sqrt[FeynArts`FAGaugeXi[W]]],
FeynArts`FAPropagatorDenominator[Momentum[k, D], m]]

FCPrepareFAAmp[%]
```

$$\text{FeynArts`FAFeynAmpDenominator}\left(\text{FeynArts`FAPropagatorDenominator}\left(P,\ \text{MW}\sqrt{\text{FeynArts`FAGaugeXi}(W)}\right),\ \text{FeynArts`FAPropagatorDenominator}(k,m)\right)$$

$$\frac{1}{\left(\overline{P}^2-\text{MW}^2\xi_W\right).\left(\overline{k}^2-m^2\right)}$$

```
FeynArts`IndexDelta[FeynArts`Index[Global`Gluon, 1],
FeynArts`Index[Global`Gluon, 2]]

FCPrepareFAAmp[%]
```

$$\text{FeynArts`IndexDelta}(\text{FeynArts`Index}(\text{Gluon}, 1), \text{FeynArts`Index}(\text{Gluon}, 2))$$

$$\delta^{\text{Glu1 Glu2}}$$

11.14 FCTP

FCTP is an alias for **FCTripleProduct**.

11.14.1 See also

Overview, FCTripleProduct.

11.14.2 Examples

```
FCTP[a, b, c]
```

$$\overline{a} \cdot \left(\overline{b} \times \overline{c}\right)$$

11.15 FCTripleProduct

FCTripleProduct[a, b, c] returns the triple product $a \cdot (b \times c)$. By default **a**,**b** and **c** are assumed to be Cartesian vectors. Wrapping the arguments with **CartesianIndex** will create an expression with open indices.

If any of the arguments is noncommutative, **DOT** will be used instead of **Times** and the function will introduce dummy indices. To give those indices some specific names, use the option **CartesianIndexNames**.

If the arguments already contain free CartesianIndices, the first such index will be used for the contraction.

To obtain an explicit expression you need to set the option **Explicit** to **True** or apply the function **Explicit**

11.15.1 See also

Overview, Eps.

11.15.2 Examples

```
FCTP[a, b, c]

% // StandardForm
```

$$\overline{a} \cdot \left(\overline{b} \times \overline{c}\right)$$

```
(*FCTripleProduct[a, b, c]*)
```

```
FCTP[a, b, c, Explicit -> True]

% // StandardForm
```

$$\overline{\epsilon}^{\overline{a}\overline{b}\overline{c}}$$

```
(*Eps[CartesianMomentum[a], CartesianMomentum[b], CartesianMomentum[c]]*)
```

```
FCTP[QuantumField[A, CartesianIndex[i]], QuantumField[B,
CartesianIndex[j]],
  QuantumField[C, CartesianIndex[k]], Explicit -> True]
```

$$\overline{\epsilon}^{ijk} A^i.B^j.C^k$$

```
FCTP[a, b, c, Explicit -> True, NonCommutative -> True, CartesianIndexNames
-> {i, j, k}]
```

$$\overline{\epsilon}^{ijk} \overline{a}^i.\overline{b}^j.\overline{c}^k$$

11.16 FermionSpinSum

FermionSpinSum[exp] converts products of closed spinor chains in **exp** into Dirac traces. Both Dirac and Majorana particles are supported. It is understood, that **exp** represents a squared amplitude.

11.16.1 See also

Overview, Spinor, ComplexConjugate, DiracTrace.

11.16.2 Examples

FeynCalc uses the customary relativistic normalization of the spinors.

```
SpinorUBar[Momentum[p], m] . SpinorU[Momentum[p], m]

FermionSpinSum[%]

DiracSimplify[%]
```

$$\bar{u}\left(\overline{p}, m\right).u\left(\overline{p}, m\right)$$

$$\text{tr}\left(\bar{\gamma}\cdot\overline{p}+m\right)$$

$$4m$$

```
SpinorVBar[Momentum[p], m] . SpinorV[Momentum[p], m]

FermionSpinSum[%]

DiracSimplify[%]
```

$$\bar{v}\left(\overline{p}, m\right).v\left(\overline{p}, m\right)$$

$$\text{tr}\left(\bar{\gamma}\cdot\overline{p}-m\right)$$

$$-4m$$

```
amp = SpinorUBar[k1, m] . GS[p] . GA[5] . SpinorU[p1, m]

ampSq = amp ComplexConjugate[amp]
```

$$\bar{u}(\text{k1}, m).\left(\bar{\gamma}\cdot\overline{p}\right).\bar{\gamma}^5.u(\text{p1}, m)$$

$$\bar{u}(\text{k1}, m).\left(\bar{\gamma}\cdot\overline{p}\right).\bar{\gamma}^5.u(\text{p1}, m)\left(-\left(\varphi(\overline{\text{p1}}, m)\right).\bar{\gamma}^5.\left(\bar{\gamma}\cdot\overline{p}\right).\left(\varphi(\overline{\text{k1}}, m)\right)\right)$$

```
FermionSpinSum[ampSq]

DiracSimplify[%]
```

$$-\text{tr}\left(\left(\bar{\gamma}\cdot\overline{\text{k1}}+m\right).\left(\bar{\gamma}\cdot\overline{p}\right).\bar{\gamma}^5.\left(\bar{\gamma}\cdot\overline{\text{p1}}+m\right).\bar{\gamma}^5.\left(\bar{\gamma}\cdot\overline{p}\right)\right)$$

$$-4\overline{p}^2\left(\overline{\text{k1}}\cdot\overline{\text{p1}}\right)+8\left(\overline{\text{k1}}\cdot\overline{p}\right)\left(\overline{p}\cdot\overline{\text{p1}}\right)-4m^2\overline{p}^2$$

11.17 FeynRule

FeynRule[lag, {fields}] derives the Feynman rule corresponding to the field configuration **fields** of the Lagrangian **lag**.

FeynRule does not calculate propagator Feynman rules.

The option **ZeroMomentumInsertion** can be used for twist-2 and higher twist operators.

FeynRule is not very versatile and was primarily developed for QCD calculations. It is often more useful when dealing with bosonic fields than with fermions. If you need a more powerful and universal solution for deriving Feynman rules, have a look at the standalone Mathematica Package FeynRules (not related to FeynCalc).

11.17.1 See also

Overview

11.17.2 Examples

```
?Lagrangian
```

Symbol

Lagrangian["oqu"] gives the unpolarized OPE quark operator.

Lagrangian["oqp"] gives the polarized quark OPE operator.

Lagrangian["ogu"] gives the unpolarized gluon OPE operator.

Lagrangian["ogp"] gives the polarized gluon OPE operator.

Lagrangian["ogd"] gives the sigma- term part of the QCD Lagrangian.

Lagrangian["QCD"] gives the gluon self interaction part of the QCD Lagrangian.

ϕ^4 Feynman rule

```
- \[Lambda]/4! QuantumField[\[Phi]]^4

FeynRule[%, {QuantumField[\[Phi]][p1], QuantumField[\[Phi]][p2],
   QuantumField[\[Phi]][p3], QuantumField[\[Phi]][p4]}]
```

$$-\frac{\lambda\phi^4}{24}$$

$$-i\lambda$$

Quark-gluon vertex Feynman rule

```
I QuantumField[AntiQuarkField] . GA[\[Mu]] . CovariantD[\[Mu]] .
QuantumField[QuarkField]

FeynRule[%, {QuantumField[GaugeField, {\[Mu]}, {a}][p1],
   QuantumField[QuarkField][p2], QuantumField[AntiQuarkField][p3]}]
```

$$i\bar{\psi}.\bar{\gamma}^{\mu}.D_{\mu}.\psi$$

$$iT^a g_s \bar{\gamma}^{\mu}$$

4-gluon vertex Feynman rule

```
-(1/4) FieldStrength[\[Alpha], \[Beta], i] . FieldStrength[\[Alpha],
\[Beta], i]

FeynRule[%, {QuantumField[GaugeField, {\[Mu]}, {a}][p1],
QuantumField[GaugeField, {\[Nu]}, {b}][p2],
    QuantumField[GaugeField, {\[Rho]}, {c}][p3], QuantumField[GaugeField,
{\[Sigma]}, {d}][p4]}]

GluonVertex[{p, \[Mu], a}, {q, \[Nu], b}, {r, \[Rho], c}, {s, \[Sigma], d},
Dimension -> 4, Explicit -> True]

FCCanonicalizeDummyIndices[% - %%] // Factor
```

$$-\frac{1}{4}F^i_{\alpha\beta}.F^i_{\alpha\beta}$$

$$ig_s^2 f^{adFCGV(si49)} f^{bcFCGV(si49)} \left(\bar{g}^{\mu\rho}\bar{g}^{\nu\sigma} - \bar{g}^{\mu\nu}\bar{g}^{\rho\sigma}\right) + ig_s^2 f^{acFCGV(si49)} f^{bdFCGV(si49)} \left(\bar{g}^{\mu\sigma}\bar{g}^{\nu\rho} - \bar{g}^{\mu\nu}\bar{g}^{\rho\sigma}\right) + ig_s^2 f^{abFCGV(si49)} f^{cdFCGV(si49)} \left(\bar{g}^{\mu\sigma}\bar{g}^{\nu\rho} - \bar{g}^{\mu\rho}\bar{g}^{\nu\sigma}\right)$$

$$-ig_s^2 \left(f^{adFCGV(u56)} f^{bcFCGV(u56)} \left(\bar{g}^{\mu\nu}\bar{g}^{\rho\sigma} - \bar{g}^{\mu\rho}\bar{g}^{\nu\sigma}\right) + f^{acFCGV(u56)} f^{bdFCGV(u56)} \left(\bar{g}^{\mu\nu}\bar{g}^{\rho\sigma} - \bar{g}^{\mu\sigma}\bar{g}^{\nu\rho}\right) + f^{abFCGV(u56)} f^{cdFCGV(u56)} \left(\bar{g}^{\mu\rho}\bar{g}^{\nu\sigma} - \bar{g}^{\mu\sigma}\bar{g}^{\nu\rho}\right)\right)$$

$$0$$

3-gluon vertex Feynman rule

```
-(1/4) FieldStrength[\[Alpha], \[Beta], i] . FieldStrength[\[Alpha],
\[Beta], i]

FeynRule[%, {QuantumField[GaugeField, {\[Mu]}, {a}][p],
QuantumField[GaugeField, {\[Nu]}, {b}][q],
    QuantumField[GaugeField, {\[Rho]}, {c}][r]}]

GluonVertex[{p, \[Mu], a}, {q, \[Nu], b}, {r, \[Rho], c}, Dimension -> 4,
Explicit -> True]

ExpandScalarProduct[% - %%] // Factor
```

$$-\frac{1}{4}F^i_{\alpha\beta}.F^i_{\alpha\beta}$$

$$g_s f^{abc}\left(\bar{g}^{\mu\nu}\left(\overline{p}^{\rho}-\overline{q}^{\rho}\right)-\bar{g}^{\mu\rho}\left(\overline{p}^{\nu}-\overline{r}^{\nu}\right)+\bar{g}^{\nu\rho}\left(\overline{q}^{\mu}-\overline{r}^{\mu}\right)\right)$$

$$g_s f^{abc}\left(\bar{g}^{\mu\nu}\left(\overline{p}-\overline{q}\right)^{\rho}+\bar{g}^{\mu\rho}\left(\overline{r}-\overline{p}\right)^{\nu}+\bar{g}^{\nu\rho}\left(\overline{q}-\overline{r}\right)^{\mu}\right)$$

$$0$$

Higgs EFT interaction vertex

```
heftInt = -(1/4) CH FieldStrength[mu, nu, a] . FieldStrength[mu, nu, a] .
QuantumField[H]
```

$$-\frac{1}{4}\,\mathrm{CH}F^a_{\mathrm{mu\ nu}}.F^a_{\mathrm{mu\ nu}}.H$$

Hgg vertex Feynman rules

```
FeynRule[heftInt, {QuantumField[GaugeField, {i}, {a}][p1],
QuantumField[GaugeField,
     {j}, {b}][p2], QuantumField[H][p3]}]
```

$$-i\,\mathrm{CH}\delta^{ab}\left(\overline{\mathrm{p2}}^i\overline{\mathrm{p1}}^j-\bar{g}^{ij}\left(\overline{\mathrm{p1}}\cdot\overline{\mathrm{p2}}\right)\right)$$

Hggg vertex Feynman rules

```
FeynRule[heftInt, {QuantumField[GaugeField, {i}, {a}][p1],
QuantumField[GaugeField,
      {j}, {b}][p2], QuantumField[GaugeField, {k}, {c}][p3],
      QuantumField[H][p4]}] // Simplify
```

$$\mathrm{CH}g_s f^{abc}\left(\bar{g}^{ij}\left(\overline{\mathrm{p1}}^k-\overline{\mathrm{p2}}^k\right)-\bar{g}^{ik}\left(\overline{\mathrm{p1}}^j-\overline{\mathrm{p3}}^j\right)+\bar{g}^{jk}\left(\overline{\mathrm{p2}}^i-\overline{\mathrm{p3}}^i\right)\right)$$

Hgggg vertex Feynman rules

```
FeynRule[heftInt, {QuantumField[GaugeField, {i}, {a}][p1],
QuantumField[GaugeField, {j},
       {b}][p2], QuantumField[GaugeField, {k}, {c}][p3],
     QuantumField[GaugeField, {l}, {d}][p4], QuantumField[H][p5]}] //
   FCCanonicalizeDummyIndices[#, SUNIndexNames -> {e}] & // Collect2[#,
SUNF,
    FCFactorOut -> I CH SMP["g_s"]^2] &
```

$$i\,\mathrm{CH} g_s^2 \left(f^{ade} f^{bce} \left(\bar{g}^{ik} \bar{g}^{jl} - \bar{g}^{ij} \bar{g}^{kl}\right) + f^{ace} f^{bde} \left(\bar{g}^{il} \bar{g}^{jk} - \bar{g}^{ij} \bar{g}^{kl}\right) + f^{abe} f^{cde} \left(\bar{g}^{il} \bar{g}^{jk} - \bar{g}^{ik} \bar{g}^{jl}\right)\right)$$

Some OPE-related examples:

2-quark Feynman rules (unpolarized).

```
Lagrangian["oqu"]
FeynRule[%, {QuantumField[QuarkField][p], QuantumField[AntiQuarkField][q]},
   ZeroMomentumInsertion -> False] // Factor
```

$$i^m \bar{\psi}.\left(\bar{\gamma}\cdot\Delta\right).{D_\Delta}^{m-1}.\psi$$

$$-i i^m \left(\bar{\gamma}\cdot\Delta\right).\left(\vec{\partial}_\Delta\right)^{m-1}$$

```
?ZeroMomentumInsertion
```

Symbol

ZeroMomentumInsertion is an option of FeynRule, Twist2GluonOperator and Twist2QuarkOperator.

```
ExpandPartialD[Lagrangian["oqu"] /. OPEm -> 3]
```

$$\begin{aligned} & i g_s^2 T^{c152}.T^{c153}.\left(\bar{\gamma}\cdot\Delta\right).\bar{\psi}.A_\Delta^{c152}.A_\Delta^{c153}.\psi - g_s T^{c152}.\left(\bar{\gamma}\cdot\Delta\right).\bar{\psi}.A_\Delta^{c152}.\left(\partial_\Delta\psi\right) \\ & - g_s T^{c153}.\left(\bar{\gamma}\cdot\Delta\right).\bar{\psi}.A_\Delta^{c153}.\left(\partial_\Delta\psi\right) - g_s T^{c153}.\left(\bar{\gamma}\cdot\Delta\right).\bar{\psi}.\left(\partial_\Delta A_\Delta^{c153}\right).\psi - i\left(\bar{\gamma}\cdot\Delta\right).\bar{\psi}.\left(\partial_\Delta\partial_\Delta\psi\right) \end{aligned}$$

```
Lagrangian["oqu"]
FeynRule[% /. OPEm -> 3, {QuantumField[QuarkField][p],
QuantumField[AntiQuarkField][q],
   QuantumField[GaugeField, {\[Mu]}, {a}][r]}, ZeroMomentumInsertion ->
True]
FCReplaceMomenta[%, {r -> -p - q}] // ExpandScalarProduct // Expand
```

$$i^m \bar{\psi}.(\bar{\gamma}\cdot\Delta).{D_\Delta}^{m-1}.\psi$$

$$T^a \Delta^\mu g_s \left(-(\bar{\gamma}\cdot\Delta)\right)(2(\Delta\cdot\overline{q})+\Delta\cdot\bar{r})$$

$$T^a \Delta^\mu g_s \bar{\gamma}\cdot\Delta(\Delta\cdot\overline{p}) - T^a \Delta^\mu g_s \bar{\gamma}\cdot\Delta(\Delta\cdot\overline{q})$$

```
Lagrangian["oqu"]
FeynRule[% /. OPEm -> 4, {QuantumField[QuarkField][p],
QuantumField[AntiQuarkField][q],
   QuantumField[GaugeField, {\[Mu]}, {a}][r]}, ZeroMomentumInsertion ->
True]
FCReplaceMomenta[%, {r -> -p - q}] // ExpandScalarProduct // Expand
```

$$i^m \bar{\psi}.(\bar{\gamma}\cdot\Delta).{D_\Delta}^{m-1}.\psi$$

$$T^a \Delta^\mu g_s \bar{\gamma}\cdot\Delta\left(3(\Delta\cdot\overline{q})^2+(\Delta\cdot\bar{r})^2+3(\Delta\cdot\overline{q})(\Delta\cdot\bar{r})\right)$$

$$T^a \Delta^\mu g_s \bar{\gamma}\cdot\Delta(\Delta\cdot\overline{p})^2 + T^a \Delta^\mu g_s \bar{\gamma}\cdot\Delta(\Delta\cdot\overline{q})^2 - T^a \Delta^\mu g_s \bar{\gamma}\cdot\Delta(\Delta\cdot\overline{p})(\Delta\cdot\overline{q})$$

2-gluon Feynman rules (unpolarized).

```
Lagrangian["ogu"]
FeynRule[%, {QuantumField[GaugeField, {\[Mu]}, {a}][p],
QuantumField[GaugeField, {\[Nu]},
      {b}][q]}, ZeroMomentumInsertion -> False] // Factor
```

$$\frac{1}{2} i^{m-1} F^{\text{FCGV(a)}}_{\text{FCGV}(\alpha)\Delta}.\left(D_\Delta^{\text{FCGV(a)FCGV(b)}}\right)^{m-2}.F^{\text{FCGV(b)}}_{\text{FCGV}(\alpha)\Delta}$$

$$-i^m \delta^{ab}\left(\vec{\partial}_\Delta\right)^{m-2}\left(-\bar{g}^{\mu\nu}(\Delta\cdot\overline{p})(\Delta\cdot\overline{q})+\Delta^\nu\overline{q}^\mu(\Delta\cdot\overline{p})+\Delta^\mu\overline{p}^\nu(\Delta\cdot\overline{q})-\Delta^\mu\Delta^\nu(\overline{p}\cdot\overline{q})\right)$$

2-gluon Feynman rules (polarized).

```
Lagrangian["ogp"]
FeynRule[%, {QuantumField[GaugeField, {\[Mu]}, {a}][p],
QuantumField[GaugeField, {\[Nu]},
       {b}][q]}, ZeroMomentumInsertion -> False] // Factor

Factor2[Calc[% /. p -> -q, Assumptions -> Automatic]]
```

$$\frac{1}{2} i^m \bar{\epsilon}^{\text{FCGV}(\alpha)\text{FCGV}(\beta)\text{FCGV}(\gamma)\Delta} . F^{\text{FCGV(a)}}_{\text{FCGV}(\beta)\text{FCGV}(\gamma)} \cdot \left(D_\Delta^{\text{FCGV(a)FCGV(b)}} \right)^{m-2} . F^{\text{FCGV(b)}}_{\text{FCGV}(\alpha)\Delta}$$

$$i^{m+1} \delta^{ab} \left(\vec{\partial}_\Delta \right)^{m-2} \left(\Delta^\mu \bar{\epsilon}^{\nu\Delta\overline{pq}} - \Delta^\nu \bar{\epsilon}^{\mu\Delta\overline{pq}} - (\Delta \cdot \overline{p}) \, \bar{\epsilon}^{\mu\nu\Delta\overline{q}} + (\Delta \cdot \overline{q}) \, \bar{\epsilon}^{\mu\nu\Delta\overline{p}} \right)$$

0

Compare with the Feynman rule tabulated in Twist2GluonOperator.

```
Twist2GluonOperator[q, {\[Mu], a}, {\[Nu], b}, Polarization -> 1, Explicit
-> True]
```

$$i \left(1 - (-1)^m\right) \delta^{ab} (\Delta \cdot q)^{m-1} \epsilon^{\mu\nu\Delta q}$$

quark-quark -gluon-gluon Feynman rule (unpolarized).

```
Lagrangian["oqu"]

frule = FeynRule[%, {QuantumField[QuarkField][p],
QuantumField[AntiQuarkField][q],
     QuantumField[GaugeField, {\[Mu]}, {a}][r], QuantumField[GaugeField,
{\[Nu]}, {b}][s]},
    ZeroMomentumInsertion -> True, InitialFunction -> Identity];
```

$$i^m \bar{\psi} . (\bar{\gamma} \cdot \Delta) . {D_\Delta}^{m-1} . \psi$$

DiracTrick: Error! DiracTrick has encountered a fatal problem and must abort the computation. The problem reads:
Unexpanded nested DOTs detected!

$Aborted

```
LeafCount[frule]
```

1

```
Twist2QuarkOperator[{p}, {q}, {r, \[Mu], a}, {s, \[Nu], b}, Polarization ->
0]
```

$$-(-1)^m \Delta^\mu \Delta^\nu g_s^2 (\bar{\gamma}\cdot\Delta).\left(T^a.T^b\left(\sum_{i=0}^{-3+m}(i+1)(\Delta\cdot q)^j(-(\Delta\cdot p))^{-i+m-3}(\Delta\cdot q+\Delta\cdot r)^{i-j}\right)\right.$$
$$\left.+T^b.T^a\left(\sum_{i=0}^{-3+m}(i+1)(\Delta\cdot q)^j(-(\Delta\cdot p))^{-i+m-3}(\Delta\cdot q+\Delta\cdot s)^{i-j}\right)\right)$$

```
(*Twist2QuarkOperator[{p},{q},{r,\[Mu],a},{s,\[Nu],b},Polarization->0]
Calc[frule-%%/.OPEm->5/.s->-p-q-r/.D->4]*)
```

11.18 FieldDerivative

FieldDerivative[f[x], x, li1, li2, ...] is the derivative of **f[x]** with respect to space-time variables **x** and with Lorentz indices **li1, li2, ...**, where **li1, li2, ...** have head **LorentzIndex**.

FieldDerivative[f[x], x, li1, li2, ...] can be given as **FieldDerivative[f[x], x, {l1, l2, ...}]**, where $l1$ is $li1$ without the head.

FieldDerivative is defined only for objects with head **QuantumField**. If the space-time derivative of other objects is wanted, the corresponding rule must be specified.

11.18.1 See also

Overview, FCPartialD, ExpandPartialD.

11.18.2 Examples

```
QuantumField[A, {\[Mu]}][x] . QuantumField[B, {\[Nu]}][y] . QuantumField[C,
{\[Rho]}][x] . QuantumField[D, {\[Sigma]}][y]
```

$$A_\mu(x).B_\nu(y).C_\rho(x).D_\sigma(y)$$

```
FieldDerivative[%, x, {\[Mu]}] // DotExpand
```

$$A_\mu(x).B_\nu(y).\left(\left(\partial_\mu C_\rho\right)\right)(x).D_\sigma(y)+\left(\left(\partial_\mu A_\mu\right)\right)(x).B_\nu(y).C_\rho(x).D_\sigma(y)$$

```
FieldDerivative[%%, y, {\[Nu]}] // DotExpand
```

$$A_\mu(x).B_\nu(y).C_\rho(x).\left(\left(\partial_\nu D_\sigma\right)\right)(y)+A_\mu(x).\left(\left(\partial_\nu B_\nu\right)\right)(y).C_\rho(x).D_\sigma(y)$$

11.19 FDr

FDr is the shorthand notation for **FieldDerivative**.

11.19.1 See also

Overview, FieldDerivative.

11.19.2 Examples

11.20 FieldStrength

FieldStrength[mu, nu, a] is the field strength tensor $\partial_\mu A^a_\nu - \partial_\nu A^a_\mu + g_s A^b_\mu A^c_\nu f^{abc}$.

FieldStrength[mu, nu] is the field strength tensor $(\partial_\mu A_\nu - \partial_\nu A_\mu)$.

The name of the field (A) and the coupling constant (g) can be set through the options or by additional arguments. The first two indices are interpreted as type **LorentzIndex**, except **OPEDelta**, which is converted to **Momentum[OPEDelta]**.

11.20.1 See also

Overview

11.20.2 Examples

```
FieldStrength[\[Mu], \[Nu]]
```

$$F_{\mu\nu}$$

```
FieldStrength[\[Mu], \[Nu], a]
```

$$F^a_{\mu\nu}$$

```
FieldStrength[\[Mu], \[Nu], Explicit -> True]
```

$$(\partial_\mu A_\nu) - (\partial_\nu A_\mu)$$

```
FieldStrength[\[Mu], \[Nu], a, Explicit -> True]
```

$$g_s f^{ab19\ c20} A_\mu^{b19}.A_\nu^{c20} + (\partial_\mu A_\nu^a) - (\partial_\nu A_\mu^a)$$

```
FieldStrength[\[Mu], \[Nu], a, CouplingConstant -> -SMP["g_s"], Explicit ->
True]
```

$$-g_s f^{ab21\ c22} A_\mu^{b21}.A_\nu^{c22} + (\partial_\mu A_\nu^a) - (\partial_\nu A_\mu^a)$$

11.21 FunctionalD

FunctionalD[exp, {QuantumField[name, LorentzIndex[mu], ..., SUNIndex[a]][p], ...}] calculates the functional derivative of exp with respect to the QuantumField list (with incoming momenta p, etc.) and does the Fourier transform.

FunctionalD[expr, {QuantumField[name, LorentzIndex[mu], ... SUNIndex[a]], ...}] calculates the functional derivative and does partial integration but omits the x-space delta functions.

FunctionalD is a low level function used in **FeynRule**.

11.21.1 See also

Overview, FeynRule, QuantumField.

11.21.2 Examples

Instead of the usual $\delta\phi(x)/\delta\phi(y) = \delta^{(D)}(x-y)$ the arguments and the δ function are omitted, i.e. for the program for simplicity: $\delta\phi/\delta\phi = 1$.

```
FunctionalD[QuantumField[\[Phi]], QuantumField[\[Phi]]]
```

$$1$$

```
FunctionalD[QuantumField[\[Phi]]^2, QuantumField[\[Phi]]]
```

$$2\phi$$

Instead of the usual $(\delta\partial_\mu\phi(x))/\delta\phi(y) = \partial_\mu\delta^{(D)}(x-y)$ the arguments are omitted, and the ∂_μ operator is specified by default to be an integration by parts operator, i.e. the right hand side will be just **Null** or, more precisely, (by default) $-\partial_{mu}$.

```
FunctionalD[QuantumField[FCPartialD[LorentzIndex[\[Mu]]], \[Phi]],
QuantumField[\[Phi]]]
```

$$-\vec{\delta}_{\mu}$$

```
(QuantumField[FCPartialD[LorentzIndex[\[Mu]]], \[Phi]] .
QuantumField[FCPartialD[LorentzIndex[\[Mu]]]
       , \[Phi]] - m^2 QuantumField[\[Phi]] . QuantumField[\[Phi]])/2

FunctionalD[%, QuantumField[\[Phi]]]
```

$$\frac{1}{2}\left(((\partial_{\mu}\phi)).((\partial_{\mu}\phi)) - m^2\phi.\phi\right)$$

$$m^2(-\phi) - (\partial_{\mu}\partial_{\mu}\phi)$$

Consider $S[A] = -\int d^D x \frac{1}{4} F_a^{\mu\nu}(x) F_{\mu\nu}(x)$

First approach:

```
F1 = FieldStrength[\[Mu], \[Nu], a, {A, b, c}, 1, Explicit -> True]

F2 = FieldStrength[\[Mu], \[Nu], a, {A, d, e}, 1, Explicit -> True]

S[A] = -1/4 F1 . F2
```

$$f^{abc} A_{\mu}^{b}.A_{\nu}^{c} + (\partial_{\mu}A_{\nu}^{a}) - (\partial_{\nu}A_{\mu}^{a})$$

$$f^{ade} A_{\mu}^{d}.A_{\nu}^{e} + (\partial_{\mu}A_{\nu}^{a}) - (\partial_{\nu}A_{\mu}^{a})$$

$$-\frac{1}{4}\left(f^{abc} A_{\mu}^{b}.A_{\nu}^{c} + (\partial_{\mu}A_{\nu}^{a}) - (\partial_{\nu}A_{\mu}^{a})\right).\left(f^{ade} A_{\mu}^{d}.A_{\nu}^{e} + (\partial_{\mu}A_{\nu}^{a}) - (\partial_{\nu}A_{\mu}^{a})\right)$$

In order to derive the equation of motion, the functional derivative of S with respect to A_{σ}^{g} has to be set to zero. Bearing in mind that for FeynCalc we have to be precise as to where which operators (coming from the substitution of the derivative of the delta function) act.

Act with the functional derivative operator on the first field strength:

$0 = (\delta S)/(\delta A_{\sigma}^{g}(y)) = -2/4 \int d^D x (\delta/(\delta A_{\sigma}^{g}(y)) F_{\mu\nu}^{a}(x)) F_a^{\mu\nu}(x)$

See what happens with just $(\delta S[A])/(\delta A_{\sigma}^{g})$

```
Ag = QuantumField[A, {\[Sigma]}, {g}]
```

$$A^{g}_{\sigma}$$

```
FunctionalD[F1, Ag]
```

$$A^{b}_{\mu}\bar{g}^{\nu\sigma}f^{abg} - A^{c}_{\nu}\bar{g}^{\mu\sigma}f^{acg} + \delta^{ag}\vec{\partial}_{\mu}(-\bar{g}^{\nu\sigma}) + \delta^{ag}\vec{\partial}_{\nu}\bar{g}^{\mu\sigma}$$

Use **FCCanonicalizeDummyIndices** to minimize the number of dummy indices.

```
t1 = FCCanonicalizeDummyIndices[%, SUNIndexNames -> {c1}] /. c1 -> c
```

$$A^{c}_{\mu}\bar{g}^{\nu\sigma}f^{acg} - A^{c}_{\nu}\bar{g}^{\mu\sigma}f^{acg} + \delta^{ag}\vec{\partial}_{\mu}(-\bar{g}^{\nu\sigma}) + \delta^{ag}\vec{\partial}_{\nu}\bar{g}^{\mu\sigma}$$

Instead of inserting the definition for the second $F^{\mu\nu}_{a}$, introduce a **QuantumField** object with antisymmetry built into the Lorentz indices:

```
F /: QuantumField[pard___, F, \[Beta]_, \[Alpha]_, s_] :=
-QuantumField[pard, F, \[Alpha], \[Beta], s] /; ! OrderedQ[{\[Beta],
\[Alpha]}]
```

```
QuantumField[F, {\[Mu], \[Nu]}, {a}]

% /. {\[Mu] :> \[Nu], \[Nu] :> \[Mu]}
```

$$F^{a}_{\mu\nu}$$

$$-F^{a}_{\mu\nu}$$

```
t2 = Contract[ExpandPartialD[-1/2 t1 . QuantumField[F, LorentzIndex[\[Mu]],
LorentzIndex[\[Nu]],
        SUNIndex[a]]]] /. Dot -> Times
```

$$-\frac{1}{2}A^{c}_{\mu}f^{acg}F^{a}_{\mu\sigma} - \frac{1}{2}A^{c}_{\nu}f^{acg}F^{a}_{\nu\sigma} + \frac{1}{2}\left(\left(\partial_{\mu}F^{g}_{\mu\sigma}\right)\right) + \frac{1}{2}\left(\left(\partial_{\nu}F^{g}_{\nu\sigma}\right)\right)$$

```
t3 = FCCanonicalizeDummyIndices[t2, LorentzIndexNames -> {mu},
SUNIndexNames -> {aa,
      cc}] /. {mu -> \[Mu], aa -> a, cc -> c}
```

$$A_{\mu}^{a} f^{acg} F_{\mu\sigma}^{c} + \left(\partial_{\mu} F_{\mu\sigma}^{g}\right)$$

```
t4 = FCE[t3] /. SUNF[a, c, g] -> -SUNF[g, c, a]
```

$$\left(\partial_{\mu} F_{\mu\sigma}^{g}\right) - A_{\mu}^{a} f^{gca} F_{\mu\sigma}^{c}$$

Since the variational derivative vanishes **t4** implies that $0 = D_{\mu} F_{g}^{\mu\sigma}$

Second approach:

It is of course also possible to do the functional derivative on the $S[A]$ with both field strength tensors inserted.

```
S[A]
```

$$-\frac{1}{4}\left(f^{abc} A_{\mu}^{b}.A_{\nu}^{c} + \left(\partial_{\mu} A_{\nu}^{a}\right) - \left(\partial_{\nu} A_{\mu}^{a}\right)\right).\left(f^{ade} A_{\mu}^{d}.A_{\nu}^{e} + \left(\partial_{\mu} A_{\nu}^{a}\right) - \left(\partial_{\nu} A_{\mu}^{a}\right)\right)$$

```
r1 = FunctionalD[S[A], Ag]
```

$$-\frac{1}{4} f^{abc} f^{adg} A_{\mu}^{b}.A_{\sigma}^{c}.A_{\mu}^{d} + \frac{1}{4} f^{abc} f^{aeg} A_{\sigma}^{b}.A_{\nu}^{c}.A_{\nu}^{e} - \frac{1}{4} f^{abg} f^{ade} A_{\mu}^{b}.A_{\mu}^{d}.A_{\sigma}^{e}$$
$$-\frac{1}{4} f^{abg} A_{\mu}^{b}.\left(\left(\partial_{\mu} A_{\sigma}^{a}\right)\right) + \frac{1}{4} f^{abg} A_{\mu}^{b}.\left(\left(\partial_{\sigma} A_{\mu}^{a}\right)\right) + \frac{1}{4} f^{acg} f^{ade} A_{\nu}^{c}.A_{\sigma}^{d}.A_{\nu}^{e}$$
$$-\frac{1}{4} f^{acg} A_{\nu}^{c}.\left(\left(\partial_{\nu} A_{\sigma}^{a}\right)\right) + \frac{1}{4} f^{acg} A_{\nu}^{c}.\left(\left(\partial_{\sigma} A_{\nu}^{a}\right)\right) - \frac{1}{4} f^{adg}\left(\left(\partial_{\mu} A_{\sigma}^{a}\right)\right).A_{\mu}^{d} + \frac{1}{4} f^{adg}\left(\left(\partial_{\sigma} A_{\mu}^{a}\right)\right).A_{\mu}^{d} - \frac{1}{4} f^{aeg}\left(\left(\partial_{\nu} A_{\sigma}^{a}\right)\right).A_{\nu}^{e} + \frac{1}{4} f^{aeg}$$

This is just functional derivation and partial integration and simple contraction of indices. At first no attempt is made to rename dummy indices (since this is difficult in general).

With a general replacement rule only valid for commuting fields the color indices can be canonicalized a bit more. The idea is to use the commutative properties of the vector fields, and canonicalize the color indices by a trick.

```
Commutator[QuantumField[aaa___FCPartialD, A, bbb__],
QuantumField[ccc___FCPartialD, A,
      ddd__]] = 0;

r2 = r1 // DotSimplify // FCCanonicalizeDummyIndices[#, SUNIndexNames ->
{a1, b1, c1, d1},
       LorentzIndexNames -> {mu, nu, rho}] & // ReplaceAll[#, {a1 -> a, b1
-> b, c1 -> c,
       d1 -> d, mu -> \[Mu], nu -> \[Nu], rho -> \[Rho]}] & // Collect2[#,
SUNF] &
```

$$\frac{1}{2} f^{adg} f^{bcd} A^a_\mu.A^b_\mu.A^c_\sigma + \frac{1}{2} f^{acd} f^{bdg} A^a_\mu.A^b_\mu.A^c_\sigma + f^{abg} \left(2A^a_\mu.\left(\left(\partial_\mu A^b_\sigma\right)\right) - A^a_\mu.\left(\left(\partial_\sigma A^b_\mu\right)\right) - A^a_\sigma.\left(\left(\partial_\mu A^b_\mu\right)\right)\right) + \left(\partial_\mu \partial_\mu A^g_\sigma\right) - \left(\partial_\mu \partial_\sigma A^g_\mu\right)$$

Inspection reveals that still terms are the same. Gather the terms with two **f**'s:

```
twof = Select[r2 // Expand, Count[#, SUNF[__]] === 2 &]
```

$$\frac{1}{2} f^{adg} f^{bcd} A^a_\mu.A^b_\mu.A^c_\sigma + \frac{1}{2} f^{acd} f^{bdg} A^a_\mu.A^b_\mu.A^c_\sigma$$

```
twofnew = ((twof[[1]] /. {a -> b, b -> a}) + twof[[2]]) // DotSimplify
```

$$f^{acd} f^{bdg} A^a_\mu.A^b_\mu.A^c_\sigma$$

```
r3 = r2 - twof + twofnew
```

$$f^{acd} f^{bdg} A^a_\mu.A^b_\mu.A^c_\sigma + f^{abg} \left(2A^a_\mu.\left(\left(\partial_\mu A^b_\sigma\right)\right) - A^a_\mu.\left(\left(\partial_\sigma A^b_\mu\right)\right) - A^a_\sigma.\left(\left(\partial_\mu A^b_\mu\right)\right)\right) + \left(\partial_\mu \partial_\mu A^g_\sigma\right) - \left(\partial_\mu \partial_\sigma A^g_\mu\right)$$

Check that this is now indeed the same as the **t4** result from the first attempt.

```
t4
```

$$\left(\partial_\mu F^g_{\mu\sigma}\right) - A^a_\mu f^{gca} F^c_{\mu\sigma}$$

```
w0 = RightPartialD[\[Mu]] . FieldStrength[\[Mu], \[Sigma], g, {A, a, b}, 1]
+
   QuantumField[A, LorentzIndex[\[Mu]], SUNIndex[c]] . FieldStrength[\[Mu],
\[Sigma], a,
      {A, b, d}, 1] SUNF[g, c, a]
```

$$f^{gca} A^{c}_{\mu}.F^{a\{A,b,d\}1}_{\mu\sigma} + \vec{\partial}_{\mu}.F^{g\{A,a,b\}1}_{\mu\sigma}$$

```
w1 = Explicit[w0]
```

$$f^{gca} A^{c}_{\mu}.\left(f^{abd} A^{b}_{\mu}.A^{d}_{\sigma} + (\partial_{\mu} A^{a}_{\sigma}) - (\partial_{\sigma} A^{a}_{\mu})\right) + \vec{\partial}_{\mu}.\left(f^{gab} A^{a}_{\mu}.A^{b}_{\sigma} + (\partial_{\mu} A^{g}_{\sigma}) - (\partial_{\sigma} A^{g}_{\mu})\right)$$

```
w2 = ExpandPartialD[w1] // DotSimplify
```

$$\begin{aligned} &-f^{abd} f^{acg} A^{b}_{\mu}.A^{c}_{\mu}.A^{d}_{\sigma} + f^{abg} A^{a}_{\mu}.\left((\partial_{\mu} A^{b}_{\sigma})\right) \\ &+ f^{abg} A^{b}_{\sigma}.\left((\partial_{\mu} A^{a}_{\mu})\right) - f^{acg} A^{c}_{\mu}.\left((\partial_{\mu} A^{a}_{\sigma})\right) + f^{acg} A^{c}_{\mu}.\left((\partial_{\sigma} A^{a}_{\mu})\right) + (\partial_{\mu}\partial_{\mu} A^{g}_{\sigma}) - \left(\partial_{\mu}\partial_{\sigma} A^{g}_{\mu}\right) \end{aligned}$$

```
dif1 = w2 - r3
```

$$\begin{aligned} &-f^{abd} f^{acg} A^{b}_{\mu}.A^{c}_{\mu}.A^{d}_{\sigma} - f^{acd} f^{bdg} A^{a}_{\mu}.A^{b}_{\mu}.A^{c}_{\sigma} + f^{abg} A^{a}_{\mu}.\left((\partial_{\mu} A^{b}_{\sigma})\right) \\ &-f^{abg}\left(2A^{a}_{\mu}.\left((\partial_{\mu} A^{b}_{\sigma})\right) - A^{a}_{\mu}.\left((\partial_{\sigma} A^{b}_{\mu})\right) - A^{a}_{\sigma}.\left((\partial_{\mu} A^{b}_{\mu})\right)\right) + f^{abg} A^{b}_{\sigma}.\left((\partial_{\mu} A^{a}_{\mu})\right) - f^{acg} A^{c}_{\mu}.\left((\partial_{\mu} A^{a}_{\sigma})\right) + f^{acg} A^{c}_{\mu}.\left((\partial_{\sigma} A^{a}_{\mu})\right) \end{aligned}$$

As expected:

```
dif1 // FCCanonicalizeDummyIndices
```

$$0$$

Finally, unset the commutator of the bosonic fields.

```
UnDeclareCommutator[QuantumField[aaa___FCPartialD, A, bbb__],
    QuantumField[ccc___FCPartialD, A, ddd__]] = 0;
```

11.22 GhostPropagator

GhostPropagator[p, a, b] gives the ghost propagator where **a** and **b** are the color indices.

GhostPropagator[p] omits the δ_{ab}.

GHP can be used as an abbreviation of **GhostPropagator**.

11.22.1 See also

Overview, GluonPropagator, QCDFeynmanRuleConvention, GluonGhostVertex.

11.22.2 Examples

```
GhostPropagator[p, a, b]
```

```
Explicit[%]
```

$$\Pi_{ab}(p)$$

$$\frac{i\delta^{ab}}{p^2}$$

```
GHP[p]
```

```
Explicit[%]
```

$$\Pi_{u}(p)$$

$$\frac{i}{p^2}$$

11.23 GHP

GHP[p, a, b] gives the ghost propagator where **a** and **b** are the color indices.

GHP[p] omits the δ_{ab}.

11.23.1 See also

Overview, GhostPropagator, GluonPropagator, GluonGhostVertex.

11.23.2 Examples

```
GHP[p, a, b]
```

$$\Pi_{ab}(p)$$

```
GHP[p] // Explicit
```

$$\frac{i}{p^2}$$

```
GHP[p, c1, c2]
```

$$\Pi_{\text{c1 c2}}(p)$$

```
StandardForm[FCE[GHP[-k, c3, c4] // Explicit]]

(*I FAD[k] SD[c3, c4]*)
```

11.24 GluonGhostVertex

GluonGhostVertex[{p, mu, a}, {q, nu, b}, {k, rho, c}] or **GluonGhostVertex[p, mu, a , q, nu, b , k, rho, c]** yields the Gluon-Ghost vertex. The first argument represents the gluon and the third argument the outgoing ghost field (but incoming 4-momentum).

GGV can be used as an abbreviation of **GluonGhostVertex**.The dimension and the name of the coupling constant are determined by the options **Dimension** and **CouplingConstant**.

11.24.1 See also

Overview, GluonPropagator, GluonSelfEnergy, GluonVertex, QCDFeynmanRuleConvention, GhostPropagator.

11.24.2 Examples

```
GluonGhostVertex[{p, \[Mu], a}, {q, \[Nu], b}, {k, \[Rho], c}]

Explicit[%]
```

$$\tilde{\Lambda}^{\mu}(k) f^{abc}$$

$$g_s\,(-k^{\mu})\,f^{abc}$$

11.25 GGV

GGV is equivalent to **GluonGhostVertex**.

11.25.1 See also

Overview, GluonGhostVertex.

11.25.2 Examples

11.26 GluonPropagator

GluonPropagator[p, {mu, a}, {nu, b}] or **GluonPropagator[p, mu, a, nu, b]** yields the gluon propagator.

GluonPropagator[p, {mu}, {nu}] or **GluonPropagator[p, mu, nu]** omits the **SUNDelta**.

GP can be used as an abbreviation of **GluonPropagator**.

The gauge and the dimension are determined by the options **Gauge** and **Dimension**. The following settings of **Gauge** are possible:

- **1** for the Feynman gauge
- **alpha** for the general covariant gauge
- **{Momentum[n] , 1}** for the axial gauge

11.26.1 See also

Overview, GluonSelfEnergy, GluonVertex, GluonGhostVertex, GhostPropagator, GluonGhostVertex.

11.26.2 Examples

```
GluonPropagator[p, \[Mu], a, \[Nu], b]

Explicit[%]
```

$$\Pi^{\mu\nu}_{ab}(p)$$

$$-\frac{i\delta^{ab}g^{\mu\nu}}{p^2}$$

```
GP[p, \[Mu], a, \[Nu], b, Gauge -> \[Alpha]]

Explicit[%]
```

$$\Pi_{ab}^{\mu\nu}(p)$$

$$\frac{i\delta^{ab}\left(\frac{(1-\alpha)p^\mu p^\nu}{p^2}-g^{\mu\nu}\right)}{p^2}$$

```
GluonPropagator[p, \[Mu], a, \[Nu], b, Gauge -> {Momentum[n], 1}, Explicit
-> True]
```

$$\frac{i\delta^{ab}\left(\frac{p^\mu\bar{n}^\nu+p^\nu\bar{n}^\mu}{(\bar{n}\cdot p+i\eta)}-\frac{\bar{n}^2p^\mu p^\nu-p^2\bar{n}^\mu\bar{n}^\nu}{(\bar{n}\cdot p+i\eta)^{21}}-g^{\mu\nu}\right)}{p^2}$$

```
GP[p, \[Mu], \[Nu]]
```

$$\Pi_{g}^{\mu\nu}(p)$$

```
Explicit[%]
```

$$-\frac{ig^{\mu\nu}}{p^2}$$

```
GluonPropagator[p, \[Mu], a, \[Nu], b, CounterTerm -> 1] // Explicit
```

$$-\frac{iC_Ag_s^2S_n\delta^{ab}\left(\frac{11p^\mu p^\nu}{3}-\frac{19}{6}p^2g^{\mu\nu}\right)}{\varepsilon}$$

```
GluonPropagator[p, \[Mu], a, \[Nu], b, CounterTerm -> 2] // Explicit
```

$$-\frac{iC_Ag_s^2S_n\delta^{ab}\left(-\frac{1}{6}p^2g^{\mu\nu}-\frac{1}{3}p^\mu p^\nu\right)}{\varepsilon}$$

```
GluonPropagator[p, \[Mu], a, \[Nu], b, CounterTerm -> 3] // Explicit
```

$$-\frac{2iT_f\,g_s^2S_n\delta^{ab}\left(\frac{4}{3}p^2g^{\mu\nu}-\frac{4p^\mu p^\nu}{3}\right)}{\varepsilon}$$

```
GluonPropagator[p, \[Mu], a, \[Nu], b, CounterTerm -> 4] // Explicit
```

$$-\frac{iC_A g_s^2 S_n \delta^{ab} \left(\frac{10 p^\mu p^\nu}{3} - \frac{10}{3} p^2 g^{\mu\nu}\right)}{\varepsilon}$$

```
GluonPropagator[p, \[Mu], a, \[Nu], b, CounterTerm -> 5] // Explicit
```

$$\frac{iC_A g_s^2 S_n \delta^{ab} \left(\frac{10 p^\mu p^\nu}{3} - \frac{10}{3} p^2 g^{\mu\nu}\right)}{\varepsilon} + \frac{iT_f g_s^2 S_n \delta^{ab} \left(\frac{4}{3} p^2 g^{\mu\nu} - \frac{4 p^\mu p^\nu}{3}\right)}{\varepsilon}$$

11.27 GP

GP is equivalent to **GluonPropagator**.

11.27.1 See also

Overview, GluonPropagator.

11.27.2 Examples

11.28 GluonSelfEnergy

GluonSelfEnergy[{mu, a}, {nu, b}] yields the 1-loop gluon self-energy.

11.28.1 See also

Overview, GluonPropagator, GluonGhostVertex, GluonVertex, GhostPropagator.

11.28.2 Examples

```
GluonSelfEnergy[{\[Mu], a}, {\[Nu], b}, Momentum -> p]
```

$$\frac{1}{2} i \left(\frac{20}{3\varepsilon} - \frac{62}{9}\right) C_A g_s^2 \delta^{ab} \left(p^\mu p^\nu - p^2 g^{\mu\nu}\right) + i \left(\frac{20}{9} - \frac{8}{3\varepsilon}\right) T_f g_s^2 \delta^{ab} \left(p^\mu p^\nu - p^2 g^{\mu\nu}\right)$$

```
GluonSelfEnergy[{\[Mu], a}, {\[Nu], b}, Gauge -> \[Xi], Momentum -> q]
```

$$\frac{1}{2} i C_A \left(\frac{2\left(\frac{13}{3} - \xi\right)}{\varepsilon} - \frac{1}{2}(1-\xi)^2 + 2(1-\xi) - \frac{62}{9}\right) g_s^2 \delta^{ab} \left(q^\mu q^\nu - q^2 g^{\mu\nu}\right) + i \left(\frac{20}{9} - \frac{8}{3\varepsilon}\right) T_f g_s^2 \delta^{ab} \left(q^\mu q^\nu - q^2 g^{\mu\nu}\right)$$

11.29 GluonVertex

GluonVertex[{p, mu, a}, {q, nu, b}, {k, la, c}] or **GluonVertex[p, mu, a, q, nu, b, k, la, c]** yields the 3-gluon vertex.

GluonVertex[{p, mu}, {q, nu}, {k, la}] yields the 3-gluon vertex without color structure and the coupling constant.

GluonVertex[{p, mu, a}, {q, nu, b}, {k, la, c}, {s, si, d}] or **GluonVertex[{mu, a}, {nu, b}, {la, c}, {si, d}]** or **GluonVertex[p, mu, a, q, nu, b, k, la, c , s, si, d]** or **GluonVertex[mu, a, nu, b, la, c, si, d]** yields the 4-gluon vertex.

GV can be used as an abbreviation of **GluonVertex**.

The dimension and the name of the coupling constant are determined by the options **Dimension** and **CouplingConstant**. All momenta are flowing into the vertex.

11.29.1 See also

Overview, GluonPropagator, GluonGhostVertex.

11.29.2 Examples

```
GluonVertex[{p, \[Mu], a}, {q, \[Nu], b}, {r, \[Rho], c}]

Explicit[%]
```

$$f^{abc} V^{\mu\nu\rho}(p, q, r)$$

$$g_s f^{abc} \left(g^{\mu\nu} \left(p^{\rho} - q^{\rho}\right) + g^{\mu\rho} \left(r^{\nu} - p^{\nu}\right) + g^{\nu\rho} \left(q^{\mu} - r^{\mu}\right)\right)$$

```
GV[{p, \[Mu]}, {q, \[Nu]}, {r, \[Rho]}]

Explicit[%]
```

$$V^{\mu\nu\rho}(p, q, r)$$

$$g_s \left(g^{\mu\nu} \left(p^{\rho} - q^{\rho}\right) + g^{\mu\rho} \left(r^{\nu} - p^{\nu}\right) + g^{\nu\rho} \left(q^{\mu} - r^{\mu}\right)\right)$$

```
GluonVertex[{p, \[Mu], a}, {q, \[Nu], b}, {r, \[Rho], c}, {s, \[Sigma], d}]

Explicit[%]
```

$$V_{abcd}^{\mu\nu\rho\sigma}(p, q, r, s)$$

$$-ig_s^2 \left(f^{adFCGV(u19)} f^{bcFCGV(u19)} \left(g^{\mu\nu} g^{\rho\sigma} - g^{\mu\rho} g^{\nu\sigma}\right) + f^{acFCGV(u19)} f^{bdFCGV(u19)} \left(g^{\mu\nu} g^{\rho\sigma} - g^{\mu\sigma} g^{\nu\rho}\right) + f^{abFCGV(u19)} f^{cdFCGV(u19)} \left(g^{\mu\rho} g^{\nu\sigma} - g^{\mu\sigma} g^{\nu\rho}\right) \right)$$

```
GV[{\[Mu], a}, {\[Nu], b}, {\[Rho], c}, {\[Sigma], d}]

Explicit[%]
```

$$V^{abcd}$$

$$-ig_s^2 \left(f^{adFCGV(u20)} f^{bcFCGV(u20)} \left(g^{\mu\nu} g^{\rho\sigma} - g^{\mu\rho} g^{\nu\sigma}\right) + f^{acFCGV(u20)} f^{bdFCGV(u20)} \left(g^{\mu\nu} g^{\rho\sigma} - g^{\mu\sigma} g^{\nu\rho}\right) + f^{abFCGV(u20)} f^{cdFCGV(u20)} \left(g^{\mu\rho} g^{\nu\sigma} - g^{\mu\sigma} g^{\nu\rho}\right) \right)$$

11.30 GV

GV is equivalent to **GluonVertex**.

11.30.1 See also

Overview, GluonVertex.

11.30.2 Examples

11.31 QuarkGluonVertex

QuarkGluonVertex[mu, a] gives the Feynman rule for the quark-gluon vertex.

QGV can be used as an abbreviation of **QuarkGluonVertex**.

The dimension and the name of the coupling constant are determined by the options **Dimension** and **CouplingConstant**.

11.31.1 See also

Overview, GluonVertex.

11.31.2 Examples

```
QuarkGluonVertex[\[Mu], a, Explicit -> True]
```

$$ig_s T^a.\gamma^\mu$$

```
QGV[\[Mu], a]
```

$$Q_a^\mu$$

```
Explicit[%]
```

$$ig_s T^a.\gamma^\mu$$

```
QuarkGluonVertex[\[Mu], a, CounterTerm -> 1, Explicit -> True]
```

$$\frac{2ig_s^3 S_n \left(C_F - \frac{C_A}{2}\right) T^a.\gamma^\mu}{\varepsilon}$$

```
QuarkGluonVertex[\[Mu], a, CounterTerm -> 2, Explicit -> True]
```

$$\frac{3iC_A g_s^3 S_n T^a.\gamma^\mu}{\varepsilon}$$

```
QuarkGluonVertex[\[Mu], a, CounterTerm -> 3, Explicit -> True]
```

$$\frac{2ig_s^3 S_n \left(C_A + C_F\right) T^a.\gamma^\mu}{\varepsilon}$$

```
QuarkGluonVertex[{p, \[Mu], a}, {q}, {k}, OPE -> True, Explicit -> True]
```

$$\Omega\Delta^\mu g_s(\gamma\cdot\Delta).T^a\left(\sum_{i=0}^{-2+m}(-1)^i(k\cdot\Delta)^i(\Delta\cdot q)^{-2-i+m}\right)+ig_s T^a.\gamma^\mu$$

```
QuarkGluonVertex[{p, \[Mu], a}, {q}, {k}, OPE -> False, Explicit -> True]
```

$$ig_s T^a.\gamma^\mu$$

11.32 QGV

QGV is equivalent to **QuarkGluonVertex**.

11.32.1 See also

Overview, QuarkGluonVertex.

11.32.2 Examples

11.33 QuarkPropagator

QuarkPropagator[p] is the massless quark propagator.

QuarkPropagator[{p, m}] gives the quark propagator with mass m.

QP can be used as an abbreviation of **QuarkPropagator**.

11.33.1 See also

Overview, GluonPropagator, QuarkGluonVertex.

11.33.2 Examples

```
QuarkPropagator[p, Explicit -> True]
```

$$\frac{i\gamma \cdot p}{p^2}$$

```
QuarkPropagator[{p, m}, Explicit -> True]
```

$$\frac{i(m + \gamma \cdot p)}{p^2 - m^2}$$

```
QP[{p, m}]
```

```
Explicit[%]
```

$$\Pi_q(p)$$

$$\frac{i(m + \gamma \cdot p)}{p^2 - m^2}$$

11.34 QP

QP is an alias for **QuarkPropagator**.

QP[p] is the massless quark propagator.

QP[{p, m}] gives the quark propagator with mass **m**.

11.34.1 See also

Overview, QuarkPropagator.

11.34.2 Examples

11.35 ScalarGluonVertex

ScalarGluonVertex[{p}, {q}, {mu, a}] or **ScalarGluonVertex[p, q, mu, a]** yields the scalar-scalar-gluon vertex, where **p** and **q** are incoming momenta.

ScalarGluonVertex[{mu, a}, {nu, b}] yields the scalar-scalar-gluon-gluon vertex, where **p** and **q** are incoming momenta.

The dimension and the name of the coupling constant are determined by the options **Dimension** and **CouplingConstant**.

11.35.1 See also

Overview

11.35.2 Examples

```
ScalarGluonVertex[{p}, {q}, {\[Mu], a}]
```

$$iT^a g_s (p - q)^\mu$$

11.36 ShiftPartialD

ShiftPartialD[exp, {FCPartialD[i1], FCPartialD[i2], ...}, field] uses integration-by-parts identities to shift the derivatives of **QuantumField**s such, that a term containing derivatives with indices **i1, i2, ...** acting on **field** is eliminated from the final expression.

Notice that one must explicitly specify the type of the indices, e.g. by writing **FCPartialD[LorentzIndex[mu]]** or **FCPartialD[CartesianIndex[i]]**. Furthermore, the function always assumes that the surface term vanishes.

Often, when dealing with large expressions one would to integrate by parts only certain terms but not every term containing given fields and derivatives. In such situation one can specify a filter function via the option **Select**.

11.36.1 See also

Overview, ExplicitPartialD, ExpandPartialD, QuantumField

11.36.2 Examples

```
exp1 = QuantumField[QuarkFieldPsiDagger, PauliIndex[di1]] .
RightPartialD[CartesianIndex[i
    ]] . QuantumField[\[Phi]] . RightPartialD[CartesianIndex[j]] .
    QuantumField[QuarkFieldPsi, PauliIndex[di2]]
```

$$\psi^{\dagger\,\mathrm{di1}}.\vec{\partial}_i.\phi.\vec{\partial}_j.\psi^{\mathrm{di2}}$$

```
exp1 // ExpandPartialD
```

$$\psi^{\dagger\,\mathrm{di1}}.\phi.\left(\partial_i\partial_j\psi^{\mathrm{di2}}\right)+\psi^{\dagger\,\mathrm{di1}}.\left(\partial_i\phi\right).\left(\partial_j\psi^{\mathrm{di2}}\right)$$

```
ShiftPartialD[exp1, {FCPartialD[CartesianIndex[i]]}, QuarkFieldPsi,
FCVerbose -> -1]
```

$$-\left(\partial_i\psi^{\dagger\,\mathrm{di1}}\right).\phi.\left(\partial_j\psi^{\mathrm{di2}}\right)$$

This expression vanishes if one integrates by parts the term containing $\partial_\mu A_\nu$

```
exp2 = QuantumField[GaugeField, LorentzIndex[nu]] .
QuantumField[FCPartialD[LorentzIndex[mu]], FCPartialD[LorentzIndex[mu]],
     FCPartialD[LorentzIndex[rho]], FCPartialD[LorentzIndex[rho]],
FCPartialD[LorentzIndex[nu]], FCPartialD[LorentzIndex[tau]],
     GaugeField, LorentzIndex[tau]] +
QuantumField[FCPartialD[LorentzIndex[mu]], GaugeField, LorentzIndex[nu]] .
QuantumField[FCPartialD[LorentzIndex[rho]],
     FCPartialD[LorentzIndex[rho]], FCPartialD[LorentzIndex[mu]],
FCPartialD[LorentzIndex[nu]], FCPartialD[LorentzIndex[tau]], GaugeField,
LorentzIndex[tau]]
```

$$A_{\text{nu}} . (\partial_{\text{mu}}\partial_{\text{mu}}\partial_{\text{rho}}\partial_{\text{rho}}\partial_{\text{nu}}\partial_{\text{tau}}A_{\text{tau}}) + (\partial_{\text{mu}}A_{\text{nu}}) . (\partial_{\text{rho}}\partial_{\text{rho}}\partial_{\text{mu}}\partial_{\text{nu}}\partial_{\text{tau}}A_{\text{tau}})$$

By default **`ShiftPartialD`** will also apply IBPs to the other term, which is not useful here

```
ShiftPartialD[exp2, {FCPartialD[LorentzIndex[mu]]}, GaugeField]
```

Applying the following IBP relation: $\{(\partial_{\text{mu}}A_{\text{nu}}) . (\partial_{\text{rho}}\partial_{\text{rho}}\partial_{\text{mu}}\partial_{\text{nu}}\partial_{\text{tau}}A_{\text{tau}}) \to -A_{\text{nu}} . (\partial_{\text{mu}}\partial_{\text{mu}}\partial_{\text{rho}}\partial_{\text{rho}}\partial_{\text{nu}}\partial_{\text{tau}}A_{\text{tau}})\}$

Applying the following IBP relation: $\{A_{\text{nu}} . (\partial_{\text{mu}}\partial_{\text{mu}}\partial_{\text{rho}}\partial_{\text{rho}}\partial_{\text{nu}}\partial_{\text{tau}}A_{\text{tau}}) \to -2\,(\partial_{\text{mu}}A_{\text{nu}}) . (\partial_{\text{rho}}\partial_{\text{rho}}\partial_{\text{mu}}\partial_{\text{nu}}\partial_{\text{tau}}A_{\text{tau}}) - (\partial_{\text{mu}}\partial_{\text{mu}}A_{\text{nu}}) . (\partial_{\text{rho}}\partial_{\text{rho}}\partial_{\text{nu}}\partial_{\text{tau}}A_{\text{tau}})\}$

$$-A_{\text{nu}} . (\partial_{\text{mu}}\partial_{\text{mu}}\partial_{\text{rho}}\partial_{\text{rho}}\partial_{\text{nu}}\partial_{\text{tau}}A_{\text{tau}}) - 2\,(\partial_{\text{mu}}A_{\text{nu}}) . (\partial_{\text{rho}}\partial_{\text{rho}}\partial_{\text{mu}}\partial_{\text{nu}}\partial_{\text{tau}}A_{\text{tau}}) - (\partial_{\text{mu}}\partial_{\text{mu}}A_{\text{nu}}) . (\partial_{\text{rho}}\partial_{\text{rho}}\partial_{\text{nu}}\partial_{\text{tau}}A_{\text{tau}})$$

Using a suitable filter function we can readily achieve the desired result

```
ShiftPartialD[exp2, {FCPartialD[LorentzIndex[mu]]}, GaugeField, Select ->
     Function[x, FreeQ[x, QuantumField[GaugeField, LorentzIndex[nu]]]]]
```

Applying the following IBP relation: $\{(\partial_{\text{mu}}A_{\text{nu}}) . (\partial_{\text{rho}}\partial_{\text{rho}}\partial_{\text{mu}}\partial_{\text{nu}}\partial_{\text{tau}}A_{\text{tau}}) \to -A_{\text{nu}} . (\partial_{\text{mu}}\partial_{\text{mu}}\partial_{\text{rho}}\partial_{\text{rho}}\partial_{\text{nu}}\partial_{\text{tau}}A_{\text{tau}})\}$

$$0$$

11.37 SquareAmplitude

SquareAmplitude[m1, m2] multiplies the amplitudes from the list **m1** with their complex conjugate from the list **m2** to obtain the list of products $m1_i m2_j$. This function can be useful when exporting amplitudes obtained with FeynCalc to FORM.

11.37.1 See also

Overview

11.37.2 Examples

```
Clear[a1, a2, a3, b1, b2, b3]
```

```
SquareAmplitude[{a1, a2, a3}, {b1, b2, b3}]
```

$$\{\text{a1 b1}, \text{a1 b2}, \text{a1 b3}, \text{a2 b1}, \text{a2 b2}, \text{a2 b3}, \text{a3 b1}, \text{a3 b2}, \text{a3 b3}\}$$

```
SquareAmplitude[{a1, a2, a3}, {b1, b2, b3}, List -> False]
```

$$\text{a1 b1} + \text{a1 b2} + \text{a1 b3} + \text{a2 b1} + \text{a2 b2} + \text{a2 b3} + \text{a3 b1} + \text{a3 b2} + \text{a3 b3}$$

When the option **Real** is set to **True**, the amplitudes are assumed to have no imaginary part

```
SquareAmplitude[{a1, a2, a3}, {b1, b2, b3}, Real -> True, List -> False]
```

$$\text{a1 b1} + 2\,\text{a2 b1} + \text{a2 b2} + 2\,\text{a3 b1} + 2\,\text{a3 b2} + \text{a3 b3}$$

The option **Indexed** allows us to attach a marker to each contribution

```
SquareAmplitude[{a1, a2, a3}, {b1, b2, b3}, Real -> True, List -> False,
Indexed -> mark]
```

$$\text{a1 b1 mark}(1,1) + 2\,\text{a2 b1 mark}(2,1) + \text{a2 b2 mark}(2,2) + 2\,\text{a3 b1 mark}(3,1) + 2\,\text{a3 b2 mark}(3,2) + \text{a3 b3 mark}(3,3)$$

11.38 SMP

SMP[par] displays a symbol for the model parameter **par**. Typical parameters are masses, coupling constants, mixing angles etc.

Parameters that are complex, like a CKM matrix element, have an **I** as an additional argument, e.g. **SMP["V_ud", I]** and **SMP["V_ud", -I]**.

SMP[] shows the list of all available parameters.

11.38.1 See also

Overview, SMVertex, SMPToSymbol.

11.38.2 Examples

Electron mass m_e

```
SMP["m_e"]
```

$$m_e$$

Weak coupling constant g_W

```
SMP["g_W"]
```

$$g_W$$

List all available SMP's

```
SMP[]
```

N_F	N_F
m_e	m_e
m_μ	m_mu
m_τ	m_tau
m_u	m_u
m_d	m_d
m_c	m_c
m_s	m_s
m_t	m_t
m_b	m_b
m_H	m_H
m_W	m_W
m_Z	m_Z
m_q	m_q
m_Q	m_Q
m_{q_u}	m_qu
m_{q_d}	m_qd
m_l	m_l
m_π	m_pi
g	g
g_s	g_s
e	e
e_Q	e_Q
Q_u	Q_u
Q_d	Q_d
G_F	G_F
g_W	g_W
g'_W	g'_W
$\cos(\theta_W)$	cos_W
$\sin(\theta_W)$	sin_W
θ_W	theta_W
$\cos(\theta_C)$	cos_C
$\sin(\theta_C)$	sin_C
θ_C	theta_C
α	alpha_fs
α_s	alpha_s
δ_ψ	d_psi
δ_ϕ	d_phi
δ_A	d_A
δ_m	d_m
δ_u	d_u
δ_ξ	d_xi
δ_e	d_e
δ_g	d_g
Z_ψ	Z_psi
Z_ϕ	Z_phi
Z_A	Z_A
Z_m	Z_m
Z_u	Z_u
Z_ξ	Z_xi
Z_e	Z_e
Z_g	Z_g
δZ_ψ	dZ_psi
δZ_ϕ	719dZ_phi
δZ_A	dZ_A
δZ_m	dZ_m
δZ_u	dZ_u
δZ_ξ	dZ_xi

```
SMP /@ Last /@ SMP[]
```

$$\Big\{N_F, m_e, m_\mu, m_\tau, m_u, m_d, m_c, m_s, m_t, m_b, m_H, m_W, m_Z, m_q, m_Q, m_{q_u}, m_{q_d}, m_l, m_\pi, \mathrm{g}, g_s,$$
$$\mathrm{e}, e_Q, Q_u, Q_d, G_F, g_W, g'_W, \cos(\theta_W), \sin(\theta_W), \theta_W, \cos(\theta_C), \sin(\theta_C), \theta_C, \alpha, \alpha_s, \delta_\psi, \delta_\phi,$$
$$\delta_A, \delta_m, \delta_u, \delta_\xi, \delta_e, \delta_g, Z_\psi, Z_\phi, Z_A, Z_m, Z_u, Z_\xi, Z_e, Z_g, \delta Z_\psi, \delta Z_\phi, \delta Z_A, \delta Z_m, \delta Z_u, \delta Z_\xi, \delta Z_e,$$
$$\delta Z_g, \delta_\psi^{\mathrm{MS}}, \delta_\phi^{\mathrm{MS}}, \delta_A^{\mathrm{MS}}, \delta_m^{\mathrm{MS}}, \delta_u^{\mathrm{MS}}, \delta_\xi^{\mathrm{MS}}, \delta_e^{\mathrm{MS}}, \delta_g^{\mathrm{MS}}, Z_\psi^{\mathrm{MS}}, Z_\phi^{\mathrm{MS}}, Z_A^{\mathrm{MS}}, Z_m^{\mathrm{MS}}, Z_u^{\mathrm{MS}}, Z_\xi^{\mathrm{MS}}, Z_e^{\mathrm{MS}}, Z_g^{\mathrm{MS}},$$
$$\delta Z_\psi^{\mathrm{MS}}, \delta Z_\phi^{\mathrm{MS}}, \delta Z_A^{\mathrm{MS}}, \delta Z_m^{\mathrm{MS}}, \delta Z_u^{\mathrm{MS}}, \delta Z_\xi^{\mathrm{MS}}, \delta Z_e^{\mathrm{MS}}, \delta Z_g^{\mathrm{MS}}, \overline{\delta_\psi^{\mathrm{MS}}}, \overline{\delta_\phi^{\mathrm{MS}}}, \overline{\delta_A^{\mathrm{MS}}}, \overline{\delta_m^{\mathrm{MS}}}, \overline{\delta_u^{\mathrm{MS}}}, \overline{\delta_\xi^{\mathrm{MS}}},$$
$$\overline{\delta_e^{\mathrm{MS}}}, \overline{\delta_g^{\mathrm{MS}}}, \overline{\delta Z_\psi^{\mathrm{MS}}}, \overline{Z_\phi^{\mathrm{MS}}}, \overline{Z_A^{\mathrm{MS}}}, \overline{Z_m^{\mathrm{MS}}}, \overline{Z_u^{\mathrm{MS}}}, \overline{Z_\xi^{\mathrm{MS}}}, \overline{Z_e^{\mathrm{MS}}}, \overline{Z_g^{\mathrm{MS}}}, \overline{Z_\psi^{\mathrm{MS}}}, \overline{\delta Z_\phi^{\mathrm{MS}}}, \overline{\delta Z_A^{\mathrm{MS}}},$$
$$\overline{\delta Z_m^{\mathrm{MS}}}, \overline{\delta Z_u^{\mathrm{MS}}}, \overline{\delta Z_\xi^{\mathrm{MS}}}, \overline{\delta Z_e^{\mathrm{MS}}}, \overline{\delta Z_g^{\mathrm{MS}}}, \delta_\psi^{\mathrm{OS}}, \delta_\phi^{\mathrm{OS}}, \delta_A^{\mathrm{OS}}, \delta_m^{\mathrm{OS}}, \delta_u^{\mathrm{OS}}, \delta_\xi^{\mathrm{OS}}, \delta_e^{\mathrm{OS}}, \delta_g^{\mathrm{OS}}, Z_\psi^{\mathrm{OS}}, Z_\phi^{\mathrm{OS}}, Z_A^{\mathrm{OS}},$$
$$Z_m^{\mathrm{OS}}, Z_u^{\mathrm{OS}}, Z_\xi^{\mathrm{OS}}, Z_e^{\mathrm{OS}}, Z_g^{\mathrm{OS}}, \delta Z_\psi^{\mathrm{OS}}, \delta Z_\phi^{\mathrm{OS}}, \delta Z_A^{\mathrm{OS}}, \delta Z_m^{\mathrm{OS}}, \delta Z_u^{\mathrm{OS}}, \delta Z_\xi^{\mathrm{OS}}, \delta Z_e^{\mathrm{OS}}, \delta Z_g^{\mathrm{OS}}, V_{\mathrm{ud}}, V_{\mathrm{ud}}^*, V_{\mathrm{us}},$$
$$V_{\mathrm{us}}^*, V_{\mathrm{ub}}, V_{\mathrm{ub}}^*, V_{\mathrm{cd}}, V_{\mathrm{cd}}^*, V_{\mathrm{cs}}, V_{\mathrm{cs}}^*, V_{\mathrm{cb}}, V_{\mathrm{cb}}^*, V_{\mathrm{td}}, V_{\mathrm{td}}^*, V_{\mathrm{ts}}, V_{\mathrm{ts}}^*, V_{\mathrm{tb}}, V_{\mathrm{tb}}^*, s_{12}, s_{13}, s_{23}, c_{12}, c_{13}, c_{23}\Big\}$$

11.39 SMVertex

SMVertex is a library of SM vertices. Currently it implements only few vertices and is not really useful.

11.39.1 See also

Overview, SMP.

11.39.2 Examples

This is the γWW vertex (all momenta ingoing)

```
SMVertex["AWW", p, \[Mu], q, \[Nu], k, \[Rho]]
```

$$-i\,\mathrm{e}\left(\bar{g}^{\mu\rho}\left(\overline{p}-\overline{k}\right)^{\nu}+\bar{g}^{\nu\rho}\left(\overline{k}-\overline{q}\right)^{\mu}+\bar{g}^{\mu\nu}\left(\overline{q}-\overline{p}\right)^{\rho}\right)$$

This is the HHH-coupling

```
SMVertex["HHH"]
```

$$-\frac{3i\,\mathrm{e}m_H^2}{2m_W\left(\sin(\theta_W)\right)}$$

This is the He-coupling

```
SMVertex["eeH"]
```

$$-\frac{i\, e m_e}{2 m_W\, (\sin(\theta_W))}$$

11.40 ToStandardMatrixElement

`ToStandardMatrixElement[exp]` wraps Dirac structures, color structures and polarization vectors with the head **`StandardMatrixElement`**.

The idea of having standard matrix elements stems from A. Denner's "Techniques for the calculation of electroweak radiative corrections at the one-loop level and results for W-physics at LEP200", cf. arXiv:0709.1075.

11.40.1 See also

Overview, DiracSubstitute5, DiracGamma, ToDiracGamma67, Spinor.

11.40.2 Examples

```
Spinor[Momentum[k2, D], 0, 1] . GAD[\[Mu]] . Spinor[-Momentum[k1, D], 0,
1]*
   Spinor[-Momentum[ps, D], SMP["m_s"], 1] . GAD[\[Mu]] .
Spinor[Momentum[pd, D], SMP["m_d"], 1]

ToStandardMatrixElement[%]
```

$$(\varphi(\mathrm{k2})).\gamma^{\mu}.(\varphi(-\mathrm{k1}))\,(\varphi(-\mathrm{ps}, m_s))\,.\gamma^{\mu}.\,(\varphi(\mathrm{pd}, m_d))$$

$$\left\|(\varphi(\mathrm{k2})).\gamma^{\mu}.\bar{\gamma}^{6}.(\varphi(-\mathrm{k1}))\,(\varphi(-\mathrm{ps}, m_s))\,.\gamma^{\mu}.\bar{\gamma}^{6}.\,(\varphi(\mathrm{pd}, m_d))\right\| + \left\|(\varphi(\mathrm{k2})).\gamma^{\mu}.\bar{\gamma}^{7}.(\varphi(-\mathrm{k1}))\,(\varphi(-\mathrm{ps}, m_s))\,.\gamma^{\mu}.\bar{\gamma}^{6}.\,(\varphi(\mathrm{pd}, m_d))\right\| + \left\|(\varphi(\mathrm{k2})).\gamma^{\mu}.\bar{\gamma}^{6}.(\varphi(-\mathrm{k1}))\,(\varphi(-\mathrm{ps}, m_s))\,.\gamma^{\mu}.\bar{\gamma}^{7}.\,(\varphi(\mathrm{pd}, m_d))\right\| + \left\|(\varphi(\mathrm{k2})).\gamma^{\mu}.\bar{\gamma}^{7}.(\varphi(-\mathrm{k1}))\,(\varphi(-\mathrm{ps}, m_s))\,.\gamma^{\mu}.\bar{\gamma}^{7}.\,(\varphi(\mathrm{pd}, m_d))\right\|$$

11.41 QCDFeynmanRuleConvention

`QCDFeynmanRuleConvention` fixes the sign convention in the QCD Feynman rules for the ghost propagator and the ghost-gluon vertex.This is done by setting the value of **`QCDFeynmanRuleConvention[GhostPropagator]`** and **`QCDFeynmanRuleConvention[GluonGhostVertex]`**.

The default values are **`1`** for both, which corresponds to the convention used in most books. Setting them to **`-1`** enforces the convention that can be found e.g. in the book "Applications of Perturbative QCD" by R. Field.

11.41.1 See also

Overview, GluonGhostVertex, GhostPropagator.

11.41.2 Examples

Enforce the convention as in "Applications of Perturbative QCD" by R. Field.

```
QCDFeynmanRuleConvention[GhostPropagator] = -1;

QCDFeynmanRuleConvention[GluonGhostVertex] = -1;
```

```
GHP[p, a, b] // Explicit
```

$$-\frac{i\delta^{ab}}{p^2}$$

```
GGV[{p, \[Mu], a}, {q, \[Nu], b}, {k, \[Rho], c}] // Explicit
```

$$g_s k^{\mu} f^{abc}$$

Back to the standard convention.

```
QCDFeynmanRuleConvention[GhostPropagator] = 1

QCDFeynmanRuleConvention[GluonGhostVertex] = 1
```

$$1$$

$$1$$

```
GHP[p, a, b] // Explicit
```

$$\frac{i\delta^{ab}}{p^2}$$

```
GGV[{p, \[Mu], a}, {q, \[Nu], b}, {k, \[Rho], c}] // Explicit
```

$$g_s\,(-k^{\mu})\,f^{abc}$$

12 Tables

12.1 Amplitude

Amplitude is a database of Feynman amplitudes. **Amplitude["name"]** returns the amplitude corresponding to the string **"name"**. A list of all defined names is obtained with **Amplitude[]**. New amplitudes can be added to the file **"Amplitude.m"**. It is strongly recommended to use names that reflect the process.

The option **Gauge -> 1** means 't Hooft Feynman gauge;

Polarization -> 0 gives unpolarized OPE-type amplitudes, **Polarization -> 1** the polarized ones.

12.1.1 See also

Overview, FeynAmp.

12.1.2 Examples

```
Amplitude[] // Length
```

98

This is the amplitude of a gluon self-energy diagram:

```
Amplitude["se1g1"]

Explicit[%]
```

$$\text{SUNDeltaContract}\Big(f^{\text{FCGV(a)FCGV(c)FCGV(e)}} f^{\text{FCGV(b)FCGV(d)FCGV(f)}} \Pi^{\text{FCGV}(\beta)\text{FCGV}(\sigma)}_{\text{FCGV(e)FCGV(f)}}(\text{FCGV(q)}) V^{\text{FCGV}(\mu)\text{FCGV}(\alpha)\text{FCGV}(\beta)}(\text{FCGV(p)}, -\text{FCGV(p)} - \text{FCGV(q)}, \text{FCGV(q)}) V^{\text{FCGV}(\nu)\text{FCGV}(\rho)\text{FCGV}(\sigma)}(-\text{FCGV(p)}, \text{FCGV(p)} - \text{FCGV(q)}, \text{FCGV(q)}) \Pi^{\text{FCGV}(\alpha)\text{FCGV}(\rho)}_{\text{FCGV(c)FCGV(d)}}(\text{FCGV(p)} - \text{FCGV(q)})\Big)$$

$$-\frac{1}{\text{FCGV(q)}^2(\text{FCGV(p)}-\text{FCGV(q)})^2}g_s^2 g^{\text{FCGV}(\alpha)\text{FCGV}(\rho)} g^{\text{FCGV}(\beta)\text{FCGV}(\sigma)} f^{\text{FCGV(a)FCGV(d)FCGV(f)}} f^{\text{FCGV(b)FCGV(d)FCGV(f)}} \Big($$
$$-\text{FCGV(q)}^{\text{FCGV}(\alpha)}\Big) + g^{\text{FCGV}(\alpha)\text{FCGV}(\mu)}\left(2\,\text{FCGV(p)}^{\text{FCGV}(\beta)} - \text{FCGV(q)}^{\text{FCGV}(\beta)}\right)$$
$$+ g^{\text{FCGV}(\alpha)\text{FCGV}(\beta)}\left(2\,\text{FCGV(q)}^{\text{FCGV}(\mu)} - \text{FCGV(p)}^{\text{FCGV}(\mu)}\right)\Big)\left(g^{\text{FCGV}(\rho)\text{FCGV}(\sigma)}\left(\text{FCGV(p)}^{\text{FCGV}(\nu)}\right.\right.$$
$$-\left.2\,\text{FCGV(q)}^{\text{FCGV}(\nu)}\right) + g^{\text{FCGV}(\nu)\text{FCGV}(\sigma)}\left(\text{FCGV(p)}^{\text{FCGV}(\rho)} + \text{FCGV(q)}^{\text{FCGV}(\rho)}\right)$$
$$+\left.g^{\text{FCGV}(\nu)\text{FCGV}(\rho)}\left(\text{FCGV(q)}^{\text{FCGV}(\sigma)} - 2\,\text{FCGV(p)}^{\text{FCGV}(\sigma)}\right)\right)$$

12.2 AnomalousDimension

`AnomalousDimension[name]` is a database of anomalous dimensions of twist 2 operators.

`AnomalousDimension["gnsqg0"]` yields the non-singlet one-loop contribution to the anomalous dimension $\gamma_{S,qg}^{(0),m}$ in the MS-bar scheme etc.

12.2.1 See also

Overview, SplittingFunction, SumS, SumT.

12.2.2 Examples

Polarized case:

```
SetOptions[AnomalousDimension, Polarization -> 1]
```

$$\{\text{Polarization} \to 1, \text{Simplify} \to \text{FullSimplify}\}$$

$\gamma_{NS,qq}^{(0)}$ polarized:

```
AnomalousDimension[gnsqq0]
```

$$C_F\left(8S_1(m-1) + \frac{4}{m} + \frac{4}{m+1} - 6\right)$$

$\gamma_{S,qg}^{(0)}$ polarized:

```
AnomalousDimension[gsqg0]
```

$$\left(\frac{8}{m}-\frac{16}{m+1}\right)T_f$$

$\gamma_{S,gq}^{(0)}$polarized:

```
AnomalousDimension[gsgq0]
```

$$\left(\frac{4}{m+1}-\frac{8}{m}\right)C_F$$

$\gamma_{S,gg}^{(0)}$ polarized:

```
AnomalousDimension[gsgg0]
```

$$C_A\left(8S_1(m-1)-\frac{8}{m}+\frac{16}{m+1}-\frac{22}{3}\right)+\frac{8T_f}{3}$$

$\gamma_{PS,qq}^{(0)}$ polarized:

```
AnomalousDimension[gpsqq1]
```

$$16\left(\frac{2}{m^3}-\frac{1}{m^2}+\frac{1}{m+1}+\frac{3}{(m+1)^2}+\frac{2}{(m+1)^3}-\frac{1}{m}\right)C_FT_f$$

$\gamma_{NS,qq}^{(1)}$ polarized:

```
AnomalousDimension[gnsqq1]
```

$$\begin{aligned}
&-C_AC_F\left(-\frac{16\tilde{S}_2(m-1)}{m}-\frac{16\tilde{S}_2(m-1)}{m+1}+16\tilde{S}_3(m-1)-32\tilde{S}_{12}(m-1)+\frac{44}{3m^2}-\frac{536}{9}S_1(m-1)\right.\\
&\left.+\frac{88}{3}S_2(m-1)-16S_3(m-1)+\frac{212}{9m}-\frac{748}{9(m+1)}-\frac{4}{3(m+1)^2}-\frac{16}{(m+1)^3}+\frac{17}{3}\right)\\
&-\left(C_F^2\left(\frac{32\tilde{S}_2(m-1)}{m}+\frac{32\tilde{S}_2(m-1)}{m+1}-32\tilde{S}_3(m-1)+64\tilde{S}_{12}(m-1)+\frac{8}{m^3}\right.\right.\\
&+\frac{16S_1(m-1)}{m^2}+\frac{16S_1(m-1)}{(m+1)^2}+\frac{16S_2(m-1)}{m}+\frac{16S_2(m-1)}{m+1}-24S_2(m-1)\\
&\left.\left.+32S_{12}(m-1)+32S_{21}(m-1)-\frac{40}{m}+\frac{40}{m+1}+\frac{16}{(m+1)^2}+\frac{40}{(m+1)^3}+3\right)\right)\\
&-C_FN_f\left(-\frac{8}{3m^2}+\frac{80}{9}S_1(m-1)-\frac{16}{3}S_2(m-1)-\frac{8}{9m}+\frac{88}{9(m+1)}-\frac{8}{3(m+1)^2}-\frac{2}{3}\right)
\end{aligned}$$

$\gamma_{S,qg}^{(1)}$ polarized:

```
AnomalousDimension[gsqg1]
```

$$16C_AT_f\left(-\frac{2\tilde{S}_2(m-1)}{m}+\frac{4\tilde{S}_2(m-1)}{m+1}+\frac{2}{m^3}-\frac{2S_1(m-1)}{m^2}-\frac{3}{m^2}-\frac{S_1^2(m-1)}{m}+\frac{2S_1^2(m-1)}{m+1}\right.$$
$$\left.+\frac{4S_1(m-1)}{(m+1)^2}-\frac{S_2(m-1)}{m}+\frac{2S_2(m-1)}{m+1}-\frac{4}{m}+\frac{3}{m+1}+\frac{8}{(m+1)^2}+\frac{12}{(m+1)^3}\right)+8C_FT_f\left(-\frac{2}{m^3}\right.$$
$$\left.-\frac{1}{m^2}+\frac{2S_1^2(m-1)}{m}-\frac{4S_1^2(m-1)}{m+1}-\frac{2S_2(m-1)}{m}+\frac{4S_2(m-1)}{m+1}+\frac{14}{m}-\frac{19}{m+1}-\frac{8}{(m+1)^2}+\frac{4}{(m+1)^3}\right)$$

$\gamma_{S,gq}^{(1)}$ polarized:

```
AnomalousDimension[gsgq1]
```

$$8C_AC_F\left(\frac{4\tilde{S}_2(m-1)}{m}-\frac{2\tilde{S}_2(m-1)}{m+1}-\frac{4}{m^3}+\frac{28}{3m^2}-\frac{2S_1^2(m-1)}{m}+\frac{S_1^2(m-1)}{m+1}+\frac{16S_1(m-1)}{3m}\right.$$
$$\left.-\frac{5S_1(m-1)}{3(m+1)}+\frac{2S_2(m-1)}{m}-\frac{S_2(m-1)}{m+1}-\frac{56}{9m}-\frac{20}{9(m+1)}-\frac{38}{3(m+1)^2}-\frac{6}{(m+1)^3}\right)$$
$$+32C_FT_f\left(-\frac{2}{3m^2}-\frac{2S_1(m-1)}{3m}+\frac{S_1(m-1)}{3(m+1)}+\frac{7}{9m}-\frac{2}{9(m+1)}+\frac{1}{3(m+1)^2}\right)$$
$$+4C_F^2\left(\frac{4}{m^3}+\frac{8S_1(m-1)}{m^2}-\frac{12}{m^2}+\frac{4S_1^2(m-1)}{m}-\frac{2S_1^2(m-1)}{m+1}-\frac{8S_1(m-1)}{m}+\frac{2S_1(m-1)}{m+1}\right.$$
$$\left.-\frac{4S_1(m-1)}{(m+1)^2}+\frac{4S_2(m-1)}{m}-\frac{2S_2(m-1)}{m+1}+\frac{15}{m}-\frac{6}{m+1}+\frac{3}{(m+1)^2}-\frac{2}{(m+1)^3}\right)$$

$\gamma_{S,gg}^{(1)}$ polarized:

```
v1 = AnomalousDimension[gsgg1]
```

$$4C_A^2\left(\frac{8\tilde{S}_2(m-1)}{m}-\frac{16\tilde{S}_2(m-1)}{m+1}+4\tilde{S}_3(m-1)-8\tilde{S}_{12}(m-1)-\frac{8}{m^3}+\frac{8S_1(m-1)}{m^2}+\frac{58}{3m^2}-\frac{16S_1(m-1)}{(m+1)^2}\right.$$
$$+\frac{134}{9}S_1(m-1)+\frac{8S_2(m-1)}{m}-\frac{16S_2(m-1)}{m+1}+4S_3(m-1)-8S_{12}(m-1)-8S_{21}(m-1)-\frac{107}{9m}+\frac{241}{9(m+1)}$$
$$\left.-\frac{86}{3(m+1)^2}-\frac{48}{(m+1)^3}-\frac{16}{3}\right)+32C_AT_f\left(-\frac{1}{3m^2}-\frac{5}{9}S_1(m-1)+\frac{14}{9m}-\frac{19}{9(m+1)}-\frac{1}{3(m+1)^2}+\frac{1}{3}\right)$$
$$+8\left(\frac{4}{m^3}-\frac{10}{m^2}-\frac{10}{m+1}+\frac{2}{(m+1)^2}+\frac{4}{(m+1)^3}+\frac{10}{m}+1\right)C_FT_f$$

$\gamma_{S,gg}^{(1)}$ polarized (different representation):

```
v2 = AnomalousDimension[GSGG1];
```

Check that all odd moments give the same for the two representations of $\gamma^{(1)}_{S,gg}$:

```
Table[v1 - v2 /. OPEm -> ij, {ij, 1, 17, 2}] // Simplify
```

$$\{0, 0, 0, 0, 0, 0, 0, 0, 0\}$$

12.3 CheckDB

CheckDB[exp, fil] saves (with **Put**) or retrieves (with **Get**) **exp** from a file **fil**. It checks if the setting of the option **Directory** is a valid directory name and if **fil** is a valid file name and does exist. If it does, **Get[fil]** is executed. If **fil** does not exist, **exp** gets evaluated and saved to **fil**.

Saving and evaluating can be further controlled with the options **ForceSave** and **NoSave**. If the option **Check** is set to **False** the return value is what is evaluated (see above). If **Check** is set to **True** the return value is **True** or **False** depending on whether the evaluation of **exp** agrees with what is loaded from **fil** or **fil** does not exist.

Default value of Check : **False**. If **fil** ends with **".Gen"** or **".Mod"**, the setting of **Directory** is ignored and **fil** is saved in the **"CouplingVectors"** subdirectory of **"Phi"**. If **fil** ends with **".Fac"**, the setting of Directory is ignored and **fil** is saved in the **"Factors"** subdirectory of **"Phi"**.

If **fil** is a file name with full path, the setting of **Directory** is also ignored.

12.3.1 See also

Overview

12.3.2 Examples

The first time the Table function is evaluated and the result saved into the test.s file.

```
CheckDB[Table[WriteString["stdout", "test "]; i, {i, 2}], "test.s"]
```

test test

$$\{1, 2\}$$

Executing the same a second time will just load the result from test.s and not evaluate the Table function.

```
CheckDB[Table[WriteString["stdout", "test "]; i, {i, 2}], "test.s"]
```

$$\{1, 2\}$$

This shows the actual saved value of test.s.

```
Import[ToFileName[Directory /. Options[CheckDB], "test.s"], "Text"]
```

$$\{1, 2\}$$

```
DeleteFile[ToFileName[Directory /. Options[CheckDB], "test.s"]]
```

12.4 Convolute

`Convolute[f, g, x]` convolutes $f(x)$ and $g(x)$, i.e., $\int_0^1 dx_1 \int_0^1 dx_2 \delta(x - x_1 x_2) f(x_1) g(x_2)$.

`Convolute[f, g]` is equivalent to **`Convolute[f, g, x]`**.

`Convolute[exp, {x1, x2}]` assumes that **`exp`** is polynomial in **`x1`** and **`x2`**. Convolute uses table-look-up and does not do any integral calculations, only linear algebra.

12.4.1 See also

Overview, PlusDistribution, ConvoluteTable.

12.4.2 Examples

```
Convolute[1, 1] /. FCGV[z_] :> ToExpression[z]
```

$$-\log(x)$$

```
Convolute[x, x] /. FCGV[z_] :> ToExpression[z]
```

$$-x^2 \log(x)$$

```
Convolute[1, x] /. FCGV[z_] :> ToExpression[z]
```

$$-x \log(x)$$

```
Convolute[1, 1/(1 - x)] /. FCGV[z_] :> ToExpression[z]
```

$$\frac{\log(x)}{x-1}$$

```
Convolute[1, PlusDistribution[1/(1 - x)]] /. FCGV[z_] :> ToExpression[z]
```

$$\frac{\log(x)}{x-1}$$

```
Convolute[1/(1 - x), x] /. FCGV[z_] :> ToExpression[z]
```

$$\frac{x\log(x)}{x-1}$$

```
Convolute[1/(1 - x), 1/(1 - x)] /. FCGV[z_] :> ToExpression[z]
```

$$-\frac{\log(x)}{(x-1)^2}$$

```
Convolute[1, Log[1 - x]] /. FCGV[z_] :> ToExpression[z]
```

$$-\log(1-x)\log(x)$$

```
Convolute[1, x Log[1 - x]] /. FCGV[z_] :> ToExpression[z]
```

$$-x\log(1-x)\log(x)$$

```
Convolute[1/(1 - x), Log[1 - x]] /. FCGV[z_] :> ToExpression[z]
```

$$\frac{\log(1-x)\log(x)}{x-1}$$

```
Convolute[1/(1 - x), x Log[1 - x]] /. FCGV[z_] :> ToExpression[z]
```

$$\frac{x \log(1-x) \log(x)}{x-1}$$

```
Convolute[Log[1 - x]/(1 - x), x] /. FCGV[z_] :> ToExpression[z]
```

$$\frac{x \log(1-x) \log(x)}{x-1}$$

```
Convolute[1, x Log[x]] /. FCGV[z_] :> ToExpression[z]
```

$$-x \log^2(x)$$

```
Convolute[Log[1 - x], x] /. FCGV[z_] :> ToExpression[z]
```

$$-x \log(1-x) \log(x)$$

```
Convolute[1/(1 - x), Log[x]/(1 - x)] /. FCGV[z_] :> ToExpression[z]
```

$$-\frac{\log^2(x)}{(x-1)^2}$$

```
Convolute[1, Log[x]] /. FCGV[z_] :> ToExpression[z]
```

$$-\log^2(x)$$

```
Convolute[x, x Log[x]] /. FCGV[z_] :> ToExpression[z]
```

$$-x^2 \log^2(x)$$

```
Convolute[1/(1 - x), Log[x]] /. FCGV[z_] :> ToExpression[z]
```

$$\frac{\log^2(x)}{x-1}$$

```
Convolute[1, Log[x]/(1 - x)] /. FCGV[z_] :> ToExpression[z]
```

$$\frac{\log^2(x)}{x-1}$$

```
Convolute[1/(1 - x), x Log[x]] /. FCGV[z_] :> ToExpression[z]
```

$$\frac{x \log^2(x)}{x-1}$$

```
Convolute[Log[x]/(1 - x), x] /. FCGV[z_] :> ToExpression[z]
```

$$\frac{x \log^2(x)}{x-1}$$

```
Convolute[1, x Log[x]] /. FCGV[z_] :> ToExpression[z]
```

$$-x \log^2(x)$$

```
Convolute[Log[x], x] /. FCGV[z_] :> ToExpression[z]
```

$$-x \log^2(x)$$

```
Convolute[1/(1 - x), Log[1 - x]/(1 - x)] /. FCGV[z_] :> ToExpression[z]
```

$$-\frac{\log(1-x)\log(x)}{(x-1)^2}$$

```
Convolute[Log[1 - x]/(1 - x), Log[1 - x]] /. FCGV[z_] :> ToExpression[z]
```

$$\frac{\log^2(1-x)\log(x)}{x-1}$$

12.5 ConvoluteTable

`ConvoluteTable[f, g, x]` yields the convolution of **`f`** and **`g`**. **`ConvoluteTable`** is called by **`Convolute`**.

12.5.1 See also

Overview, PlusDistribution, Convolute.

12.5.2 Examples

```
ConvoluteTable[1, 1, x]
```

$$-\log(x)$$

```
ConvoluteTable[x, x]
```

$$\text{False}[x, x]$$

```
ConvoluteTable[1, x, x]
```

$$1 - x$$

12.6 CounterTerm

`CounterTerm[name]` is a database of counter terms. **`CounterTerm`** is also an option for the Feynman rule functions **`QuarkGluonVertex`**, **`GluonPropagator`**, **`QuarkPropagator`**.

12.6.1 See also

Overview, CounterT, QuarkGluonVertex, GluonPropagator, QuarkPropagator.

12.6.2 Examples

```
CounterTerm[Zm]
```

$$\frac{C_F g_s^4 \left(\frac{4\left(\frac{11C_A}{2} + \frac{9C_F}{2} - 2N_f T_f\right)}{\varepsilon^2} + \frac{2\left(\frac{97C_A}{12} + \frac{3C_F}{4} - \frac{5N_f T_f}{3}\right)}{\varepsilon} \right)}{256\pi^4} + \frac{3C_F g_s^2}{8\pi^2 \varepsilon} + 1$$

12.7 Gamma1

Gamma1[al, ga, be, de] is a special product of **Gamma** functions expanded up to order **Epsilon^2**.

12.7.1 See also

Overview, Gamma2, Gamma3.

12.7.2 Examples

12.8 Gamma2

Gamma2[x, y] is a special product of **Gamma** functions expanded up to order **Epsilon^3** when positive integer arguments are given.

12.8.1 See also

Overview, Gamma1, Gamma3.

12.8.2 Examples

12.9 Gamma3

Gamma3[al, be, ga, ep] is a special product of Gamma functions expanded up to order **Epsilon^n** when positive integer arguments are given (the order **n** is determined by the option **EpsilonOrder**).

12.9.1 See also

Overview, Gamma1, Gamma2.

12.9.2 Examples

12.10 GammaEpsilon

GammaEpsilon[exp] gives a series expansion of **Gamma[exp]** in **Epsilon** up to order **6** (where **EulerGamma** is neglected).

12.10.1 See also

Overview, GammaExpand, Series2.

12.10.2 Examples

If the argument is of the form **(1+a Epsilon)** the result is not calculated but tabulated.

```
GammaEpsilon[1 + a Epsilon]
```

$$\varepsilon^5\left(-\frac{a^5\zeta(5)}{5}-\frac{1}{36}\pi^2a^5\zeta(3)\right)+\frac{1}{160}\pi^4a^4\varepsilon^4-\frac{1}{3}a^3\varepsilon^3\zeta(3)+\frac{1}{12}\pi^2a^2\varepsilon^2+\text{C\$16471}\varepsilon^6+1$$

```
GammaEpsilon[1 - Epsilon/2]
```

$$\text{C\$16508}\varepsilon^6+\frac{\pi^4\varepsilon^4}{2560}+\frac{\pi^2\varepsilon^2}{48}+\varepsilon^5\left(\frac{\pi^2\zeta(3)}{1152}+\frac{\zeta(5)}{160}\right)+\frac{\varepsilon^3\zeta(3)}{24}+1$$

For other arguments the expansion is calculated.

```
GammaEpsilon[Epsilon]
```

$$\text{C\$17651}\varepsilon^6+\frac{\pi^4\varepsilon^3}{160}+\frac{\pi^2\varepsilon}{12}+\frac{1}{\varepsilon}+\frac{1}{720}\varepsilon^5\left(\frac{61\pi^6}{168}+10\psi^{(2)}(1)^2\right)+\frac{\varepsilon^2\psi^{(2)}(1)}{6}$$
$$+\frac{\varepsilon^6\left(-84\pi^2\zeta(5)+\frac{21\pi^4\psi^{(2)}(1)}{4}+\psi^{(6)}(1)\right)}{5040}+\frac{1}{120}\varepsilon^4\left(\frac{5\pi^2\psi^{(2)}(1)}{3}-24\zeta(5)\right)$$

```
GammaEpsilon[x]
```

$$\Gamma(x)$$

12.11 Integrate2

Integrate2 is like **Integrate**, but **Integrate2[a_Plus, b__] := Map[Integrate2[#, b]&, a]** (more linear algebra and partial fraction decomposition is done)

Integrate2[f[x] DeltaFunction[x], x] -> f[0]

Integrate2[f[x] DeltaFunction[x0-x], x] -> f[x0]

Integrate2[f[x] DeltaFunction[a + b x], x] -> Integrate[f[x] (1/Abs[b]) DeltaFunction[a/b + x], x], where **Abs[b] -> b**, if **b** is a symbol, and if **b = -c**, then **Abs[-c] -> c**, i.e., the variable contained in **b** is supposed to be positive.

π^2 is replaced by **6 Zeta2**.

Integrate2[1/(1-y), {y, x, 1}] is interpreted as distribution, i.e. as **Integrate2[-1/(1-y)], {y, 0, x}] -> Log[1-y]**.

Integrate2[1/(1-x), {x, 0, 1}] -> 0

Since **Integrate2** does do a reordering and partial fraction decomposition before calling the integral table of **Integrate3**, it will in general be slower compared to Integrate3 for sums of integrals. I.e., if the integrand has already an expanded form and if partial fraction decomposition is not necessary it is more effective to use **Integrate3**.

12.11.1 See also

Overview, DeltaFunction, Integrate3, Integrate5, SumS, SumT.

12.11.2 Examples

```
Integrate2[Log[1 + x] Log[x]/(1 - x), {x, 0, 1}] // Timing
```

$$\left\{0.057955, \zeta(3) - \frac{3}{2}\zeta(2)\log(2)\right\}$$

Since **Integrate2** uses table-look-up methods it is much faster than Mathematica's Integrate.

```
Integrate2[PolyLog[2, x^2], {x, 0, 1}]
```

$$\zeta(2) - 4 + 4\log(2)$$

```
Integrate2[PolyLog[3, -x], {x, 0, 1}]
```

$$\frac{\zeta(2)}{2} - \frac{3\zeta(3)}{4} + 1 - 2\log(2)$$

```
Integrate2[PolyLog[3, 1/(1 + x)], {x, 0, 1}]
```

$$\zeta(2)(-\log(2)) + \frac{3\zeta(3)}{4} + \frac{\log^3(2)}{3} - \log^2(2) + 2\log(2)$$

```
Integrate2[DeltaFunction[1 - x] f[x], {x, 0, 1}]
```

$$f(1)$$

Integrate2 does integration in a Hadamard sense, i.e., $\int_0^1 f(x)\,dx$ means actually expanding the result of $\int_\delta^{1-\delta} f(x)\,dx$ up to $\mathcal{O}(\delta)$ and neglecting all δ-dependent terms. E.g. $\int_\delta^{1-\delta} \frac{1}{1-x}\,dx = -\log(1-x)\Big|_\delta^{1-\delta} = -\log(\delta) + log(1) \Rightarrow 0$

```
Integrate2[1/(1 - x), {x, 0, 1}]
```

$$0$$

In the physics literature sometimes the "+" notation is used. In FeynCalc the $(frac11-x)_+$ is represented by **PlusDistribution}[1/(1-x)]** or just **1/(1-x)**

```
Integrate2[PlusDistribution[1/(1 - x)], {x, 0, 1}]
```

$$0$$

```
Integrate2[PolyLog[2, 1 - x]/(1 - x)^2, {x, 0, 1}]
```

$$2 - \zeta(2)$$

```
Integrate2[(Log[x] Log[1 + x])/(1 + x), {x, 0, 1}]
```

$$-\frac{\zeta(3)}{8}$$

```
Integrate2[Log[x]^2/(1 - x), {x, 0, 1}]
```

$$2\zeta(3)$$

```
Integrate2[PolyLog[2, -x]/(1 + x), {x, 0, 1}]
```

$$\frac{\zeta(3)}{4} - \frac{1}{2}\zeta(2)\log(2)$$

```
Integrate2[Log[x] PolyLog[2, x], {x, 0, 1}]
```

$$3 - 2\zeta(2)$$

```
Integrate2[x PolyLog[3, x], {x, 0, 1}]
```

$$-\frac{\zeta(2)}{4} + \frac{\zeta(3)}{2} + \frac{3}{16}$$

```
Integrate2[(Log[x]^2 Log[1 - x])/(1 + x), {x, 0, 1}]
```

$$\zeta(4) + \zeta(2)\log^2(2) - 4\,\mathrm{Li}_4\left(\frac{1}{2}\right) - \frac{\log^4(2)}{6}$$

```
Integrate2[PolyLog[2, ((x (1 - z) + z) (1 - x + x z))/z]/(1 - x + x z), {x,
0, 1}]
```

$$\begin{aligned}
&\frac{2i\pi\,\mathrm{Li}_2(-z)}{1-z} - \frac{4\,\mathrm{Li}_3\left(\frac{1-z}{2}\right)}{1-z} + \frac{4\,\mathrm{Li}_3(1-z)}{1-z} + \frac{2\,\mathrm{Li}_3(-z)}{1-z} + \frac{4\,\mathrm{Li}_3\left(\frac{1}{z+1}\right)}{1-z} - \frac{4\,\mathrm{Li}_3\left(\frac{1-z}{z+1}\right)}{1-z} \\
&- \frac{4\,\mathrm{Li}_3\left(\frac{z+1}{2}\right)}{1-z} - \frac{2\,\mathrm{Li}_2(1-z)\log(z)}{1-z} - \frac{2\,\mathrm{Li}_2(-z)\log(z)}{1-z} + \frac{4\,\mathrm{Li}_2(-z)\log(1-z)}{1-z} \\
&- \frac{2S_{12}(1-z)}{1-z} + \frac{i\pi\zeta(2)}{1-z} - \frac{\zeta(2)\log(z)}{1-z} + \frac{2\zeta(2)\log(1-z)}{1-z} + \frac{6\zeta(2)\log(z+1)}{1-z} \\
&- \frac{4\zeta(2)\log(2)}{1-z} + \frac{2\zeta(3)}{1-z} + \frac{\log^3(z)}{6(1-z)} + \frac{4\log^3(2)}{3(1-z)} - \frac{\log(1-z)\log^2(z)}{1-z} - \frac{\log(z+1)\log^2(z)}{1-z} \\
&- \frac{i\pi\log^2(z)}{2(1-z)} - \frac{2\log(1-z)\log^2(z+1)}{1-z} - \frac{2\log^2(2)\log(1-z)}{1-z} - \frac{2\log^2(2)\log(z+1)}{1-z} \\
&+ \frac{4\log(1-z)\log(z+1)\log(z)}{1-z} + \frac{2i\pi\log(z+1)\log(z)}{1-z} + \frac{4\log(2)\log(1-z)\log(z+1)}{1-z}
\end{aligned}$$

```
Apart[Integrate2[x^(OPEm - 1) PolyLog[3, 1 - x], {x, 0, 1}], OPEm]
```

$$-\frac{\zeta(2)}{m^2} - \frac{\zeta(2)}{m-1} + \frac{\zeta(2) + \zeta(2)\,(-S_1(m-2)) + S_{12}(m) + \zeta(3)}{m}$$

```
Integrate2[x^(OPEm - 1) Log[1 - x] Log[x] Log[1 + x]/(1 + x), {x, 0, 1}] //
Simplify

% /. OPEm -> 2

N[%]
```

$$\frac{1}{24}(-1)^m \Big(48\zeta(4) + 30\zeta(2)\log^2(2) + 6\zeta(2)S_{-1}^2(m-1) + 18\zeta(2)S_2(m-1)$$
$$- 24\zeta(2)S_{1-1}(m-1) - 12S_{-2}(m-1)\left(\zeta(2) - \log(4)S_{-1}(m-1) - \log^2(2)\right)$$
$$- 36\zeta(2)\log(2)S_1(m-1) + 12S_{-1}(m-1)(\zeta(2)\log(8) - 2\zeta(3)) + 39\zeta(3)S_1(m-1)$$
$$+ 24S_{-2-1-1}(m-1) + 24S_{-1-2-1}(m-1) + 24S_{-1-1-2}(m-1) + 24S_{1-21}(m-1)$$
$$+ 24S_{1-12}(m-1) + 24S_{2-11}(m-1) - 12\log^2(2)S_2(m-1) + 24\log(2)S_3(m-1)$$
$$- 24\log(2)S_{-21}(m-1) - 24\log(2)S_{-12}(m-1) - 48\,\mathrm{Li}_4\left(\frac{1}{2}\right) - 63\zeta(3)\log(2) - 2\log^4(2)\Big)$$

$$\frac{1}{24}\Big(48\zeta(2) + 48\zeta(4) + 30\zeta(2)\log^2(2) + 12\left(\zeta(2) - \log^2(2) + \log(4)\right) - 36\zeta(2)\log(2) - 48\,\mathrm{Li}_4\left(\frac{1}{2}\right)$$
$$- 12(\zeta(2)\log(8) - 2\zeta(3)) + 39\zeta(3) - 63\zeta(3)\log(2) - 144 - 2\log^4(2) - 12\log^2(2) + 72\log(2)\Big)$$

0.0505138

```
Integrate2[x^(OPEm - 1) (PolyLog[3, (1 - x)/(1 + x)] - PolyLog[3, -((1 -
x)/(1 + x))]), {x, 0, 1}]
```

$$\frac{3\zeta(2)(-1)^m\log(2)}{2m} - \frac{3\zeta(2)\log(2)}{2m} + \frac{\zeta(2)(-1)^m S_{-1}(m)}{m} - \frac{\zeta(2)S_{-1}(m)}{2m} + \frac{\zeta(2)(-1)^m S_1(m)}{2m}$$
$$- \frac{\zeta(2)S_1(m)}{m} + \frac{(-1)^m S_{-3}(m)}{m} + \frac{(-1)^m S_{-2}(m)S_1(m)}{m} + \frac{S_1(m)S_2(m)}{m} + \frac{S_3(m)}{m}$$
$$- \frac{(-1)^m S_{-21}(m)}{m} - \frac{S_{-1-2}(m)}{m} - \frac{(-1)^m S_{-12}(m)}{m} - \frac{S_{21}(m)}{m} - \frac{7(-1)^m\zeta(3)}{8m} + \frac{21\zeta(3)}{8m}$$

```
DataType[OPEm, PositiveInteger]

Integrate2[x^(OPEm - 1) DeltaFunction[1 - x], {x, 0, 1}]
```

True

1

This is the polarized non-singlet spin splitting function whose first moment vanishes.

```
t = SplittingFunction[PQQNS] /. FCGV[z_] :> ToExpression[z]
```

$$
\begin{aligned}
&-8C_F\left(C_F-\frac{C_A}{2}\right)\left(\frac{\left(x^2+1\right)\left(-2\zeta(2)-4\,\mathrm{Li}_2(-x)+\log^2(x)-4\log(x+1)\log(x)\right)}{x+1}+4(1-x)\right.\\
&\left.+2(x+1)\log(x)\right)+C_AC_F\left(\frac{4\left(x^2+1\right)\log^2(x)}{1-x}+8\zeta(2)(x+1)+\left(\frac{536}{9}-16\zeta(2)\right)\left(\frac{1}{1-x}\right)_+\right.\\
&\left.+\delta(1-x)\left(\frac{88\zeta(2)}{3}-24\zeta(3)+\frac{17}{3}\right)+\frac{4}{9}(53-187x)-\frac{4}{3}\left(5x-\frac{22}{1-x}+5\right)\log(x)\right)\\
&+C_FN_f\left(-\frac{8\left(x^2+1\right)\log(x)}{3(1-x)}+\left(-\frac{16\zeta(2)}{3}-\frac{2}{3}\right)\delta(1-x)+\frac{88x}{9}-\frac{80}{9}\left(\frac{1}{1-x}\right)_+-\frac{8}{9}\right)\\
&+C_F^2\left(-\frac{16\left(x^2+1\right)\log(1-x)\log(x)}{1-x}+\delta(1-x)(-24\zeta(2)+48\zeta(3)+3)\right.\\
&\left.-40(1-x)-4(x+1)\log^2(x)-8\left(2x+\frac{3}{1-x}\right)\log(x)\right)
\end{aligned}
$$

```
t // Expand
```

$$
\begin{aligned}
&8\zeta(2)C_AC_F-\frac{16x^2C_AC_F\,\mathrm{Li}_2(-x)}{x+1}-\frac{16C_AC_F\,\mathrm{Li}_2(-x)}{x+1}-\frac{8\zeta(2)x^2C_AC_F}{x+1}+\frac{4x^2C_AC_F\log^2(x)}{1-x}\\
&+\frac{4x^2C_AC_F\log^2(x)}{x+1}-\frac{16x^2C_AC_F\log(x)\log(x+1)}{x+1}+\frac{88}{3}\zeta(2)C_AC_F\delta(1-x)+\frac{17}{3}C_AC_F\delta(1-x)\\
&+8\zeta(2)xC_AC_F-\frac{8\zeta(2)C_AC_F}{x+1}-16\zeta(2)\left(\frac{1}{1-x}\right)_+C_AC_F-24\zeta(3)C_AC_F\delta(1-x)\\
&-\frac{892}{9}xC_AC_F+\frac{536}{9}\left(\frac{1}{1-x}\right)_+C_AC_F+\frac{4C_AC_F\log^2(x)}{1-x}+\frac{4C_AC_F\log^2(x)}{x+1}\\
&+\frac{4}{3}C_AC_F\log(x)+\frac{4}{3}xC_AC_F\log(x)+\frac{88C_AC_F\log(x)}{3(1-x)}-\frac{16C_AC_F\log(x)\log(x+1)}{x+1}\\
&+\frac{356C_AC_F}{9}-\frac{8x^2C_FN_f\log(x)}{3(1-x)}-\frac{16}{3}\zeta(2)C_FN_f\delta(1-x)-\frac{2}{3}C_FN_f\delta(1-x)\\
&+\frac{88}{9}xC_FN_f-\frac{80}{9}\left(\frac{1}{1-x}\right)_+C_FN_f-\frac{8C_FN_f\log(x)}{3(1-x)}-\frac{8C_FN_f}{9}+\frac{32x^2C_F^2\,\mathrm{Li}_2(-x)}{x+1}\\
&+\frac{32C_F^2\,\mathrm{Li}_2(-x)}{x+1}+\frac{16\zeta(2)x^2C_F^2}{x+1}-\frac{8x^2C_F^2\log^2(x)}{x+1}-\frac{16x^2C_F^2\log(1-x)\log(x)}{1-x}\\
&+\frac{32x^2C_F^2\log(x)\log(x+1)}{x+1}-24\zeta(2)C_F^2\delta(1-x)+3C_F^2\delta(1-x)+\frac{16\zeta(2)C_F^2}{x+1}\\
&+48\zeta(3)C_F^2\delta(1-x)+72xC_F^2-4xC_F^2\log^2(x)-\frac{8C_F^2\log^2(x)}{x+1}-4C_F^2\log^2(x)-32xC_F^2\log(x)\\
&-\frac{16C_F^2\log(1-x)\log(x)}{1-x}-\frac{24C_F^2\log(x)}{1-x}-16C_F^2\log(x)+\frac{32C_F^2\log(x)\log(x+1)}{x+1}-72C_F^2
\end{aligned}
$$

```
Integrate2[t, {x, 0, 1}] // Timing
```

$$\{0.040008, 0\}$$

Expanding **t** with respect to **x** yields a form already suitable for **Integrate3** and therefore the following is faster:

```
Integrate3[Expand[t, x], {x, 0, 1}] // Expand // Timing
```

$$\{0.018181, 0\}$$

```
Clear[t];

Integrate2[DeltaFunction[1 - x] f[x], {x, 0, 1}]
```

$$f(1)$$

```
Integrate2[x^5 Log[1 + x]^2, {x, 0, 1}]

N[%]
```

$$\frac{46\log(2)}{45} - \frac{6959}{10800}$$

$$0.0641986$$

```
NIntegrate[x^5 Log[1 + x]^2, {x, 0, 1}]
```

$$0.0641986$$

```
Integrate2[x^(OPEm - 1) Log[1 + x]^2, {x, 0, 1}]
```

$$-\frac{2(-1)^m S_1^2(m)}{m} + \frac{(-1)^m S_1\left(\frac{m-1}{2}\right) S_1(m)}{m} - \frac{S_1\left(\frac{m-1}{2}\right) S_1(m)}{m} + \frac{(-1)^m S_1\left(\frac{m}{2}\right) S_1(m)}{m}$$
$$+ \frac{S_1\left(\frac{m}{2}\right) S_1(m)}{m} + \frac{(-1)^m S_2\left(\frac{m-1}{2}\right)}{2m} - \frac{S_2\left(\frac{m-1}{2}\right)}{2m} + \frac{(-1)^m S_2\left(\frac{m}{2}\right)}{2m} + \frac{S_2\left(\frac{m}{2}\right)}{2m}$$
$$- \frac{2(-1)^m S_2(m)}{m} - \frac{2(-1)^m S_{-11}(m)}{m} + \frac{4(-1)^m \log(2) S_1(m)}{m} - \frac{(-1)^m \log(2) S_1\left(\frac{m-1}{2}\right)}{m}$$
$$+ \frac{\log(2) S_1\left(\frac{m-1}{2}\right)}{m} - \frac{(-1)^m \log(2) S_1\left(\frac{m}{2}\right)}{m} - \frac{\log(2) S_1\left(\frac{m}{2}\right)}{m} - \frac{(-1)^m \log^2(2)}{m} + \frac{\log^2(2)}{m}$$

12.12 Integrate3

Integrate3 contains the integral table used by **Integrate2**. Integration is performed in a distributional sense. **Integrate3** works more effectively on a sum of expressions if they are expanded or collected with respect to the integration variable. See the examples in **Integrate2**.

12.12.1 See also

Overview, Integrate2.

12.12.2 Examples

```
Integrate3[x^OPEm Log[x], {x, 0, 1}]
```

$$-\frac{1}{(m+1)^2}$$

```
Integrate3[(x^OPEm Log[x] Log[1 - x])/(1 - x), {x, 0, 1}]
```

$$\zeta(2)S_1(m) - S_{12}(m) - S_{21}(m) + \zeta(3)$$

```
Integrate3[a (x^OPEm Log[x] Log[1 - x])/(1 - x) + b (x^OPEm PolyLog[3,
-x])/(1 + x), {x, 0, 1}]
```

$$a\left(\zeta(2)S_1(m) - S_{12}(m) - S_{21}(m) + \zeta(3)\right) + b(-1)^m\left(\frac{\zeta(2)^2}{8} + \frac{1}{2}\zeta(2)S_{-2}(m) - \frac{3}{4}\zeta(3)S_{-1}(m) + S_{3,-1}(m) + \log(2)\left(S_3(m) - S_{-3}(m)\right) - \frac{3}{4}\zeta(3)\log(2)\right)$$

```
Integrate3[DeltaFunctionPrime[1 - x], {x, 0, 1}]
```

$$0$$

```
Integrate3[f[x] DeltaFunctionPrime[1 - x], {x, 0, 1}]
```

$$f'(1)$$

```
Integrate3[1/(1 - x), {x, 0, 1}]
```

$$0$$

12.13 Integrate5

Integrate5 is an alternative implementation along the lines of **Integrate2**.

12.13.1 See also

Overview, Integrate2.

12.13.2 Examples

12.14 InverseMellin

InverseMellin[exp, y] performs the inverse Mellin transform of polynomials in OPE. The inverse transforms are not calculated but a table-lookup is done.

WARNING: do not "trust" the results for the inverse Mellin transform involving SumT's; there is an unresolved inconsistency here (related to $(-1)^m$).

12.14.1 See also

Overview, DeltaFunction, Integrate2, OPEm, SumS, SumT.

12.14.2 Examples

```
InverseMellin[1/OPEm, y]
```

$$y^{m-1}$$

```
InverseMellin[1/(OPEm + 3), y]
```

$$y^{m+2}$$

```
InverseMellin[1, y]
```

$$y^{m-1}\delta(1-y)$$

```
InverseMellin[1/OPEm^4, y]
```

$$-\frac{1}{6}y^{m-1}\log^3(y)$$

```
InverseMellin[1/OPEm + 1, y]
```

$$y^{m-1}\delta(1-y)+y^{m-1}$$

```
InverseMellin[1/i + 1, y, i]
```

$$y^{i-1}\delta(1-y)+y^{i-1}$$

The inverse operation to **InverseMellin** is done by **Integrate2**.

```
Integrate2[InverseMellin[1/OPEm, y], {y, 0, 1}]
```

$$\frac{1}{m}$$

Below is a list of all built-in basic inverse Mellin transforms .

```
list = {1, 1/(OPEm + n), 1/(-OPEm + n), PolyGamma[0, OPEm], SumS[1, -1 +
OPEm],
    SumS[1, -1 + OPEm]/(OPEm - 1), SumS[1, -1 + OPEm]/(1 - OPEm), SumS[1,
-1 + OPEm]/(OPEm + 1),
    SumS[1, -1 + OPEm]/OPEm^2, SumS[1, -1 + OPEm]/OPEm, SumS[1, -1 +
OPEm]^2/OPEm,
    SumS[2, -1 + OPEm], SumS[2, -1 + OPEm]/OPEm, SumS[3, -1 + OPEm],
SumS[1, 1, -1 + OPEm],
    SumS[1, OPEm - 1]^2, SumS[1, 2, -1 + OPEm], SumS[2, 1, -1 +
OPEm],SumS[1, -1 + OPEm]^3,
    SumS[1, -1 + OPEm] SumS[2, -1 + OPEm], SumS[1, 1, 1, -1 + OPEm]};
```

```
im[z_] := z -> InverseMellin[z, y]
```

```
im[OPEm^(-3)]
```

$$\frac{1}{m^3} \to \frac{1}{2} y^{m-1} \log^2(y)$$

```
im[OPEm^(-2)]
```

$$\frac{1}{m^2} \to -y^{m-1} \log(y)$$

```
im[PolyGamma[0, OPEm]]
```

$$\psi^{(0)}(m) \to -\gamma y^{m-1} \delta(1-y) - \left(\frac{1}{1-y}\right)_+ y^{m-1}$$

```
im[SumS[1, OPEm - 1]]
```

$$S_1(m-1) \to \left(\frac{1}{1-y}\right)_+ \left(-y^{m-1}\right)$$

```
im[SumS[1, OPEm - 1]/(OPEm - 1)]
```

$$\frac{S_1(m-1)}{m-1} \to -y^{m-2} \log(1-y)$$

```
im[SumS[1, OPEm - 1]/(OPEm + 1)]
```

$$\frac{S_1(m-1)}{m+1} \to -y^{m-1} + y^m - y^m \log(1-y) + y^m \log(y)$$

```
im[SumS[1, -1 + OPEm]/OPEm^2]
```

$$\frac{S_1(m-1)}{m^2} \to y^{m-1} \left(\zeta(2) - \mathrm{Li}_2(y) - \frac{1}{2} \log^2(y)\right)$$

```
im[SumS[1, -1 + OPEm]/OPEm]
```

$$\frac{S_1(m-1)}{m} \to y^{m-1}(\log(y) - \log(1-y))$$

```
im[SumS[1, -1 + OPEm]^2/OPEm]
```

$$\frac{S_1^2(m-1)}{m} \to y^{m-1}\left(-3\zeta(2) + \mathrm{Li}_2(1-y) + 2\,\mathrm{Li}_2(y) + \log^2(1-y) + \frac{\log^2(y)}{2}\right)$$

```
im[SumS[2, OPEm - 1]]
```

$$S_2(m-1) \to y^{m-1}\left(\zeta(2)\delta(1-y) + \frac{\log(y)}{1-y}\right)$$

```
im[SumS[2, OPEm - 1]/OPEm]
```

$$\frac{S_2(m-1)}{m} \to y^{m-1}\left(\zeta(2) - \mathrm{Li}_2(1-y) - \frac{1}{2}\log^2(y)\right)$$

```
im[SumS[3, OPEm - 1]]
```

$$S_3(m-1) \to y^{m-1}\left(\zeta(3)\delta(1-y) - \frac{\log^2(y)}{2(1-y)}\right)$$

```
im[SumS[1, 1, OPEm - 1]]
```

$$S_{11}(m-1) \to y^{m-1}\left(\frac{\log(1-y)}{1-y}\right)_+$$

```
im[SumS[2, 1, OPEm - 1]]
```

$$S_{21}(m-1) \to y^{m-1}\left(\frac{\mathrm{Li}_2(y)}{1-y} - \zeta(2)\left(\frac{1}{1-y}\right)_+ + 2\zeta(3)\delta(1-y)\right)$$

```
im[SumS[1, 1, 1, OPEm - 1]]
```

$$S_{111}(m-1) \to -\frac{1}{2} y^{m-1} \left(\frac{\log^2(1-y)}{1-y}\right)_+$$

```
Clear[im, list];
```

12.15 Kummer

`Kummer[i][exp]` applies Kummer relation number **`i`** ($i = 1, ...24, 94, 95, 96$) to all **`Hypergeometric2F1`** in exp.

$i = 94$ corresponds to Eq. 9.131.2, $i = 95$ to Eq. 9.132.1 and $i = 96$ to Eq. 9.132.2 in Gradshteyn & Ryzhik.

12.15.1 See also

Overview, HypergeometricAC.

12.15.2 Examples

```
Hypergeometric2F1[a, b, c, z] == Kummer[2][Hypergeometric2F1[a, b, c, z]]
```

$${}_2F_1(a,b;c;z) = (1-z)^{-a-b+c}\, {}_2F_1(c-a,c-b;c;z)$$

```
Hypergeometric2F1[a, b, c, z] == Kummer[3][Hypergeometric2F1[a, b, c, z]]
```

$${}_2F_1(a,b;c;z) = (1-z)^{-a}\, {}_2F_1\left(a,c-b;c;-\frac{z}{1-z}\right)$$

```
Hypergeometric2F1[a, b, c, z] == Kummer[4][Hypergeometric2F1[a, b, c, z]]
```

$${}_2F_1(a,b;c;z) = (1-z)^{-b}\, {}_2F_1\left(b,c-a;c;-\frac{z}{1-z}\right)$$

```
Hypergeometric2F1[a, b, c, 1 - z] == Kummer[6][Hypergeometric2F1[a, b, c, 1
- z]]
```

$${}_2F_1(a,b;c;1-z) = z^{-a-b+c}\, {}_2F_1(c-a,c-b;c;1-z)$$

```
Hypergeometric2F1[a, b, a + b + 1 - c, 1 - z] ==
Kummer[6][Hypergeometric2F1[a, b, a + b + 1 - c, 1 - z]]
```

$$ {}_2F_1(a,b;a+b-c+1;1-z) = z^{1-c}\,{}_2F_1(a-c+1,b-c+1;a+b-c+1;1-z) $$

```
Hypergeometric2F1[a, b, c, 1 - z] == Kummer[7][Hypergeometric2F1[a, b, c, 1
- z]]
```

$$ {}_2F_1(a,b;c;1-z) = z^{-a}\,{}_2F_1\left(a,c-b;c;-\frac{1-z}{z}\right) $$

```
Hypergeometric2F1[a, b, a + b + 1 - c, 1 - z] ==
Kummer[7][Hypergeometric2F1[a, b, a + b + 1 - c, 1 - z]]
```

$$ {}_2F_1(a,b;a+b-c+1;1-z) = z^{-a}\,{}_2F_1\left(a,a-c+1;a+b-c+1;-\frac{1-z}{z}\right) $$

```
Hypergeometric2F1[a, b, c, 1 - z] == Kummer[8][Hypergeometric2F1[a, b, c, 1
- z]]
```

$$ {}_2F_1(a,b;c;1-z) = z^{-b}\,{}_2F_1\left(b,c-a;c;-\frac{1-z}{z}\right) $$

```
Hypergeometric2F1[a, b, a + b + 1 - c, 1 - z] ==
Kummer[8][Hypergeometric2F1[a, b, a + b + 1 - c, 1 - z]]
```

$$ {}_2F_1(a,b;a+b-c+1;1-z) = z^{-b}\,{}_2F_1\left(b,b-c+1;a+b-c+1;-\frac{1-z}{z}\right) $$

```
Hypergeometric2F1[a, b, c, z^(-1)] == Kummer[10][Hypergeometric2F1[a, b, c,
z^(-1)]]
```

$$ {}_2F_1\left(a,b;c;\frac{1}{z}\right) = (1-z)^{-a-b+c}(-z)^{a+b-c}\,{}_2F_1\left(c-a,c-b;c;\frac{1}{z}\right) $$

```
Hypergeometric2F1[a, a + 1 - c, a + 1 - b, z^(-1)] ==
Kummer[10][Hypergeometric2F1[a, a + 1 - c, a + 1 - b, z^(-1)]]
```

$$ {}_2F_1\left(a, a-c+1; a-b+1; \frac{1}{z}\right) = (1-z)^{-a-b+c}(-z)^{a+b-c}\,{}_2F_1\left(1-b, c-b; a-b+1; \frac{1}{z}\right) $$

```
Hypergeometric2F1[a, b, c, z^(-1)] == Kummer[11][Hypergeometric2F1[a, b, c,
z^(-1)]]
```

$$ {}_2F_1\left(a, b; c; \frac{1}{z}\right) = \left(-\frac{1-z}{z}\right)^{-a}(-z)^{a}\,{}_2F_1\left(a, c-b; c; \frac{1}{1-z}\right) $$

```
Hypergeometric2F1[a, a + 1 - c, a + 1 - b, z^(-1)] ==
Kummer[11][Hypergeometric2F1[a, a + 1 - c, a + 1 - b, z^(-1)]]
```

$$ {}_2F_1\left(a, a-c+1; a-b+1; \frac{1}{z}\right) = \left(-\frac{1-z}{z}\right)^{-a}(-z)^{a}\,{}_2F_1\left(a, c-b; a-b+1; \frac{1}{1-z}\right) $$

```
Hypergeometric2F1[a, b, c, z^(-1)] == Kummer[12][Hypergeometric2F1[a, b, c,
z^(-1)]]
```

$$ {}_2F_1\left(a, b; c; \frac{1}{z}\right) = (1-z)^{-b}(-z)^{b-a}\,{}_2F_1\left(b, c-a; c; \frac{1}{1-z}\right) $$

```
Hypergeometric2F1[a, a + 1 - c, a + 1 - b, z^(-1)] ==
Kummer[12][Hypergeometric2F1[a, a + 1 - c, a + 1 - b, z^(-1)]]
```

$$ {}_2F_1\left(a, a-c+1; a-b+1; \frac{1}{z}\right) = (-z)^{1-c}(1-z)^{-a+c-1}\,{}_2F_1\left(1-b, a-c+1; a-b+1; \frac{1}{1-z}\right) $$

```
Hypergeometric2F1[a, b, c, z^(-1)] == Kummer[14][Hypergeometric2F1[a, b, c,
z^(-1)]]
```

$$ {}_2F_1\left(a, b; c; \frac{1}{z}\right) = (1-z)^{-a-b+c}(-z)^{a+b-c}\,{}_2F_1\left(c-a, c-b; c; \frac{1}{z}\right) $$

```
Hypergeometric2F1[b + 1 - c, b, b + 1 - a, z^(-1)] ==
Kummer[14][Hypergeometric2F1[b + 1 - c, b, b + 1 - a, z^(-1)]]
```

$$ {}_2F_1\left(b, b-c+1; -a+b+1; \frac{1}{z}\right) = (1-z)^{-a-b+c}(-z)^{a+b-c}\,{}_2F_1\left(1-a, c-a; -a+b+1; \frac{1}{z}\right) $$

```
Hypergeometric2F1[a, b, c, z^(-1)] == Kummer[15][Hypergeometric2F1[a, b, c,
z^(-1)]]
```

$$ {}_2F_1\left(a, b; c; \frac{1}{z}\right) = (1-z)^{-b}(-z)^{b}\,{}_2F_1\left(b, c-a; c; \frac{1}{1-z}\right) $$

```
Hypergeometric2F1[b + 1 - c, b, b + 1 - a, z^(-1)] ==
Kummer[15][Hypergeometric2F1[b + 1 - c, b, b + 1 - a, z^(-1)]]
```

$$ {}_2F_1\left(b, b-c+1; -a+b+1; \frac{1}{z}\right) = (1-z)^{-b+c-1}(-z)^{b-c+1}\,{}_2F_1\left(1-a, b-c+1; -a+b+1; \frac{1}{1-z}\right) $$

```
Hypergeometric2F1[a, b, c, z^(-1)] == Kummer[16][Hypergeometric2F1[a, b, c,
z^(-1)]]
```

$$ {}_2F_1\left(a, b; c; \frac{1}{z}\right) = (1-z)^{-a}(-z)^{a}\,{}_2F_1\left(a, c-b; c; \frac{1}{1-z}\right) $$

```
Hypergeometric2F1[b + 1 - c, b, b + 1 - a, z^(-1)] ==
Kummer[16][Hypergeometric2F1[b + 1 - c, b, b + 1 - a, z^(-1)]]
```

$$ {}_2F_1\left(b, b-c+1; -a+b+1; \frac{1}{z}\right) = (1-z)^{-b}(-z)^{b}\,{}_2F_1\left(b, c-a; -a+b+1; \frac{1}{1-z}\right) $$

```
Hypergeometric2F1[a + 1 - c, b + 1 - c, 2 - c, z] ==
Kummer[18][Hypergeometric2F1[a + 1 - c, b + 1 - c, 2 - c, z]]
```

$$ {}_2F_1(a-c+1, b-c+1; 2-c; z) = (1-z)^{-a-b+c}\,{}_2F_1(1-a, 1-b; 2-c; z) $$

```
Hypergeometric2F1[a, b, c, z] == Kummer[18][Hypergeometric2F1[a, b, c, z]]
```

$$ {}_2F_1(a,b;c;z) = (1-z)^{-a-b+c}\,{}_2F_1(c-a,c-b;c;z) $$

```
Hypergeometric2F1[a + 1 - c, b + 1 - c, 2 - c, z] ==
Kummer[19][Hypergeometric2F1[a + 1 - c, b + 1 - c, 2 - c, z]]
```

$$ {}_2F_1(a-c+1,b-c+1;2-c;z) = (1-z)^{-a+c-1}\,{}_2F_1\left(1-b,a-c+1;2-c;\frac{z}{z-1}\right) $$

```
Hypergeometric2F1[a, b, c, z] == Kummer[19][Hypergeometric2F1[a, b, c, z]]
```

$$ {}_2F_1(a,b;c;z) = (1-z)^{-a}\,{}_2F_1\left(a,c-b;c;\frac{z}{z-1}\right) $$

```
Hypergeometric2F1[a + 1 - c, b + 1 - c, 2 - c, z] ==
Kummer[20][Hypergeometric2F1[a + 1 - c, b + 1 - c, 2 - c, z]]
```

$$ {}_2F_1(a-c+1,b-c+1;2-c;z) = (1-z)^{-b+c-1}\,{}_2F_1\left(1-a,b-c+1;2-c;\frac{z}{z-1}\right) $$

```
Hypergeometric2F1[a, b, c, z] == Kummer[20][Hypergeometric2F1[a, b, c, z]]
```

$$ {}_2F_1(a,b;c;z) = (1-z)^{-b}\,{}_2F_1\left(b,c-a;c;\frac{z}{z-1}\right) $$

```
Hypergeometric2F1[c - a, c - b, c + 1 - a - b, 1 - z] ==
Kummer[22][Hypergeometric2F1[c - a, c - b, c + 1 - a - b, 1 - z]]
```

$$ {}_2F_1(c-a,c-b;-a-b+c+1;1-z) = z^{1-c}\,{}_2F_1(1-a,1-b;-a-b+c+1;1-z) $$

```
Hypergeometric2F1[a, b, c, 1 - z] == Kummer[22][Hypergeometric2F1[a, b, c,
1 - z]]
```

$$ {}_2F_1(a,b;c;1-z) = z^{-a-b+c}\,{}_2F_1(c-a,c-b;c;1-z) $$

```
Hypergeometric2F1[c - a, c - b, c + 1 - a - b, 1 - z] ==
Kummer[23][Hypergeometric2F1[c - a, c - b, c + 1 - a - b, 1 - z]]
```

$$ {}_2F_1(c-a,c-b;-a-b+c+1;1-z) = (1-z)^{a-c}\,{}_2F_1\left(1-a,c-a;-a-b+c+1;1-\frac{1}{1-z}\right) $$

```
Hypergeometric2F1[a, b, c, 1 - z] == Kummer[23][Hypergeometric2F1[a, b, c,
1 - z]]
```

$$ {}_2F_1(a,b;c;1-z) = (1-z)^{-a}\,{}_2F_1\left(a,c-b;c;1-\frac{1}{1-z}\right) $$

```
Hypergeometric2F1[c - a, c - b, c + 1 - a - b, 1 - z] ==
Kummer[24][Hypergeometric2F1[c - a, c - b, c + 1 - a - b, 1 - z]]
```

$$ {}_2F_1(c-a,c-b;-a-b+c+1;1-z) = z^{b-c}\,{}_2F_1\left(1-b,c-b;-a-b+c+1;-\frac{1-z}{z}\right) $$

```
Hypergeometric2F1[a, b, c, 1 - z] == Kummer[24][Hypergeometric2F1[a, b, c,
1 - z]]
```

$$ {}_2F_1(a,b;c;1-z) = z^{-b}\,{}_2F_1\left(b,c-a;c;-\frac{1-z}{z}\right) $$

```
Hypergeometric2F1[a, b, c, z] == Kummer[94][Hypergeometric2F1[a, b, c, z]]
```

$$ \begin{aligned} {}_2F_1(a,b;c;z) = {} & \frac{\Gamma(c)(1-z)^{-a-b+c}\Gamma(a+b-c)\,{}_2F_1(c-a,c-b;-a-b+c+1;1-z)}{\Gamma(a)\Gamma(b)} \\ & + \frac{\Gamma(c)\Gamma(-a-b+c)\,{}_2F_1(a,b;a+b-c+1;1-z)}{\Gamma(c-a)\Gamma(c-b)} \end{aligned} $$

```
Hypergeometric2F1[a, b, c, z] == Kummer[95][Hypergeometric2F1[a, b, c, z]]
```

$$ \begin{aligned} {}_2F_1(a,b;c;z) = {} & \frac{(1-z)^{-a}\Gamma(c)\Gamma(b-a)\,{}_2F_1\left(a,c-b;a-b+1;\frac{1}{1-z}\right)}{\Gamma(b)\Gamma(c-a)} \\ & + \frac{(1-z)^{-b}\Gamma(c)\Gamma(a-b)\,{}_2F_1\left(b,c-a;-a+b+1;\frac{1}{1-z}\right)}{\Gamma(a)\Gamma(c-b)} \end{aligned} $$

```
Hypergeometric2F1[a, b, c, z] == Kummer[96][Hypergeometric2F1[a, b, c, z]]
```

$$ {}_2F_1(a,b;c;z) = \frac{(-1)^a z^{-a}\Gamma(c)\Gamma(b-a)\,{}_2F_1\left(a,a-c+1;a-b+1;\frac{1}{z}\right)}{\Gamma(b)\Gamma(c-a)} + \frac{(-1)^b z^{-b}\Gamma(c)\Gamma(a-b)\,{}_2F_1\left(b,b-c+1;-a+b+1;\frac{1}{z}\right)}{\Gamma(a)\Gamma(c-b)} $$

12.16 Lagrangian

Lagrangian["oqu"] gives the unpolarized OPE quark operator.

Lagrangian["oqp"] gives the polarized quark OPE operator.

Lagrangian["ogu"] gives the unpolarized gluon OPE operator.

Lagrangian["ogp"] gives the polarized gluon OPE operator.

Lagrangian["ogd"] gives the sigma-term part of the QCD Lagrangian.

Lagrangian["QCD"] gives the gluon self interaction part of the QCD Lagrangian.

12.16.1 See also

Overview, FeynRule.

12.16.2 Examples

```
Lagrangian["QCD"]
```

$$ -\frac{1}{4} F^{\text{FCGV(a)}}_{\text{FCGV}(\alpha)\text{FCGV}(\beta)} . F^{\text{FCGV(a)}}_{\text{FCGV}(\alpha)\text{FCGV}(\beta)} $$

Twist-2 operator product expansion operators

```
Lagrangian["ogu"]
```

$$ \frac{1}{2} i^{m-1} F^{\text{FCGV(a)}}_{\text{FCGV}(\alpha)\Delta} . \left(D^{\text{FCGV(a)FCGV(b)}}_{\Delta}\right)^{m-2} . F^{\text{FCGV(b)}}_{\text{FCGV}(\alpha)\Delta} $$

```
Lagrangian["ogp"]
```

$$ \frac{1}{2} i^{m} \bar{\epsilon}^{\text{FCGV}(\alpha)\text{FCGV}(\beta)\text{FCGV}(\gamma)\Delta} . F^{\text{FCGV(a)}}_{\text{FCGV}(\beta)\text{FCGV}(\gamma)} . \left(D^{\text{FCGV(a)FCGV(b)}}_{\Delta}\right)^{m-2} . F^{\text{FCGV(b)}}_{\text{FCGV}(\alpha)\Delta} $$

```
Lagrangian["oqu"]
```

$$i^m \bar{\psi}.(\bar{\gamma} \cdot \Delta).{D_\Delta}^{m-1}.\psi$$

```
Lagrangian["oqp"]
```

$$i^m \bar{\psi}.\bar{\gamma}^5.(\bar{\gamma} \cdot \Delta).{D_\Delta}^{m-1}.\psi$$

12.17 Nielsen

`Nielsen[i, j, x]` denotes Nielsen's polylogarithm.

12.17.1 See also

Overview, SimplifyPolyLog.

12.17.2 Examples

```
Nielsen[1, 2, x]
```

$$S_{12}(x)$$

Numerical evaluation is done via **`N[Nielsen[n_, p_, x_]] := (-1)^(n+p-1)/(n-1)!/p! NIntegrate[Log[1-x t]^p Log[t]^(n-1)/t, {t, 0, 1}]`**

```
N[Nielsen[1, 2, .45]]
```

$$0.0728716$$

Some special values are built in.

```
{Nielsen[1, 2, 0], Nielsen[1, 2, -1], Nielsen[1, 2, 1/2], Nielsen[1, 2, 1]}
```

$$\left\{0, \frac{\zeta(3)}{8}, \frac{\zeta(3)}{8}, \zeta(3)\right\}$$

```
Nielsen[1, 2, x, PolyLog -> True]
```

$$-\text{Li}_3(1-x)+\text{Li}_2(1-x)\log(1-x)+\frac{1}{2}\log(x)\log^2(1-x)+\zeta(3)$$

```
Nielsen[1, 3, x, PolyLog -> True]
```

$$-\text{Li}_4(1-x)-\frac{1}{2}\text{Li}_2(1-x)\log^2(1-x)+\text{Li}_3(1-x)\log(1-x)-\frac{1}{6}\log(x)\log^3(1-x)+\frac{\pi^4}{90}$$

```
Nielsen[3, 1, x, PolyLog -> True]
```

$$\text{Li}_4(x)$$

12.18 SimplifyPolyLog

`SimplifyPolyLog[y]` performs several simplifications assuming that the variables occuring in the **`Log`** and **`PolyLog`** functions are between **`0`** and **`1`**.

The simplifications will in general not be valid if the arguments are complex or outside the range between 0 and 1.

12.18.1 See also

Overview, Nielsen.

12.18.2 Examples

```
SimplifyPolyLog[PolyLog[2, 1/x]]
```

$$\zeta(2)+\text{Li}_2(1-x)-\frac{1}{2}\log^2(x)+\log(1-x)\log(x)+i\pi\log(x)$$

```
SimplifyPolyLog[PolyLog[2, x]]
```

$$\zeta(2)-\text{Li}_2(1-x)-\log(1-x)\log(x)$$

```
SimplifyPolyLog[PolyLog[2, 1 - x^2]]
```

$$-\zeta(2) + 2\,\mathrm{Li}_2(1-x) - 2\,\mathrm{Li}_2(-x) - 2\log(x)\log(x+1)$$

```
SimplifyPolyLog[PolyLog[2, x^2]]
```

$$2\zeta(2) - 2\,\mathrm{Li}_2(1-x) + 2\,\mathrm{Li}_2(-x) - 2\log(1-x)\log(x)$$

```
SimplifyPolyLog[PolyLog[2, -x/(1 - x)]]
```

$$-\zeta(2) + \mathrm{Li}_2(1-x) - \frac{1}{2}\log^2(1-x) + \log(x)\log(1-x)$$

```
SimplifyPolyLog[PolyLog[2, x/(x - 1)]]
```

$$-\zeta(2) + \mathrm{Li}_2(1-x) - \frac{1}{2}\log^2(1-x) + \log(x)\log(1-x)$$

```
SimplifyPolyLog[Nielsen[1, 2, -x/(1 - x)]]
```

$$S_{12}(x) - \frac{1}{6}\log^3(1-x)$$

```
SimplifyPolyLog[PolyLog[3, -1/x]]
```

$$\mathrm{Li}_3(-x) + \zeta(2)\log(x) + \frac{\log^3(x)}{6}$$

```
SimplifyPolyLog[PolyLog[3, 1 - x]]
```

$$\mathrm{Li}_3(1-x)$$

```
SimplifyPolyLog[PolyLog[3, x^2]]
```

$$4\,\mathrm{Li}_3(-x) - 4\,\mathrm{Li}_2(1-x)\log(x) - 4S_{12}(1-x) + 4\zeta(2)\log(x) - 2\log(1-x)\log^2(x) + 4\zeta(3)$$

```
SimplifyPolyLog[PolyLog[3, -x/(1 - x)]]
```

$$-\text{Li}_3(1-x)+\text{Li}_2(1-x)\log(x)+S_{12}(1-x)+\zeta(2)\log(1-x)$$
$$-\zeta(2)\log(x)+\frac{1}{6}\log^3(1-x)-\frac{1}{2}\log(x)\log^2(1-x)+\frac{1}{2}\log^2(x)\log(1-x)$$

```
SimplifyPolyLog[PolyLog[3, 1 - 1/x]]
```

$$\text{Li}_2(1-x)\log(x)-\text{Li}_2(1-x)\log(1-x)+S_{12}(1-x)+S_{12}(x)+\frac{\log^3(x)}{6}-\frac{1}{2}\log^2(1-x)\log(x)-\zeta(3)$$

```
SimplifyPolyLog[PolyLog[4, -x/(1 - x)]]
```

$$-\text{Li}_4(x)+\frac{1}{2}\,\text{Li}_2(1-x)\log^2(1-x)-\text{Li}_2(1-x)\log(x)\log(1-x)-S_{13}(x)+S_{22}(x)$$
$$-S_{12}(1-x)\log(1-x)-S_{12}(x)\log(1-x)-\frac{1}{2}\zeta(2)\log^2(1-x)+\zeta(2)\log(x)\log(1-x)$$
$$+\zeta(3)\log(1-x)-\frac{1}{24}\log^4(1-x)+\frac{1}{2}\log(x)\log^3(1-x)-\frac{1}{2}\log^2(x)\log^2(1-x)$$

```
SimplifyPolyLog[Log[a + b/c]]
```

$$\log\left(\frac{ac+b}{c}\right)$$

```
SimplifyPolyLog[Log[1/x]]
```

$$-\log(x)$$

```
SimplifyPolyLog[ArcTanh[x]]
```

$$\frac{1}{2}\log\left(-\frac{x+1}{1-x}\right)$$

```
SimplifyPolyLog[ArcSinh[x]]
```

$$\log\left(\sqrt{x^2+1}+x\right)$$

```
SimplifyPolyLog[ArcCosh[x]]
```

$$\log\left(\sqrt{x^2-1}+x\right)$$

12.19 SPL

SPL is an abbreviation for **SimplifyPolyLog**.

12.19.1 See also

Overview, SimplifyPolyLog.

12.19.2 Examples

12.20 SplittingFunction

SplittingFunction[pxy] is a database of splitting functions in the $\overline{\text{MS}}$ scheme.

SplittingFunction["Pqq", x], **SplittingFunction["Pqg", x]**, **SplittingFunction["Pgq", x]** and **SplittingFunction["Pgg", x]** yield the lowest order splitting functions.

SplittingFunction["PQQS", x], **SplittingFunction["PQQNS", x]** and **SplittingFunction["PQG", x]** are the next to leading order splitting functions.

SplittingFunction has an option Polarization.

SplittingFunction["Pqq", x, Polarization -> 0] returns the unpolarized and **SplittingFunction["Pqq", x, Polarization -> 1]** the polarized splitting functions.

12.20.1 See also

Overview, AnomalousDimension.

12.20.2 Examples

Unpolarized case:

In general the argument should be a string, but if the variables Pqq etc. have no value, you can omit the " ".

```
SplittingFunction[Pqq, Polarization -> 0] /. FCGV[z_] :> ToExpression[z]
```

$$C_F \left(6\delta(1-x) - 4x + 8\left(\frac{1}{1-x}\right)_+ - 4\right)$$

```
SplittingFunction[Pqg, Polarization -> 0] /. FCGV[z_] :> ToExpression[z]
```

$$\left(16x^2 - 16x + 8\right) T_f$$

```
SplittingFunction[Pgq, Polarization -> 0] /. FCGV[z_] :> ToExpression[z]
```

$$\left(4x + \frac{8}{x} - 8\right) C_F$$

```
SplittingFunction[Pgg, Polarization -> 0] /. FCGV[z_] :> ToExpression[z]
```

$$8C_A\left(-x^2 + \frac{11}{12}\delta(1-x) + x + \left(\frac{1}{1-x}\right)_+ + \frac{1}{x} - 2\right) - \frac{8}{3}N_f T_f \delta(1-x)$$

```
SplittingFunction[aqq, Polarization -> 0] /. FCGV[z_] :> ToExpression[z]
```

$$C_F\left((7 - 4\zeta(2))\delta(1-x) + 2x + (2x+2)\log((1-x)x) - 4\left(\frac{\log(x)}{1-x} + \left(\frac{\log(1-x)}{1-x}\right)_+\right) - 4\right)$$

```
SplittingFunction[agq, Polarization -> 0] /. FCGV[z_] :> ToExpression[z]
```

$$C_F\left(-2x - \frac{4}{x} + \left(-2x - \frac{4}{x} + 4\right)\log((1-x)x) + 2\right)$$

```
SplittingFunction[aqg, Polarization -> 0] /. FCGV[z_] :> ToExpression[z]
```

$$T_f \left(\left(-8x^2+8x-4\right)\log((1-x)x)-4\right)$$

```
SplittingFunction[agg, Polarization -> 0] /. FCGV[z_] :> ToExpression[z]
```

Polarized case:

```
SplittingFunction[Pqq, Polarization -> 1] /. FCGV[z_] :> ToExpression[z]
```

$$C_F \left(6\delta(1-x)-4x+8\left(\frac{1}{1-x}\right)_+-4\right)$$

```
SplittingFunction[Pqg, Polarization -> 1] /. FCGV[z_] :> ToExpression[z]
```

$$(16x-8)T_f$$

```
SplittingFunction[Pgq, Polarization -> 1] /. FCGV[z_] :> ToExpression[z]
```

$$(8-4x)C_F$$

```
SplittingFunction[Pgg, Polarization -> 1] /. FCGV[z_] :> ToExpression[z]
```

$$C_A \left(\frac{22}{3}\delta(1-x)-16x+8\left(\frac{1}{1-x}\right)_++8\right)-\frac{8}{3}N_f T_f \delta(1-x)$$

```
SplittingFunction[aqq, Polarization -> 1] /. FCGV[z_] :> ToExpression[z]
```

$$C_F \left((7-4\zeta(2))\delta(1-x)+8(1-x)+2x+(2x+2)\log((1-x)x)-4\left(\frac{1}{1-x}\right)_+\log(x)-4\left(\frac{\log(1-x)}{1-x}\right)_+-4\right)$$

```
SplittingFunction[agq, Polarization -> 1] /. FCGV[z_] :> ToExpression[z]
```

$$C_F(-4x+(2x-4)\log((1-x)x)+2)$$

```
SplittingFunction[agqd, Polarization -> 1] /. FCGV[z_] :> ToExpression[z]
```

$$C_F\left((2x-4)\log((1-x)x)-2\right)$$

```
SplittingFunction[aqg, Polarization -> 1] /. FCGV[z_] :> ToExpression[z]
```

$$T_f\left((4-8x)\log((1-x)x)-4\right)$$

```
SplittingFunction[aqgd, Polarization -> 1] /. FCGV[z_] :> ToExpression[z]
```

$$T_f\left((4-8x)\log((1-x)x)-4\right)$$

```
SplittingFunction[agg, Polarization -> 1] /. FCGV[z_] :> ToExpression[z]
```

$$C_A\left(\left(\frac{67}{9}-4\zeta(2)\right)\delta(1-x)+(8x-4)\log((1-x)x)-4\left(\frac{\log(x)}{1-x}+\left(\frac{\log(1-x)}{1-x}\right)_+\right)+2\right)-\frac{20}{9}T_f\,\delta(1-x)$$

```
SplittingFunction[aggd, Polarization -> 1] /. FCGV[z_] :> ToExpression[z]
```

$$C_A\left(\left(\frac{67}{9}-4\zeta(2)\right)\delta(1-x)+(8x-4)\log((1-x)x)-4\left(\frac{\log(x)}{1-x}+\left(\frac{\log(1-x)}{1-x}\right)_+\right)+2\right)-\frac{20}{9}T_f\,\delta(1-x)$$

```
SplittingFunction[PQQS, Polarization -> 1] /. FCGV[z_] :> ToExpression[z]
```

$$C_F T_f\left(16(1-x)-16(x+1)\log^2(x)+(48x-16)\log(x)\right)$$

```
SplittingFunction[PQQNS, Polarization -> 1] /. FCGV[z_] :> ToExpression[z]
```

$$
\begin{aligned}
&-8C_F\left(C_F-\frac{C_A}{2}\right)\left(\frac{\left(x^2+1\right)\left(-2\zeta(2)-4\,\mathrm{Li}_2(-x)+\log^2(x)-4\log(x+1)\log(x)\right)}{x+1}+4(1-x)\right.\\
&\left.+2(x+1)\log(x)\right)+C_AC_F\left(\frac{4\left(x^2+1\right)\log^2(x)}{1-x}+8\zeta(2)(x+1)+\left(\frac{536}{9}-16\zeta(2)\right)\left(\frac{1}{1-x}\right)_+\right.\\
&\left.+\delta(1-x)\left(\frac{88\zeta(2)}{3}-24\zeta(3)+\frac{17}{3}\right)+\frac{4}{9}(53-187x)-\frac{4}{3}\left(5x-\frac{22}{1-x}+5\right)\log(x)\right)\\
&+C_FN_f\left(-\frac{8\left(x^2+1\right)\log(x)}{3(1-x)}+\left(-\frac{16\zeta(2)}{3}-\frac{2}{3}\right)\delta(1-x)+\frac{88x}{9}-\frac{80}{9}\left(\frac{1}{1-x}\right)_+-\frac{8}{9}\right)\\
&+C_F^2\left(-\frac{16\left(x^2+1\right)\log(1-x)\log(x)}{1-x}+\delta(1-x)(-24\zeta(2)+48\zeta(3)+3)\right.\\
&\left.-40(1-x)-4(x+1)\log^2(x)-8\left(2x+\frac{3}{1-x}\right)\log(x)\right)
\end{aligned}
$$

```
SplittingFunction[PQG, Polarization -> 1] /. FCGV[z_] :> ToExpression[z]
```

$$
\begin{aligned}
&4C_AT_f\left(-8\zeta(2)+(-16x-8)\,\mathrm{Li}_2(-x)-44x+(4-8x)\log^2(1-x)+(-8x-4)\log^2(x)+(16x-16)\log(1-x)\right.\\
&\left.+(32x+4)\log(x)+(-16x-8)\log(x)\log(x+1)+48\right)+4C_FT_f\left(8\zeta(2)-16\zeta(2)x+54x\right.\\
&\left.+(8x-4)\log^2(1-x)+(4x-2)\log^2(x)+(16-16x)\log(1-x)+(8-16x)\log(x)\log(1-x)-18\log(x)-44\right)
\end{aligned}
$$

```
SplittingFunction[PGQ, Polarization -> 1] /. FCGV[z_] :> ToExpression[z]
```

$$
\begin{aligned}
&(C_AC_F)\cdot\left((16x+32)\,\mathrm{Li}_2(-x)+16\zeta(2)x+\frac{280x}{9}+(16-8x)\log^2(1-x)+(8x+16)\log^2(x)\right.\\
&\left.+\left(\frac{8x}{3}+\frac{80}{3}\right)\log(1-x)+(16x-32)\log(x)\log(1-x)+(32-104x)\log(x)+(16x+32)\log(x)\log(x+1)\right.\\
&\left.+\frac{328}{9}\right)+(C_FT_f)\cdot\left(-\frac{32x}{9}+\left(\frac{32x}{3}-\frac{64}{3}\right)\log(1-x)-\frac{128}{9}\right)+C_F^2\cdot\left(32x+(8x-16)\log^2(1-x)\right.\\
&\left.+(8-4x)\log^2(x)+(-8x-16)\log(1-x)-(32x+64)\log(x)+(36x+48)\log(x)-68\right)
\end{aligned}
$$

```
SplittingFunction[PGG, Polarization -> 1] /. FCGV[z_] :> ToExpression[z]
```

$$
\begin{aligned}
&C_AT_f\left(-\frac{32}{3}\delta(1-x)+\frac{608x}{9}-\frac{160}{9}\left(\frac{1}{1-x}\right)_+-\left(\frac{32x}{3}+\frac{32}{3}\right)\log(x)-\frac{448}{9}\right)\\
&+C_A^2\left(\left(64x+\frac{32}{x+1}+32\right)\,\mathrm{Li}_2(-x)+\frac{64}{3}\delta(1-x)+\zeta(2)\left(64x-16\left(\frac{1}{1-x}\right)_+-\frac{16}{x+1}\right)\right.\\
&\left.+24\zeta(3)\delta(1-x)-\frac{388x}{9}+\frac{536}{9}\left(\frac{1}{1-x}\right)_+-\left(\frac{8}{x+1}+\frac{8}{1-x}+32\right)\log^2(x)+\left(\frac{232}{3}-\frac{536x}{3}\right)\log(x)\right.\\
&\left.+\left(64x-\frac{32}{1-x}-32\right)\log(1-x)\log(x)+\left(64x+\frac{32}{x+1}+32\right)\log(x+1)\log(x)-\frac{148}{9}\right)\\
&+C_FT_f\left(-8\delta(1-x)+80x+(-16x-16)\log^2(x)+(16x-80)\log(x)-80\right)
\end{aligned}
$$

13 Options

13.1 $DisableMemSet

The boolean setting of **$DisableMemSet** allows to disable memoization that is activated via **MemSet**. This can be useful for debugging purposes.

13.1.1 See also

Overview, MemSet.

13.1.2 Examples

```
$DisableMemSet
```

False

13.2 $FAPatch

$FAPatch switches on and off checking for an unpatched FeynArts installation on FeynCalc startup. Default value: **True**.

13.2.1 See also

Overview

13.2.2 Examples

```
$FAPatch
```

True

13.3 $FCCheckContext

If **$FCCheckContext** set to **True**, FeynCalc will try to detect unwanted leakage of internal objects into the **Global** or **FeynCalc** contexts. The default value is **False**, however **$FCCheckContext** will be automatically enabled in the development version.

13.3.1 See also

Overview

13.3.2 Examples

```
$FCCheckContext
```

True

13.4 $FCCloudTraditionalForm

$FCCloudTraditionalForm determines whether the cell output will be done in **TraditionalForm** when FeynCalc is run in Wolfram Cloud. This is done by setting **$Post=TraditionalForm**. The default value of **$FCCloudTraditionalForm** is **True**.

13.4.1 See also

Overview

13.4.2 Examples

```
$FCCloudTraditionalForm
```

True

13.5 $FCTraditionalFormOutput

The Boolean setting of **$FCTraditionalFormOutput** determines which output format type should be used in the notebook front end when FeynCalc is loaded. If set to **True**, FeynCalc will activate the **TraditionalForm** output. Otherwise, the **StandardForm** output (Mathematica's default) will be used.

This setting only changes the output format of the current notebook, i.e. it is not persistent and will not modify the global options of Mathematica.

If unsure, it is recommended to set **$FCTraditionalFormOutput** to **True**, so that you can benefit from the nice FeynCalc typesetting for various QFT quantities.

13.5.1 See also

Overview, FCEnableTraditionalFormOutput, FCDisableTraditionalFormOutput.

13.5.2 Examples

13.6 $FeynArtsDirectory

`$FeynArtsDirectory` specifies the location of FeynArts.

13.6.1 See also

Overview, $FeynCalcDirectory.

13.6.2 Examples

```
$FeynArtsDirectory
```

/home/vs/.Mathematica/Applications/FeynCalc/FeynArts

13.7 $FeynCalcDevelopmentVersion

The boolean setting of **`$FeynCalcDevelopmentVersion`** determines whether the current version is a development or a stable version.

13.7.1 See also

Overview

13.7.2 Examples

13.8 $FeynCalcDirectory

`$FeynCalcDirectory` specifies the location of FeynCalc.

13.8.1 See also

Overview, $FeynArtsDirectory.

13.8.2 Examples

```
$FeynCalcDirectory
```

/home/vs/.Mathematica/Applications/FeynCalc/

13.9 $FeynCalcStartupMessages

$FeynCalcStartupMessages specifies whether some additional information about FeynCalc should be displayed when the package is loaded. Its value must be set before loading FeynCalc. The default value is **True**.

13.9.1 See also

Overview

13.9.2 Examples

```
$FeynCalcStartupMessages
```

False

13.10 $LoadAddOns

$LoadAddOns[] specifies which addons should be loaded with FeynCalc. E.g. **$LoadAddOns={"FeynHelpers"}**. The value must be set before loading FeynCalc. The default value is **False**.

13.10.1 See also

Overview, $RenameFeynCalcObjects.

13.10.2 Examples

13.11 $Multiplications

$Multiplications is a set functions which should be treated as (commutative or non-commutative) multiplications by **FieldDerivative**.

13.11.1 See also

Overview, FieldDerivative.

13.11.2 Examples

```
$Multiplications
```

{Times, Dot}

13.12 $RenameFeynCalcObjects

$RenameFeynCalcObjects specifies a list of replacement rules that allow to rename FeynCalc objects on the fly to avoid conflicts with other package before FeynCalc is loaded (monkey patching). The value of **$RenameFeynCalcObjects** must be specified before loading FeynCalc.

13.12.1 See also

Overview, $LoadAddOns.

13.12.2 Examples

The following code (when executed on a fresh kernel with the last two lines uncommented) allows to load FeynCalc and Roman Lee's LiteRed on the same kernel without shadowing

```
$RenameFeynCalcObjects = {"MetricTensor" -> "FCMetricTensor", "Factor1" ->
"FCFactor1", "Factor2" -> "FCFactor2"};
(*
<<FeynCalc`
<<LiteRed`
*)
```

13.13 $Containers

$Containers is a set of heads over which **FieldDerivative** should distribute, in the following sense: Let **c** be a member of **$Containers**. Then **FieldDerivative[c[f, g, h][x], x, {mu}] -> c[FieldDerivative[f[x], x, {mu}], FieldDerivative[f[x], x, {mu}], FieldDerivative[f[x], x, {mu}]]**.

13.13.1 See also

Overview, FCCheckVersion.

13.13.2 Examples

```
$Containers
```

{}

13.14 $DistributiveFunctions

$DistributiveFunctions is a set of functions over which **FieldDerivative** should be distributed.

13.14.1 See also

Overview, FCCheckVersion.

13.14.2 Examples

```
$DistributiveFunctions
```

{Conjugate, Transpose}

13.15 $FCAdvice

If **$FCAdvice** is set to **True**, FeynCalc will display some advices on optimal Mathematica configuration for using FeynCalc.

13.15.1 See also

Overview

13.15.2 Examples

```
$FCAdvice
```

True

13.16 $FCMemoryAvailable

$FCMemoryAvailable is a global variable which is set to an integer **n**, where **n** is the available amount of main memory in MB. The default is **1/4** of **$SystemMemory**. It should be increased if possible. The higher $FCMemoryAvailable can be, the more intermediate steps do not have to be repeated by FeynCalc.

13.16.1 See also

Overview, MemSet, FCMemoryAvailable.

13.16.2 Examples

```
$SystemMemory
```

$$32898408448$$

```
Floor[$SystemMemory/10^6/4]
```

$$8224$$

```
$FCMemoryAvailable
```

$$8224$$

13.17 $FCShowIEta

The Boolean setting of **$FCShowIEta** determines whether $i\eta$ should be displayed in the typesetting of propagator objects (except for **FAD**s) or not. This setting affects only the TraditionalForm typesetting and has absolutely no influence on the internal handling of propagator denominators in FeynCalc.

13.17.1 See also

Overview, SFAD, CFAD, GFAD.

13.17.2 Examples

```
$FCShowIEta

SFAD[{p, m^2}]
```

True

$$\frac{1}{(p^2 - m^2 + i\eta)}$$

```
$FCShowIEta = False

SFAD[{p, m^2}]
```

False

$$\frac{1}{(p^2 - m^2)}$$

```
$FCShowIEta = True
```

True

13.18 $FortranContinuationCharacter

`$FortranContinuationCharacter` is the continuation character used in Write2.

13.18.1 See also

Overview, Write2.

13.18.2 Examples

```
$FortranContinuationCharacter
```

&

13.19 $KeepLogDivergentScalelessIntegrals

$KeepLogDivergentScalelessIntegrals is an experimental global option that forces FeynCalc not to set 1-loop integrals of type $\frac{1}{7}q^4$ to zero. This is useful when one has to explicitly distinguish between IR- and UV-divergences in dimensional regularization. Notice that OneLoop is not guaranteed to respect this option.

13.19.1 See also

Overview

13.19.2 Examples

```
$KeepLogDivergentScalelessIntegrals
```

False

13.20 $LeviCivitaSign

$LeviCivitaSign is a global variable that determines the sign in the result of a Dirac trace of four gamma matrices and γ^5. **$LeviCivitaSign** is by default set to **-1** which corresponds to the convention **Tr[LC[a, b, c, d, 5]] = -4*I*Eps[a, b, c, d]**. Setting **$LeviCivitaSign=-I** will switch to the FORM-convention.

13.20.1 See also

Overview, LC, Eps, DiracTrace.

13.20.2 Examples

```
$LeviCivitaSign

Tr[GA[\[Mu], \[Nu], \[Rho], \[Sigma], 5]]
```

$$-1$$

$$-4i\bar{\epsilon}^{\mu\nu\rho\sigma}$$

This sets the same convention as in FORM

```
$LeviCivitaSign = -I;

Tr[GA[\[Mu], \[Nu], \[Rho], \[Sigma], 5]]
```

$$4\bar{\epsilon}^{\mu\nu\rho\sigma}$$

Back to the standard value

```
$LeviCivitaSign = -1;
```

13.21 $LimitTo4

$LimitTo4 is a variable with default setting **False**. If set to **True**, the limit **Dimension -> 4** is performed after tensor integral decomposition.

$LimitTo4 is a global variable that determines whether UV-divergent Passarino-Veltman functions are simplified by taking the limit $D - 4 \to 0$.

A generic IR-finite Passarino-Veltman function X can be written as $X = \frac{a}{D-4} + b + \mathcal{O}(\varepsilon)$, with a being the prefactor of the pole and b being the finite part. Therefore, products of such functions with coefficients that are rational functions of D with $f(D) = f(4) + (D-4)f'(4) + \mathcal{O}(\varepsilon^2)$ can be simplified to $f(D)X = f(4)X + af'(4) + \mathcal{O}(\varepsilon)$, whenever such products appear in the reduction.

This relation is correct only if the Passarino-Veltman functions have no IR divergences, or if such divergences are regulated without using dimensional regularization.

For this reason, even when $LimitTo4 is set to **True**, the simplifications are applied only to A and B functions. Although B functions can exhibit an IR divergence, such integrals are zero in dimensional regularization, so that no mixing of ε-terms from IR and UV can occur.

The default value of **$LimitTo4** is **False**. Notice that even when the switch is set to **True**, it will essentially affect only the Passarino-Veltman reduction via **PaVeReduce**.

The modern and more flexible way to simplify amplitudes involving IR-finite **PaVe** functions is to use the special routine **PaVeLimitTo4**.

13.21.1 See also

Overview, PaVe, PaVeReduce, OneLoop, $LimitTo4IRUnsafe, PaVeLimitTo4.

13.21.2 Examples

```
$LimitTo4
```

False

13.22 $LimitTo4IRUnsafe

$LimitTo4IRUnsafe is a global variable that determines whether the simplifications described in **$LimitTo4** are applied also to C and D Passarino-Veltman functions. In this case it is assumed that such functions are either IR finite, or the IR divergences are regulated without using dimensional regularization (i.e. by introducing fictitious masses). Otherwise the results will be inconsistent. If this option is activated, it is the task of the user to ensure that IR divergences are properly regulated, such that no mixing of ε from IR and UV can occur. The default value is **False**.

13.22.1 See also

Overview, PaVe, PaVeReduce, OneLoop, $LimitTo4, PaVeLimitTo4.

13.22.2 Examples

```
$LimitTo4IRUnsafe
```

False

13.23 $VeryVerbose

$VeryVerbose is a global variable with default setting **0**. If set to **1**, **2**, . . . , less and more intermediate comments and informations are displayed during calculations.

13.23.1 See also

Overview, FCVerbose, FCPrint.

13.23.2 Examples

```
$VeryVerbose
```

0

```
$VeryVerbose = 3;
```

```
Collect2[Expand[(x - y - z)^6], x];
```

Collect2: Entering Collect2.

Collect2: Entering with: $x^6 - 6yx^5 - 6zx^5 + 15y^2x^4 + 15z^2x^4 + 30yzx^4 - 20y^3x^3 - 20z^3x^3 - 60yz^2x^3 - 60y^2zx^3 + 15y^4x^2 + 15z^4x^2 + 60yz^3x^2 + 90y^2z^2x^2 + 60y^3zx^2 - 6y^5x - 6z^5x - 30yz^4x - 60y^2z^3x - 60y^3z^2x - 30y^4zx + y^6 + z^6 + 6yz^5 + 15y^2z^4 + 20y^3z^3 + 15y^4z^2 + 6y^5z$

Collect2: After factoring out 1 : $x^6 - 6yx^5 - 6zx^5 + 15y^2x^4 + 15z^2x^4 + 30yzx^4 - 20y^3x^3 - 20z^3x^3 - 60yz^2x^3 - 60y^2zx^3 + 15y^4x^2 + 15z^4x^2 + 60yz^3x^2 + 90y^2z^2x^2 + 60y^3zx^2 - 6y^5x - 6z^5x - 30yz^4x - 60y^2z^3x - 60y^3z^2x - 30y^4zx + y^6 + z^6 + 6yz^5 + 15y^2z^4 + 20y^3z^3 + 15y^4z^2 + 6y^5z$

Collect2: Applying initial function.

Collect2: After initial function $x^6 - 6yx^5 - 6zx^5 + 15y^2x^4 + 15z^2x^4 + 30yzx^4 - 20y^3x^3 - 20z^3x^3 - 60yz^2x^3 - 60y^2zx^3 + 15y^4x^2 + 15z^4x^2 + 60yz^3x^2 + 90y^2z^2x^2 + 60y^3zx^2 - 6y^5x - 6z^5x - 30yz^4x - 60y^2z^3x - 60y^3z^2x - 30y^4zx + y^6 + z^6 + 6yz^5 + 15y^2z^4 + 20y^3z^3 + 15y^4z^2 + 6y^5z$

Collect2: Monomials w.r.t which we will collect: $\{x\}$

Collect2: Factoring with Factor

Collect2: Factoring part done, timing: 0.000423

Collect2: Expanding

Collect2: Expanding done, timing: 0.000252

Collect2: Separating the part free of the monomials (linear part)

FCSplit: Splitting, timing: 0.000333

Collect2: Separation done, timing: 0.000812

Collect2: Part that contains the monomials: $x^6 - 6yx^5 - 6zx^5 + 15y^2x^4 + 15z^2x^4 + 30yzx^4 - 20y^3x^3 - 20z^3x^3 - 60yz^2x^3 - 60y^2zx^3 + 15y^4x^2 + 15z^4x^2 + 60yz^3x^2 + 90y^2z^2x^2 + 60y^3zx^2 - 6y^5x - 6z^5x - 30yz^4x - 60y^2z^3x - 60y^3z^2x - 30y^4zx$

Collect2: Linear part: $y^6 + 6zy^5 + 15z^2y^4 + 20z^3y^3 + 15z^4y^2 + 6z^5y + z^6$

Collect2: Factoring the linear part

Collect2: Factoring done, timing: 0.002543

Collect2: Wrapping the momomials with special heads.

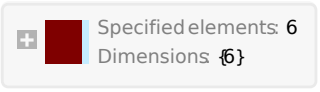
Collect2: Wrapping done, timing: 0.002084

Collect2: nx: $-6\ \text{FeynCalc`Collect`Private`monomialHead}(x)y^5 - 30z\ \text{FeynCalc`Collect`Private`monomialHead}(x)y^4 + 15\ \text{FeynCalc`Collect`Private`monomialHead}\left(x^2\right)y^4 - 60z^2\ \text{FeynCalc`Collect`Private`monomialHead}(x)y^3 + 60z\ \text{FeynCalc`Collect`Private`monomialHead}\left(x^2\right)y^3 - 20\ \text{FeynCalc`Collect`Private`monomialHead}\left(x^3\right)y^3 - 60z^3\ \text{FeynCalc`Collect`Private`monomialHead}(x)y^2 + 90z^2\ \text{FeynCalc`Collect`Private`monomialHead}\left(x^2\right)y^2 - 60z\ \text{FeynCalc`Collect`Private`monomialHead}\left(x^3\right)y^2 + 15\ \text{FeynCalc`Collect`Private`monomialHead}\left(x^4\right)y^2 - 30z^4\ \text{FeynCalc`Collect`Private`monomialHead}(x)y + 60z^3\ \text{FeynCalc`Collect`Private`monomialHead}\left(x^2\right)y - 60z^2\ \text{FeynCalc`Collect`Private`monomialHead}\left(x^3\right)y + 30z\ \text{FeynCalc`Collect`Private`monomialHead}\left(x^4\right)y - 6\ \text{FeynCalc`Collect`Private`monomialHead}\left(x^5\right)y - 6z^5\ \text{FeynCalc`Collect`Private`monomialHead}(x) + 15z^4\ \text{FeynCalc`Collect`Private`monomialHead}\left(x^2\right) - 20z^3\ \text{FeynCalc`Collect`Private`monomialHead}\left(x^3\right) + 15z^2\ \text{FeynCalc`Collect`Private`monomialHead}\left(x^4\right) - 6z\ \text{FeynCalc`Collect`Private`monomialHead}\left(x^5\right) + \text{FeynCalc`Collect`Private`monomialHead}\left(x^6\right)$

Collect2: tv: $\{$FeynCalc`Collect`Private`monomialHead(x),
FeynCalc`Collect`Private`monomialHead$\left(x^2\right)$, FeynCalc`Collect`Private`monomialHead$\left(x^3\right)$,
FeynCalc`Collect`Private`monomialHead$\left(x^4\right)$, FeynCalc`Collect`Private`monomialHead$\left(x^5\right)$,
FeynCalc`Collect`Private`monomialHead$\left(x^6\right)\}$

Collect2: There are 6 momomials to collect

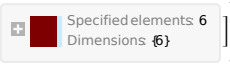
Collect2: Computing CoefficientArrays.

Collect2: CoefficientArrays: {0, SparseArray[]}

Collect2: Collecting the monomials.

Collect2: tvm: $\{x, x^2, x^3, x^4, x^5, x^6\}$

Collect2: Done collecting the monomials, timing: 0.001726

Collect2: preliminary new: FeynCalc`Collect`Private`dotHold(
FeynCalc`Collect`Private`coeffHead(SparseArray[], {x, x^2, x^3, x^4, x^5, x^6}))

Collect2: Obtaining the final result.

Collect2: The final result is ready, timing: 0.001661

Collect2: new: FeynCalc`Collect`Private`unity$x^6 - 6$ FeynCalc`Collect`Private`unity$(y+z)x^5$
$+ 15$ FeynCalc`Collect`Private`unity$(y+z)^2x^4 - 20$ FeynCalc`Collect`Private`unity$(y+z)^3x^3$
$+ 15$ FeynCalc`Collect`Private`unity$(y+z)^4x^2 - 6$ FeynCalc`Collect`Private`unity$(y+z)^5x$

Collect2: Releasing tempIso.

Collect2: Done releasing tempIso, timing: 0.000271

Collect2: Putting re togehter.

Collect2: Done putting re togehter, timing: 0.000250

Collect2: Done releasing tempIso, timing: 0.000510

Collect2: Leaving.

Collect2: Leaving with$x^6 - 6(y+z)x^5 + 15(y+z)^2x^4 - 20(y+z)^3x^3 + 15(y+z)^4x^2 - 6(y+z)^5x + (y+z)^6$

```
$VeryVerbose = 0;
```

13.24 $Abbreviations

$Abbreviations are a list of string substitution rules used when generating names for storing intermediate results. It is used by **OneLoop** and **PaVeReduce**. The elements of the list should be of the form **"name" -> "abbreviation"**.

13.24.1 See also

Overview, Abbreviation, OneLoop, PaVeReduce, WriteOut, WriteOutPaVe.

13.24.2 Examples

13.25 $AL

$AL is the head for dummy indices which may be introduced by **Amputate** and **Uncontract**. By default it is unset, but may be set to anything.

13.25.1 See also

Overview, Amputate, Uncontract.

13.25.2 Examples

```
Uncontract[ScalarProduct[p, q], q, Pair -> All]
```

$$\overline{p}^{\$AL(\$19)}\overline{q}^{\$AL(\$19)}$$

```
$AL = \[Mu];

Uncontract[ScalarProduct[p, q], q, Pair -> All]
```

$$\overline{p}^{\mu(\$20)}\overline{q}^{\mu(\$20)}$$

```
$AL =.;
```

13.26 $FCTensorList

`$FCTensorList` contains a list of all tensor heads present.

13.26.1 See also

Overview, FCTensor.

13.26.2 Examples

```
$FCTensorList
```

$$\{\text{Pair}, \text{Eps}, \text{CartesianPair}\}$$

13.27 $FeynCalcVersion

`$FeynCalcVersion` is a string that represents the version of FeynCalc.

13.27.1 See also

Overview, FCCheckVersion.

13.27.2 Examples

```
$FeynCalcVersion
```

10.0.0

13.28 $MU

$MU is the head of dummy indices which may be introduced by **Chisholm**, **Contract**, **DiracSimplify**, **FermionSpinSum** and various QCD functions. By default it is unset, but can be set to anything.

13.28.1 See also

Overview, Chisholm, Contract, DiracSimplify.

13.28.2 Examples

```
$MU
```

$MU

13.29 $NonComm

$NonComm contains a list of all noncommutative heads present.

13.29.1 See also

Overview, NonCommQ, NonCommutative.

13.29.2 Examples

```
$NonComm
```

{DiracGamma, DiracSigma, GA, GAD, GAE, GS, GSD, GSE, LeftPartialD, LeftRightPartialD, LeftRightPartialD2, FCPartialD, OPESum, QuantumField, RightPartialD, Spinor, SpinorU, SpinorUBar, SpinorV, SpinorVBar, SUNT, PauliSigma, PauliXi, PauliEta, TGA, CGA, CGAD, CGAE, CGS, CGSD, CGSE, SI, SID, SIE, CSI, CSID, CSIE, FeynCalc`Package`dotsimpHold, FCChargeConjugateTransposed, CovariantD, FieldStrength, QuarkGluonVertex, QuarkPropagator, Twist2AlienOperator, Twist2CounterOperator, Twist2QuarkOperator, ChiralityProjector, Spinor, DiracMatrix, DiracSlash}

13.30 $ScalarProducts

$ScalarProducts contains a list of all vector pairs for which a scalar product value has been defined.

13.30.1 See also

Overview, ScalarProduct.

13.30.2 Examples

```
$ScalarProducts
```

$$\{\Delta \quad \Delta\}$$

13.31 A0ToB0

A0ToB0 is an option for **A0**. If set to **True**, **A0[m^2]** is expressed by **(1 + B0[0, m^2, m^2]) m^2**.

13.31.1 See also

Overview, A0, B0, C0, D0, PaVe.

13.31.2 Examples

```
A0[m^2]
```

$$\mathrm{A}_0\left(m^2\right)$$

```
A0[m^2, A0ToB0 -> True]
```

$$-\frac{2m^2\ \mathrm{B}_0\left(0, m^2, m^2\right)}{2-D}$$

13.32 AuxiliaryMomenta

AuxiliaryMomenta is an option of **FCLoopPropagatorsToLineMomenta**, **FCLoopIntegralToGraph** and other functions. It specifies auxiliary momenta that do not correspond to external legs, i.e. don't really flow through the lines, e.g. n and $\bar{n}$ in SCET or v in HQET.

13.32.1 See also

Overview, FCLoopPropagatorsToLineMomenta, FCLoopIntegralToGraph.

13.32.2 Examples

13.33 B0Real

B0Real is an option of **B0** (default **False**). If set to **True**, **B0** is assumed to be real and the relation **B0[a, 0, a] = 2 + B0[0, a, a]** is applied.

13.33.1 See also

Overview, B0.

13.33.2 Examples

By default the arguments are not assumed real.

```
B0[s, 0, s]
```

$$\mathrm{B}_0(s, 0, s)$$

With **B0Real->True**, transformation is done.

```
B0[s, 0, s, B0Real -> True, B0Unique -> True]
```

$$\mathrm{B}_0(0, s, s) + 2$$

13.34 B0Unique

B0Unique is an option of **B0**. If set to **True**, **B0[0, 0, m2]** is replaced with **(B0[0, m2, m2]+1)** and **B0[m2, 0, m2]** simplifies to **(B0[0, m2, m2]+2)**.

13.34.1 See also

Overview, B0.

13.34.2 Examples

By default no transformation is done.

```
B0[0, 0, s]
```

$$\mathrm{B}_0(0, 0, s)$$

With **B0Real->True** following transformation is applied

```
B0[0, 0, s, B0Unique -> True]
```

$$\mathrm{B}_0(0, s, s) + 1$$

13.35 Bracket

Bracket is an option of **Convolute**, specifying the variable with respect to which the result is collected.

13.35.1 See also

Overview, Convolute.

13.35.2 Examples

13.36 BReduce

BReduce is an option for **B0**, **B00**, **B1**, **B11** determining whether reductions to **A0** and **B0** will be done.

13.36.1 See also

Overview, A0, B0, B00, B1, B11.

13.36.2 Examples

By default B_0 is not expressed in terms of A_0.

```
B0[0, s, s]
```

$$\mathrm{B}_0(0, s, s)$$

With **BReduce -> True**, transformation is done.

```
B0[0, s, s, BReduce -> True]
```

$$-\frac{(2-D)\ \mathrm{A}_0(s)}{2s}$$

13.37 CartesianIndexNames

CartesianIndexNames is an option for **FCFAConvert**, **FCCanonicalizeDummyIndices** and other functions. It renames the generic dummy Cartesian indices to the indices in the supplied list.

13.37.1 See also

Overview, FCFAConvert, FCCanonicalizeDummyIndices, CartesianIndexNames.

13.37.2 Examples

```
CLC[i1, i2, i3] CGA[i1, i2, i3]

FCCanonicalizeDummyIndices[%]
```

$$\overline{\gamma}^{\mathrm{i1}}.\overline{\gamma}^{\mathrm{i2}}.\overline{\gamma}^{\mathrm{i3}}\,\overline{\epsilon}^{\mathrm{i1\ i2\ i3}}$$

$$\overline{\gamma}^{\mathrm{FCGV(ci201)}}.\overline{\gamma}^{\mathrm{FCGV(ci202)}}.\overline{\gamma}^{\mathrm{FCGV(ci203)}}\,\overline{\epsilon}^{\mathrm{FCGV(ci201)FCGV(ci202)FCGV(ci203)}}$$

```
CLC[i1, i2, i3] CGA[i1, i2, i3]

FCCanonicalizeDummyIndices[%, CartesianIndexNames -> {i, j, k}]
```

$$\overline{\gamma}^{\mathrm{i1}}.\overline{\gamma}^{\mathrm{i2}}.\overline{\gamma}^{\mathrm{i3}}\,\overline{\epsilon}^{\mathrm{i1\ i2\ i3}}$$

$$\overline{\gamma}^{i}.\overline{\gamma}^{j}.\overline{\gamma}^{k}\,\overline{\epsilon}^{ijk}$$

13.38 ClearHeads

ClearHeads is an option of **FCLoopIsolate**, **FCDiracIsolate** and other functions. It takes a list of heads that will be replaced by **Identity** in the isolating function. This is useful for cases when we first apply the isolating function to an expression, then simplify the isolated expression and finally want to apply the isolating function again to pull out the simplified expressions out of the old heads.

13.38.1 See also

Overview, FCLoopIsolate, FCDiracIsolate.

13.38.2 Examples

13.39 Collecting

Collecting is an option of **ScalarProductCancel**, **Series2**, **TID** and related functions. Setting it to **True** will trigger some kind of collecting of the result.

13.39.1 See also

Overview, ScalarProductCancel, Series2, TID.

13.39.2 Examples

13.40 CombineGraphs

CombineGraphs is an option for **OneLoopSum**.

13.40.1 See also

Overview, OneLoopSum.

13.40.2 Examples

13.41 CounterT

CounterT is a factor used by **GluonPropagator** and **QuarkPropagator** when **CounterTerms** is set to **All**.

13.41.1 See also

Overview, CounterTerm, GluonPropagator, QuarkPropagator.

13.41.2 Examples

```
GluonPropagator[p, \[Mu], a, \[Nu], b, Explicit -> True, CounterTerm ->
All]
```

$$\text{CounterT}\left(\frac{iC_A g_s^2 S_n \delta^{ab}\left(\frac{10p^\mu p^\nu}{3}-\frac{10}{3}p^2 g^{\mu\nu}\right)}{\varepsilon}+\frac{iT_f\, g_s^2 S_n \delta^{ab}\left(\frac{4}{3}p^2 g^{\mu\nu}-\frac{4p^\mu p^\nu}{3}\right)}{\varepsilon}\right)-\frac{i\delta^{ab}g^{\mu\nu}}{p^2}$$

13.42 CouplingConstant

CouplingConstant is an option for several Feynman rule functions and for **CovariantD** and **FieldStrength**. In the convention of the add-on PHI, **CouplingConstant** is also the head of coupling constants. **CouplingConstant** takes three extra optional arguments, with heads **RenormalizationState**, **RenormalizationScheme** and **ExpansionState** respectively. E.g. **CouplingConstant[QED[1]]** is the unit charge, **CouplingConstant[ChPT2[4], 1]** is the first of the coupling constants of the Lagrangian **ChPT2[4]**.

CouplingConstant[a_, b_, c___][i_] := CouplingConstant[a, b, RenormalizationState[i], c].

13.42.1 See also

Overview, CovariantD, FieldStrength.

13.42.2 Examples

13.43 CustomIndexNames

CustomIndexNames is an option of **FCCanonicalizeDummyIndices**. It allows to specify custom names for canonicalized dummy indices of custom index heads.

13.43.1 See also

Overview, FCFAConvert, FCCanonicalizeDummyIndices, LorentzIndexNames, CartesianIndexNames.

13.43.2 Examples

```
ex = T1[MyIndex2[a], MyIndex1[b]] v1[MyIndex1[b]] v2[MyIndex2[a]] +
   T1[MyIndex2[c], MyIndex1[f]] v1[MyIndex1[f]] v2[MyIndex2[c]]
```

$$\begin{aligned}&\text{v2}(\text{MyIndex2}(a))\ \text{v1}(\text{MyIndex1}(b))\ \text{T1}(\text{MyIndex2}(a),\\&\ \ \text{MyIndex1}(b)) + \text{v2}(\text{MyIndex2}(c))\ \text{v1}(\text{MyIndex1}(f))\ \text{T1}(\text{MyIndex2}(c), \text{MyIndex1}(f))\end{aligned}$$

```
FCCanonicalizeDummyIndices[ex , Head -> {MyIndex1, MyIndex2},
  CustomIndexNames -> {{MyIndex1, {i1}}, {MyIndex2, {i2}}}]
```

$$2\ \text{v1}(\text{MyIndex1}(\text{i1}))\ \text{v2}(\text{MyIndex2}(\text{i2}))\ \text{T1}(\text{MyIndex2}(\text{i2}), \text{MyIndex1}(\text{i1}))$$

13.44 D0Convention

`D0Convention` is an option for **`Write2`**. If set to **`1`**, the convention for the arguments of **`D0`** is changed when writing a Fortran file with **`Write2`**: The fifth and sixth argument of **`D0`** are interchanged and the square root is taken of the last four arguments.

13.44.1 See also

Overview, D0, Write2.

13.44.2 Examples

13.45 DetectLoopTopologies

`DetectLoopTopologies` is an option for FDS. If set to True, FDS will try to recognize some special multiloop topologies and apply appropriate simplifications

13.45.1 See also

Overview, FDS.

13.45.2 Examples

13.46 Dimension

`Dimension` is an option of several functions and denotes the number of space-time dimensions. Possible settings are: **`4`**, **`n`**, **`d`**, **`D`**, ... , the variable does not matter, but it should have head Symbol.

13.46.1 See also

Overview, ScalarProduct.

13.46.2 Examples

```
Options[ScalarProduct]
```

$$\{\text{Dimension} \to 4, \text{FeynCalcInternal} \to \text{True}, \text{SetDimensions} \to \{4, D\}\}$$

```
ex = ScalarProduct[m, n, Dimension -> d]
```

$$m \cdot n$$

```
ex // StandardForm

(*Pair[Momentum[m, d], Momentum[n, d]]*)
```

13.47 DiracIndexNames

DiracIndexNames is an option for **FCFAConvert**, **FCCanonicalizeDummyIndices** and other functions. It renames the generic dummy Dirac indices to the indices in the supplied list.

13.47.1 See also

Overview, FCFAConvert, FCCanonicalizeDummyIndices, PauliIndexNames, LorentzIndexNames.

13.47.2 Examples

```
DCHN[GA[mu], i, j] DCHN[GA[nu], j, k]

FCCanonicalizeDummyIndices[%]
```

$$(\bar{\gamma}^{\text{mu}})_{ij} (\bar{\gamma}^{\text{nu}})_{jk}$$

$$(\bar{\gamma}^{\text{mu}})_{i\text{FCGV(di241)}} (\bar{\gamma}^{\text{nu}})_{\text{FCGV(di241)}k}$$

```
DCHN[GA[mu], i, j] DCHN[GA[nu], j, k]

FCCanonicalizeDummyIndices[%, DiracIndexNames -> {a}]
```

$$(\bar{\gamma}^{\text{mu}})_{ij}\,(\bar{\gamma}^{\text{nu}})_{jk}$$

$$(\bar{\gamma}^{\text{mu}})_{ia}\,(\bar{\gamma}^{\text{nu}})_{ak}$$

13.48 DiracSpinorNormalization

DiracSpinorNormalization is an option for **SpinorChainEvaluate**, **DiracSimplify** and other functions. It specifies the normalization of the spinor inner products $\bar{u}(p)u(p)$ and $\bar{v}(p)v(p)$. Following values are supported:

- **"Relativistic"** - this is the standard value corresponding to $\bar{u}(p)u(p) = 2m$, $\bar{v}(p)v(p) = -2m$.
- **"Rest"** - this sets $\bar{u}(p)u(p) = 1$, $\bar{v}(p)v(p) = -1$.
- **"Nonrelativistic"** - this sets $\bar{u}(p)u(p) = \frac{m}{p^0}$, $\bar{v}(p)v(p) = -\frac{m}{p^0}$.

13.48.1 See also

Overview, DiracSimplify, SpinorChainEvaluate.

13.48.2 Examples

```
SpinorUBar[p, m] . SpinorU[p, m]

DiracSimplify[%]
```

$$\bar{u}(p,m).u(p,m)$$

$$2m$$

```
SpinorUBar[p, m] . SpinorU[p, m]

DiracSimplify[%, DiracSpinorNormalization -> "Rest"]
```

$$\bar{u}(p,m).u(p,m)$$

$$1$$

```
SpinorUBar[p, m] . SpinorU[p, m]

DiracSimplify[%, DiracSpinorNormalization -> "Nonrelativistic"]
```

$$\bar{u}(p,m).u(p,m)$$

$$\frac{m}{p^0}$$

13.49 DiracTraceEvaluate

DiracTraceEvaluate is an option for **DiracTrace**, **DiracSimplify** and some other functions. If set to **False**, Dirac traces remain unevaluated. For more details, see the documentation for **DiracTrace** and **DiracSimplify**.

13.49.1 See also

Overview, DiracSimplify, DiracTrace

13.49.2 Examples

13.50 Divideout

Divideout is an option for **OPEInt** and **OPEIntegrate**. The setting is divided out at the end.

13.50.1 See also

Overview, OPEInt, OPEIntegrate.

13.50.2 Examples

13.51 DotPower

DotPower is an option for **DotSimplify**. It determines whether non-commutative powers are represented by successive multiplication or by **Power**.

13.51.1 See also

Overview, DotSimplify.

13.51.2 Examples

13.52 DotSimplifyRelations

DotSimplifyRelations is an option for **DotSimplify**. Its setting may be a list of substitution rules of the form **DotSimplifyRelations -> {a.b -> c, b^2 -> 0, ...}**.

In the rules, **Condition** should not be used and patterns should be avoided on the right-hand sides.

Notice that the performance of **DotSimplify** scales very badly with the complexity of **DotSimplifyRelations** and the number of terms of the expression.

13.52.1 See also

Overview, DotSimplify.

13.52.2 Examples

13.53 DropScaleless

DropScaleless is an option for **FCLoopIsolate**, **ApartFF**, **FourDivergence** and other functions. When set to **True**, all loop integrals that do not contain a **FeynAmpDenominator**, i.e. consist of only scalar products but no denominators, are set to zero.

13.53.1 See also

Overview, FCLoopIsolate.

13.53.2 Examples

```
FCLoopIsolate[SPD[l, q], {q}]
```

$$\text{FCGV}(\text{LoopInt})(l \cdot q)$$

```
FCLoopIsolate[SPD[l, q], {q}, DropScaleless -> True]
```

$$0$$

13.54 DropSumOver

DropSumOver is an option of **FCFAConvert**. When set to **True**, **SumOver** symbols in the FeynArts diagrams will be dropped. Those symbols are usually not needed in FeynCalc where Einstein summation always applies, but they might be kept for other purposes.

13.54.1 See also

Overview, FCFAConvert.

13.54.2 Examples

13.55 DummyIndex

DummyIndex is an option of **CovariantD** specifying an index to use as dummy summation index. If set to **Automatic**, unique indices are generated.

13.55.1 See also

Overview, CovariantD.

13.55.2 Examples

13.56 EpsDiscard

EpsDiscard is an option for **FeynCalc2FORM**. If set to **True** all Levi-Civita tensors are replaced by **0** after contraction.

13.56.1 See also

Overview, FeynCalc2FORM.

13.56.2 Examples

13.57 EpsExpand

EpsExpand is an option for **EpsEvaluate** and other functions that use **EpsEvaluate** internally. When set to **False**, sums of momenta in the **Eps** tensor will not be rewritten as a sum of **Eps** tensors.

13.57.1 See also

Overview, EpsEvaluate.

13.57.2 Examples

```
LC[mu, nu][q1 + q2, p1 + p2]

EpsEvaluate[%]
```

$$\bar{\epsilon}^{\text{mu nu } \overline{\text{q1}}+\overline{\text{q2}} \overline{\text{p1}}+\overline{\text{p2}}}$$

$$-\bar{\epsilon}^{\text{mu nu } \overline{\text{p1}} \overline{\text{q1}}} - \bar{\epsilon}^{\text{mu nu } \overline{\text{p1}} \overline{\text{q2}}} - \bar{\epsilon}^{\text{mu nu } \overline{\text{p2}} \overline{\text{q1}}} - \bar{\epsilon}^{\text{mu nu } \overline{\text{p2}} \overline{\text{q2}}}$$

```
LC[mu, nu][q1 + q2, p1 + p2]

EpsEvaluate[%, EpsExpand -> False]
```

$$\bar{\epsilon}^{\text{mu nu } \overline{\text{q1}}+\overline{\text{q2}} \overline{\text{p1}}+\overline{\text{p2}}}$$

$$-\bar{\epsilon}^{\text{mu nu } \overline{\text{p1}}+\overline{\text{p2}} \overline{\text{q1}}+\overline{\text{q2}}}$$

13.58 EpsilonOrder

`EpsilonOrder` is an option of **`OPEIntegrateDelta`** and other functions. The setting determines the order **`n`** (**`Epsilon^n`**) which should be kept.

13.58.1 See also

Overview, OPEIntegrateDelta.

13.58.2 Examples

13.59 EtaSign

`EtaSign` is an option for **`SFAD`**, **`GFAD`**, **`CFAD`** and other objects representing propagators. It specifies the default sign of the $i\eta$ prescription in the propagators, e.g. for standard Feynman propagators the value **`1`** corresponds to $\frac{1}{p^2-m^2+i\eta}$, while the value **`-1`** sets $\frac{1}{p^2-m^2-i\eta}$.

13.59.1 See also

Overview, SFAD, CFAD, GFAD.

13.59.2 Examples

Notice that if the sign of $i\eta$ is already specified in the propagator, e.g.

```
CFAD[{q, {m^2, 1}}]
```

$$\frac{1}{(q^2+m^2+i\eta)}$$

then this specification always overrides the EtaSign option. Hence

```
CFAD[{q, {m^2, 1}}, EtaSign -> -1]
```

$$\frac{1}{(q^2+m^2+i\eta)}$$

still has the positive $i\eta$.

13.60 ExceptHeads

ExceptHeads is an option of **FCLoopIsolate**, **FCDiracIsolate** and other functions. It takes a list of heads that are not allowed to appear inside isolated expression.

For example, **ExceptHeads -> {DiracGamma}** in **FCLoopIsolate** blocks loop integrals where loop momenta are contracted with **Dirac matrices**.

13.60.1 See also

Overview, FCLoopIsolate, FCDiracIsolate.

13.60.2 Examples

13.61 ExcludeMasses

ExcludeMasses is an option of **Apart2**. It allows to specify masses w.r.t which partial fraction decomposition should not be performed, e.g. **ExcludeMasses->{m1, m2, 3}**.

13.61.1 See also

Overview, Apart2.

13.61.2 Examples

```
Apart2[FAD[k, {k, m1}, {k, m2}]] // Expand
```

$$\frac{1}{\text{m1}^2\left(\text{m1}^2-\text{m2}^2\right)\left(k^2-\text{m1}^2\right)}-\frac{1}{\text{m1}^2\left(\text{m1}^2-\text{m2}^2\right)\left(k^2-\text{m2}^2\right)}-\frac{1}{\text{m1}^2\,\text{m2}^2\left(k^2-\text{m2}^2\right)}+\frac{1}{k^2\,\text{m1}^2\,\text{m2}^2}$$

```
Apart2[FAD[k, {k, m1}, {k, m2}], ExcludeMasses -> m2] // Expand
```

$$\frac{1}{\text{m1}^2\left(k^2-\text{m1}^2\right).\left(k^2-\text{m2}^2\right)}-\frac{1}{\text{m1}^2k^2.\left(k^2-\text{m2}^2\right)}$$

13.62 Expanding

Expanding is an option for **Calc**, **Contract**, **DiracSimplify**, **DotSimplify**, **SUNSimplify** etc. As option for **Contract** it specifies whether expansion w.r.t. **LorentzIndex** is done *before* contraction. If set to **False** in **DiracSimplify** or **SUNSimplify**, only a limited set of simplifications (multiplicative linearity etc.) is performed.

13.62.1 See also

Overview, Calc, Contract, DiracSimplify, DotSimplify, SUNSimplify.

13.62.2 Examples

13.63 ExtraFactor

ExtraFactor is an option for **FermionSpinSum**. The setting **ExtraFactor -> fa** multiplies the whole amplitude with the factor **fa** before squaring.

13.63.1 See also

Overview, FermionSpinSum.

13.63.2 Examples

13.64 ExtraPropagators

ExtraPropagators is an option for **FCLoopFindTopologies**. It can be used to specify extra propagators that do not explicitly appear in the input expression but must be taken into account when constructing the sets of propagators.

13.64.1 See also

Overview, FCLoopFindTopologies.

13.64.2 Examples

13.65 ExtraVariables

ExtraVariables is an option for **OneLoopSum**. It may be set to a list of variables which are also bracketed out in the result, just like **B0**, **C0**, **D0** and **PaVe**.

13.65.1 See also

Overview, OneLoop, OneLoopSum.

13.65.2 Examples

13.66 FactorFull

FactorFull is an option of **Factor2** (default **False**). If set to **False**, products like **(a-b) (a+b)** will be replaced by **a^2-b^2**.

13.66.1 See also

Overview, Factor2.

13.66.2 Examples

```
Factor2[(a - b) (a + b) (c + d) (c - d)]
```

$$(a^2 - b^2)\,(c^2 - d^2)$$

```
Factor2[(a - b) (a + b) (c + d) (c - d), FactorFull -> True]
```

$$(a - b)(a + b)(c - d)(c + d)$$

13.67 Factoring

Factoring is an option for **Collect2**, **Contract**, **Tr** and more functions. If set to **True**, the result will be factored, using **Factor2**. If set to any function **f**, this function will be used.

13.67.1 See also

Overview, Collect2, Contract, Tr.

13.67.2 Examples

13.68 FactoringDenominator

FactoringDenominator is an option for **Collect2**. It is taken into account only when the option **Numerator** is set to **True**. If **FactoringDenominator** is set to any function **f**, this function will be applied to the denominator of the fraction. The default value is **False**, i.e. the denominator will be left unchanged.

13.68.1 See also

Overview, Collect2.

13.68.2 Examples

```
ex = (x1 a^2 + y 1 a^2 + 2 a b + x2 b^2 + y2 b^2)/(a + b + c^2 +
       2 c d + d^2)
```

$$\frac{a^2\,\text{x1} + a^2 y + 2ab + b^2\,\text{x2} + b^2\,\text{y2}}{a + b + c^2 + 2cd + d^2}$$

```
Collect2[ex, a, b]
```

$$\frac{a^2(\text{x1} + y)}{a + b + c^2 + 2cd + d^2} + \frac{b^2(\text{x2} + \text{y2})}{a + b + c^2 + 2cd + d^2} + \frac{2ab}{a + b + c^2 + 2cd + d^2}$$

```
Collect2[ex, a, b, Numerator -> True]
```

$$\frac{a^2(\text{x1} + y) + 2ab + b^2(\text{x2} + \text{y2})}{a + b + c^2 + 2cd + d^2}$$

```
Collect2[ex, a, b, Numerator -> True, FactoringDenominator -> Simplify]
```

$$\frac{a^2(\text{x1} + y) + 2ab + b^2(\text{x2} + \text{y2})}{a + b + (c + d)^2}$$

13.69 Factorout

Factorout is an option for OPEInt and OPEIntegrate.

13.69.1 See also

Overview, OPEInt, OPEIntegrate.

13.69.2 Examples

13.70 FAModelsDirectory

FAModelsDirectory is an option of **FAPatch**. It points to the directory that contains FeynArts models. The default value is **"Models"** inside **$FeynArtsDirectory**.

13.70.1 See also

Overview, FAPatch.

13.70.2 Examples

13.71 FCDoControl

FCDoControl is an option for **FCPrint** that specifies which variable is used to control the debugging output of **FCPrint**. The default value is **$VeryVerbose**.

13.71.1 See also

Overview, FCPrint.

13.71.2 Examples

13.72 FCJoinDOTs

FCJoinDOTs is an option for **DotSimplify** and other functions that use **DotSimplify** internally. When set to **True**, **DotSimplify** will try to rewrite expressions like **A.X.B + A.Y.B** as **A.(X+Y).B**.

Notice that although the default value of **FCJoinDOTs** is **True**, the corresponding transformations will occur only if the option **Expanding** is set to **False** (default: **True**)

13.72.1 See also

Overview, DotSimplify.

13.72.2 Examples

```
DeclareNonCommutative[A, B, X, Y]
```

```
DotSimplify[A . X . B + A . Y . B]
```

$$A.X.B + A.Y.B$$

```
DotSimplify[A . X . B + A . Y . B, FCJoinDOTs -> True]
```

$$A.X.B + A.Y.B$$

```
DotSimplify[A . X . B + A . Y . B, FCJoinDOTs -> True, Expanding -> False]
```

$$A.(X + Y).B$$

```
DotSimplify[GA[mu, 6, nu] + GA[mu, 7, nu], FCJoinDOTs -> True, Expanding ->
False]
```

$$\bar{\gamma}^{\text{mu}}.\left(\bar{\gamma}^{6} + \bar{\gamma}^{7}\right).\bar{\gamma}^{\text{nu}}$$

13.73 FCVerbose

FCVerbose is an option for numerous functions that allows to specify a local value of **$VeryVerbose** inside those functions. When set to a positive integer, all the debugging information inside the function will be given according to the value of **FCVerbose**, while the debugging output of other functions will be still governed by the value of **$VeryVerbose**. Following values are common

- **1** - a brief description of the calculational steps including timings
- **2** - somewhat more debugging information
- **3** - lots of debugging output, probably useful only for developers

13.73.1 See also

Overview, $VeryVerbose.

13.73.2 Examples

```
DiracSimplify[GA[\[Mu], \[Nu], \[Rho], \[Mu], \[Nu]], FCVerbose -> 1]
```

DiracSimplify: Entering.

DiracSimplify: Normal mode.

DiracSimplify: Extracting Dirac objects.

DiracSimplify: Done extracting Dirac objects, timing: 0.03636

DiracSimplify: Doing index contractions.

DiracSimplify: Index contractions done, timing: 0.000453

DiracSimplify: Applying diracSimplifyEval

DiracSimplify: diracSimplifyEval done, timing: 0.01655

DiracSimplify: Inserting Dirac objects back into products.

DiracSimplify: Done inserting Dirac objects back into products, timing: 0.000132

DiracSimplify: Applying SpinorChainTrick.

DiracSimplify: Done applying SpinorChainTrick, timing: 0.001167

DiracSimplify: Creating the final replacement rule.

DiracSimplify: Final replacement rule done, timing: 0.000119

DiracSimplify: Expanding the result.

DiracSimplify: Expanding done, timing: 0.000413

DiracSimplify: Leaving.

DiracSimplify: Total timing: 0.06579

$$4\bar{\gamma}^{\rho}$$

```
DiracSimplify[GA[\[Mu], \[Nu], \[Rho], \[Mu], \[Nu]], FCVerbose -> 2]
```

DiracSimplify: Entering.

DiracSimplify: Normal mode.

DiracSimplify: Extracting Dirac objects.

DiracSimplify: Done extracting Dirac objects, timing: 0.008883

DiracSimplify: Doing index contractions.

DiracSimplify: Index contractions done, timing: 0.000426

DiracSimplify: Applying diracSimplifyEval

DiracSimplify: diracSimplifyEval: Entering

DiracSimplify: diracSimplifyEval: Applying DiracTrick.

DiracSimplify: diracSimplifyEval: DiracTrick done, timing: 0.003183

DiracSimplify: diracSimplifyEval: Applying Dotsimplify.

DiracSimplify: diracSimplifyEval: Dotsimplify done, timing: 0.002288

DiracSimplify: diracSimplifyEval: Applying DiracTrick.

DiracSimplify: diracSimplifyEval: DiracTrick done, timing: 0.006512

DiracSimplify: diracSimplifyEval: Applying Dotsimplify.

DiracSimplify: diracSimplifyEval: Dotsimplify done, timing: 0.002125

DiracSimplify: diracSimplifyEval done, timing: 0.01820

DiracSimplify: Inserting Dirac objects back into products.

DiracSimplify: Done inserting Dirac objects back into products, timing: 0.000146

DiracSimplify: Applying SpinorChainTrick.

DiracSimplify: Done applying SpinorChainTrick, timing: 0.001152

DiracSimplify: Creating the final replacement rule.

DiracSimplify: Final replacement rule done, timing: 0.000123

DiracSimplify: Expanding the result.

DiracSimplify: Expanding done, timing: 0.000374

DiracSimplify: Leaving.

DiracSimplify: Total timing: 0.03523

$$4\bar{\gamma}^{\rho}$$

```
DiracSimplify[GA[\[Mu], \[Nu], \[Rho], \[Mu], \[Nu]], FCVerbose -> 3]
```

DiracSimplify: Entering.

DiracSimplify: Entering with $\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}$

DiracSimplify: Normal mode.

DiracSimplify: Extracting Dirac objects.

DiracSimplify: dsPart: FeynCalc`DiracSimplify`Private`dsHeadAll (FeynCalc`DiracSimplify`Private`dsHead ($\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.$

DiracSimplify: freePart: 0

DiracSimplify: Done extracting Dirac objects, timing: 0.009456

DiracSimplify: diracObjects: {FeynCalc`DiracSimplify`Private`dsHead ($\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}$)}

DiracSimplify: Doing index contractions.

DiracSimplify: Index contractions done, timing: 0.000418

DiracSimplify: diracObjectsEval after index contractions: {FeynCalc`DiracSimplify`Private`dsHead ($\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}$)}

DiracSimplify: Applying diracSimplifyEval

DiracSimplify: diracSimplifyEval: Entering

DiracSimplify: diracSimplifyEval: Entering with: $\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}$

DiracSimplify: diracSimplifyEval: Applying DiracTrick.

DiracSimplify: diracSimplifyEval: DiracTrick done, timing: 0.002919

DiracSimplify: diracSimplifyEval: After DiracTrick: $4\bar{\gamma}^{\rho}$

DiracSimplify: diracSimplifyEval: Applying Dotsimplify.

DiracSimplify: diracSimplifyEval: Dotsimplify done, timing: 0.002082

DiracSimplify: diracSimplifyEval: After Dotsimplify: $4\bar{\gamma}^{\rho}$

DiracSimplify: diracSimplifyEval: Applying DiracTrick.

DiracSimplify: diracSimplifyEval: DiracTrick done, timing: 0.006242

DiracSimplify: diracSimplifyEval: After DiracTrick: $4\bar{\gamma}^{\rho}$

DiracSimplify: diracSimplifyEval: Applying Dotsimplify.

DiracSimplify: diracSimplifyEval: Dotsimplify done, timing: 0.002124

DiracSimplify: diracSimplifyEval: After Dotsimplify: $4\bar{\gamma}^{\rho}$

DiracSimplify: diracSimplifyEval: Leaving with: $4\bar{\gamma}^{\rho}$

DiracSimplify: After diracSimplifyEval: $\{4\bar{\gamma}^{\rho}\}$

DiracSimplify: diracSimplifyEval done, timing: 0.02049

DiracSimplify: Inserting Dirac objects back into products.

DiracSimplify: repRule: {FeynCalc`DiracSimplify`Private`dsHead $(\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}.\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}) \to 4\bar{\gamma}^{\rho}$}

DiracSimplify: Done inserting Dirac objects back into products, timing: 0.001005

DiracSimplify: Intermediate result: $\{4\bar{\gamma}^{\rho}\}$

DiracSimplify: Applying SpinorChainTrick.

DiracSimplify: Done applying SpinorChainTrick, timing: 0.001158

DiracSimplify: After SpinorChainTrick: $\{4\bar{\gamma}^{\rho}\}$

DiracSimplify: Creating the final replacement rule.

DiracSimplify: repRule: {FeynCalc`DiracSimplify`Private`dsHeadAll (FeynCalc`DiracSimplify`Private`dsHead ($\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}$

DiracSimplify: Final replacement rule done, timing: 0.000980

DiracSimplify: Expanding the result.

DiracSimplify: Expanding done, timing: 0.000399

DiracSimplify: After expanding: $4\bar{\gamma}^{\rho}$

DiracSimplify: Leaving.

DiracSimplify: Total timing: 0.04359

DiracSimplify: Leaving with $4\bar{\gamma}^{\rho}$

$4\bar{\gamma}^{\rho}$

13.74 FeynmanIntegralPrefactor

FeynmanIntegralPrefactor is an option for **FCFeynmanParametrize** and other functions. It denotes an implicit prefactor that has to be understood in front of a loop integral in the usual **FeynAmpDenominator**-notation. The prefactor is the quantity that multiplies the loop integral measure $d^D q_1 \ldots d^D q_n$ and plays an important role e.g. when deriving the Feynman parameter representation of the given integral. Apart from specifying an explicit value, the user may also choose from the following predefined conventions:

- "Unity" - 1 for each loop
- "Textbook" - $\frac{1}{(2\pi)^D}$ for each loop.
- "Multiloop1" - $\frac{1}{i\pi^{D/2}}$ for each loop if the integral is Minkowskian, $\frac{1}{i\pi^{D/2}}$ or $\frac{1}{i\pi^{(D-1)/2}}$ for each loop if the integral is Euclidean or Cartesian respectively.
- "Multiloop2" - like "Multiloop1" but with an extra $e^{\frac{(4-D)}{2}\gamma_E}$ for each loop

The standard value is "Multiloop1".

13.74.1 See also

Overview, FCFeynmanParametrize.

13.74.2 Examples

```
FCFeynmanParametrize[FAD[p, p - q], {p}, Names -> x, FCReplaceD -> {D -> 4
- 2 Epsilon}]
```

$$\left\{(x(1)+x(2))^{2\varepsilon-2}\left(-q^2x(1)x(2)\right)^{-\varepsilon},\Gamma(\varepsilon),\{x(1),x(2)\}\right\}$$

```
FCFeynmanParametrize[FAD[p, p - q], {p}, Names -> x, FCReplaceD -> {D -> 4
- 2 Epsilon}]

Times @@ Most[%]
```

$$\left\{(x(1)+x(2))^{2\varepsilon-2}\left(-q^2x(1)x(2)\right)^{-\varepsilon},\Gamma(\varepsilon),\{x(1),x(2)\}\right\}$$

$$\Gamma(\varepsilon)(x(1)+x(2))^{2\varepsilon-2}\left(-q^2x(1)x(2)\right)^{-\varepsilon}$$

```
FCFeynmanParametrize[FAD[p, p - q], {p}, Names -> x, FCReplaceD -> {D -> 4
- 2 Epsilon},
   FeynmanIntegralPrefactor -> "Multiloop1"]

Times @@ Most[%]
```

$$\left\{(x(1)+x(2))^{2\varepsilon-2}\left(-q^2x(1)x(2)\right)^{-\varepsilon},\Gamma(\varepsilon),\{x(1),x(2)\}\right\}$$

$$\Gamma(\varepsilon)(x(1)+x(2))^{2\varepsilon-2}\left(-q^2x(1)x(2)\right)^{-\varepsilon}$$

```
FCFeynmanParametrize[FAD[p, p - q], {p}, Names -> x, FCReplaceD -> {D -> 4
- 2 Epsilon},
   FeynmanIntegralPrefactor -> "Unity"]

Times @@ Most[%]
```

$$\left\{(x(1)+x(2))^{2\varepsilon-2}\left(-q^2x(1)x(2)\right)^{-\varepsilon},i\pi^{2-\varepsilon}\Gamma(\varepsilon),\{x(1),x(2)\}\right\}$$

$$i\pi^{2-\varepsilon}\Gamma(\varepsilon)(x(1)+x(2))^{2\varepsilon-2}\left(-q^2x(1)x(2)\right)^{-\varepsilon}$$

```
FCFeynmanParametrize[FAD[p, p - q], {p}, Names -> x, FCReplaceD -> {D -> 4
- 2 Epsilon},
   FeynmanIntegralPrefactor -> "Textbook"]

Times @@ Most[%]
```

$$\left\{(x(1)+x(2))^{2\varepsilon-2}\left(-q^2x(1)x(2)\right)^{-\varepsilon},i2^{2\varepsilon-4}\pi^{\varepsilon-2}\Gamma(\varepsilon),\{x(1),x(2)\}\right\}$$

$$i2^{2\varepsilon-4}\pi^{\varepsilon-2}\Gamma(\varepsilon)(x(1)+x(2))^{2\varepsilon-2}\left(-q^2x(1)x(2)\right)^{-\varepsilon}$$

```
FCFeynmanParametrize[FAD[p, p - q], {p}, Names -> x, FCReplaceD -> {D -> 4
- 2 Epsilon},
   FeynmanIntegralPrefactor -> "Multiloop2"]

Times @@ Most[%]
```

$$\left\{(x(1)+x(2))^{2\varepsilon-2}\left(-q^2x(1)x(2)\right)^{-\varepsilon},e^{\gamma\varepsilon}\Gamma(\varepsilon),\{x(1),x(2)\}\right\}$$

$$e^{\gamma\varepsilon}\Gamma(\varepsilon)(x(1)+x(2))^{2\varepsilon-2}\left(-q^2x(1)x(2)\right)^{-\varepsilon}$$

```
FCFeynmanParametrize[FAD[{p, m}], {p}, Names -> x, FCReplaceD -> {D -> 4 -
2 Epsilon},
   FeynmanIntegralPrefactor -> "Multiloop2"]

Times @@ Most[%]

Series[%, {Epsilon, 0, 1}] // Normal // FunctionExpand
```

$$\left\{1, -e^{\gamma\varepsilon}\Gamma(\varepsilon-1)\left(m^2\right)^{1-\varepsilon}, \{\}\right\}$$

$$-e^{\gamma\varepsilon}\Gamma(\varepsilon-1)\left(m^2\right)^{1-\varepsilon}$$

$$\frac{m^2}{\varepsilon}+\frac{1}{12}\varepsilon\left(\pi^2m^2+12m^2+6m^2\log^2\left(m^2\right)-12m^2\log\left(m^2\right)\right)+m^2+m^2\left(-\log\left(m^2\right)\right)$$

13.75 FinalFunction

FinalFunction is an option for **OneLoopSum**.

13.75.1 See also

Overview, OneLoopSum.

13.75.2 Examples

13.76 FinalSubstitutions

FinalSubstitutions is an option for **OneLoop** and **OneLoopSum** and **Write2**. All substitutions indicated hereby are done at the end of the calculation.

13.76.1 See also

Overview, OneLoop, OneLoopSum, Write2.

13.76.2 Examples

13.77 ForceSave

ForceSave is an option of **CheckDB**. Setting it to **True** forces the first argument to be evaluated even if the file specified by the second argument exists. The expression is also saved if **NoSave** is set to **False**. Default value: **False**.

13.77.1 See also

Overview, CheckDB.

13.77.2 Examples

13.78 FORMAbbreviations

FORMAbbreviations is an option for **FeynCalc2FORM**. It specifies how special symbols will be abbreviated in the resulting FORM file.

13.78.1 See also

Overview, FeynCalc2FORM.

13.78.2 Examples

13.79 FORMEpilog

FORMEpilog is an option for **FeynCalc2FORM**. It may be set to a string which is put at the end of the FORM-file.

13.79.1 See also

Overview, FeynCalc2FORM, FORMProlog.

13.79.2 Examples

13.80 FORMIdStatements

FORMIdStatements is an option for **FeynCalc2FORM**. It may be set to a string which is put after the local expression of the FORM-file. When set to **True**, FeynCalc will try to generate the statements automatically from the known values of scalar products.

13.80.1 See also

Overview, FeynCalc2FORM.

13.80.2 Examples

13.81 FORMProlog

FORMProlog is an option for **FeynCalc2FORM**. It may be set to a string which is put after the type declarations of the FORM-file.

13.81.1 See also

Overview, FeynCalc2FORM, FORMEpilog.

13.81.2 Examples

13.82 FortranFormatDoublePrecision

FortranFormatDoublePrecision is an option for **Write2**.

13.82.1 See also

Overview, Write2.

13.82.2 Examples

13.83 FunctionLimits

FunctionLimits is an option of **ILimit**, specifying which functions should be checked for finiteness.

13.83.1 See also

Overview, ILimit.

13.83.2 Examples

13.84 Gauge

Gauge is an option for **GluonPropagator**. If set to **1** the 't Hooft Feynman gauge is used.

13.84.1 See also

Overview, GluonPropagator.

13.84.2 Examples

13.85 IncomingMomenta

IncomingMomenta is an option of **FCFAConvert**. It specifies how the incoming momenta in the diagram should be named. The number and order of momenta in the list of momenta should exactly match those in **InsertFields** of FeynArts.

13.85.1 See also

Overview, IncomingMomenta.

13.85.2 Examples

13.86 IndexPosition

IndexPosition is an option for **FieldStrength**.

13.86.1 See also

Overview, FieldStrength.

13.86.2 Examples

13.87 InitialFunction

InitialFunction is an option of **FeynRule** the setting of which is applied to the first argument of FeynRule before anything else.

13.87.1 See also

Overview, FeynRule.

13.87.2 Examples

13.88 InitialSubstitutions

InitialSubstitutions is an option for **OneLoop** and **OneLoopSum** and **Write2**. All substitutions indicated hereby are done at the end of the calculation.

13.88.1 See also

Overview, OneLoop, OneLoopSum, Write2.

13.88.2 Examples

13.89 InsideDiracTrace

InsideDiracTrace is an option of **DiracSimplify** and some other functions dealing with Dirac algebra. If set to **True**, the function assumes to operate inside a Dirac trace, i.e., products of an odd number of Dirac matrices are discarded. For more details, see the documentation for **DiracSimplify**.

13.89.1 See also

Overview

13.89.2 Examples

```
DiracSimplify[GA[\[Mu], \[Nu], \[Rho]]]
```

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}.\bar{\gamma}^{\rho}$$

A trace of an 3 Dirac matrices vanishes:

```
DiracSimplify[GA[\[Mu], \[Nu], \[Rho]], InsideDiracTrace -> True]
```

$$0$$

13.90 InsidePauliTrace

InsidePauliTrace is an option of **PauliSimplify** and some other functions dealing with Pauli algebra. If set to **True**, the function assumes to operate inside a Pauli trace.

13.90.1 See also

Overview, PauliSimplify.

13.90.2 Examples

13.91 IntegralTable

IntegralTable is an option of **OneLoopSimplify**, **TwoLoopSimplify** and **FeynAmpDenominatorSimplify**. It may be set to a list of the form **{FCIntegral[...] :> bla, ...}**.

13.91.1 See also

Overview, OneLoopSimplify, TwoLoopSimplify, FeynAmpDenominatorSimplify.

13.91.2 Examples

13.92 IntermediateSubstitutions

IntermediateSubstitutions is an option of various FeynCalc functions. All substitutions indicated hereby are done at an intermediate stage of the calculation.

13.92.1 See also

Overview, OneLoop.

13.92.2 Examples

13.93 IsolateFast

IsolateFast is an option of **Isolate** and other functions using Isolate. When set to **True** and when **varlist** is empty, **Isolate** will not attempt to recognize existing abbreviations, but will immediately abbreviate the whole expression instead. This is useful for very large expressions or prefactors, where **Isolate** would otherwise require a lot of time to finish.

13.93.1 See also

Overview, Isolate, Collect2.

13.93.2 Examples

13.94 IsolateNames

IsolateNames is an option for **Isolate** and **Collect2**. Its default setting is **KK**. Instead of a symbol the setting may also be a list with the names of the **abbreviations**.

13.94.1 See also

Overview, Isolate, Collect2.

13.94.2 Examples

13.95 IsolatePlus

IsolatePlus is an option for **Isolate** and other functions using **Isolate**. If it is set to **True**, Isolate will split sums that contain elements from **vlist**, to be able to abbreviate the **vlist**-free part.

13.95.1 See also

Overview, Isolate.

13.95.2 Examples

```
Isolate[a + b + c + d, a]
```

$$a + b + c + d$$

```
Isolate[a + b + c + d, a, IsolatePlus -> True]
```

$$a + \mathrm{KK}(19)$$

13.96 IsolatePrint

IsolatePrint is an option of **Isolate**. If it is set to **OutputForm** (or any other ***Form**) the definitions of the abbreviations are printed during the operation of **Isolate**.

The setting **IsolatePrint -> False** suppresses printing.

13.96.1 See also

Overview, Isolate.

13.96.2 Examples

13.97 IsolateSplit

IsolateSplit is an option for **Isolate**. Its setting determines the maximum number of characters of **FortranForm[expr]** which are abbreviated by **Isolate**. If the expression is larger than the indicated number, it is split into smaller pieces and onto each subsum **Isolate** is applied.

With the default setting **IsolateSplit -> Infinity** no splitting is done.

13.97.1 See also

Overview, Isolate.

13.97.2 Examples

13.98 IsolateTimes

IsolateTimes is an option for **Isolate** and other functions using **Isolate**. If it is set to **True**, Isolate will be applied also to pure products.

13.98.1 See also

Overview, Isolate, Collect2.

13.98.2 Examples

By default, this expression does not become abbreviated

```
Isolate[a*b*c*d, a]
```

$$abcd$$

Now an abbreviation is introduced

```
Isolate[a*b*c*d, a, IsolateTimes -> True]
```

$$a \text{ KK}(19)$$

13.99 LarinMVV

LarinMVV is an option for **DiracTrace** and several other functions that deal with traces of Dirac matrices. The option applies only to the computation of D-dimensional traces with a single γ^5 in the Larin scheme. With **LarinMVV -> True** (default setting), such traces are computed according to Eq. (10) from arXiv:1506.04517 by S. Moch, J. A. M. Vermaseren and A. Vogt.

13.99.1 See also

Overview, DiracTrace.

13.99.2 Examples

13.100 LightPak

LightPak is an option for **FCLoopPakOrder** and other functions for finding equivalent topologies or integrals using Pak algorithm. When set to True, instead of using the full Pak algorithm (which can be slow for complicated integrals) we use only a lightweight version that is not guaranteed to find all mappings but requires significantly less time.

The light Pak algorithm is described in the pySecDec manual. Essentially, it means that in the step 5 of the full Pak algorithm we keep only the first matrix in the vector, so that the next iteration step generates significantly less matrices than in the full version.

13.100.1 See also

Overview, FCLoopPakOrder.

13.100.2 Examples

Canonicalizing this characteristic Polynomial of a loop integral with 11 propagators requires almost half a minute using the full Pak algorithm. The light Pak finishes here almost immediately.

```
poly = (x[1]*x[2]*x[3]*x[4]*x[5]*x[6]*x[7]*x[8]*(x[9]*x[10] + x[9]*x[11] +
       x[10]*x[11]) - x[1]*x[2]*x[3]*x[4]*x[5]*x[6]*x[7]*x[8]*(m1^2*x[1] +
       m1^2*x[2] +
       m1^2*x[3] + m1^2*x[4] + m1^2*x[5] + m1^2*x[6] + m1^2*x[7] +
m1^2*x[8] + m1^2*x[9] +
       m1^2*x[10] + m3^2*x[11])*(x[9]*x[10] + x[9]*x[11] + x[10]*x[11]))
```

$$x(1)x(2)x(3)x(4)x(5)x(6)x(7)x(8)(x(9)x(10)+x(11)x(10)+x(9)x(11))$$
$$-x(1)x(2)x(3)x(4)x(5)x(6)x(7)x(8)(x(9)x(10)+x(11)x(10)+x(9)x(11))\left(\mathrm{m1}^2x(1)+\mathrm{m1}^2x(2)\right.$$
$$\left.+\mathrm{m1}^2x(3)+\mathrm{m1}^2x(4)+\mathrm{m1}^2x(5)+\mathrm{m1}^2x(6)+\mathrm{m1}^2x(7)+\mathrm{m1}^2x(8)+\mathrm{m1}^2x(9)+\mathrm{m1}^2x(10)+\mathrm{m3}^2x(11)\right)$$

```
FCLoopPakOrder[poly, {x[1], x[2], x[3], x[4], x[5], x[6], x[7], x[8], x[9],
x[10], x[11]},
  LightPak -> True]
```

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{pmatrix}$$

```
canoPoly1 = FCLoopPakOrder[poly, {x[1], x[2], x[3], x[4], x[5], x[6], x[7],
x[8], x[9], x[10],x[11]},
    LightPak -> True, Rename -> True];
```

```
canoPoly2 = FCLoopPakOrder[poly /. {x[3] -> x[11], x[11] -> x[3]},
    {x[1], x[2], x[3], x[4], x[5], x[6], x[7], x[8], x[9], x[10], x[11]},
    LightPak -> True, Rename -> True];
```

Obviously, we can still find equivalence relations with using this algorithm

```
canoPoly1 - canoPoly2
```

$$0$$

13.101 Loop

Loop is an option for functions related to FeynArts integration, indicating the number of (virtual) loops.

13.101.1 See also

Overview

13.101.2 Examples

13.102 LoopMomenta

LoopMomenta is an option of **FCFAConvert**. It specifies how the loop momenta in the diagram should be named. The number and order of momenta in the list of momenta should exactly match those in **InsertFields** of FeynArts.

13.102.1 See also

Overview, FCFAConvert.

13.102.2 Examples

13.103 LorentzIndexNames

LorentzIndexNames is an option for **FCFAConvert**, **FCCanonicalizeDummyIndices** and other functions. It renames the generic dummy Lorentz indices to the indices in the supplied list.

13.103.1 See also

Overview, FCFAConvert, FCCanonicalizeDummyIndices, CartesianIndexNames.

13.103.2 Examples

```
LC[i1, i2, i3, i4] GA[i1, i2, i3, i4]

FCCanonicalizeDummyIndices[%]
```

$$\bar{\gamma}^{\text{i1}}.\bar{\gamma}^{\text{i2}}.\bar{\gamma}^{\text{i3}}.\bar{\gamma}^{\text{i4}}\bar{\epsilon}^{\text{i1 i2 i3 i4}}$$

$$\bar{\gamma}^{\text{FCGV(li191)}}.\bar{\gamma}^{\text{FCGV(li192)}}.\bar{\gamma}^{\text{FCGV(li193)}}.\bar{\gamma}^{\text{FCGV(li194)}}\bar{\epsilon}^{\text{FCGV(li191)FCGV(li192)FCGV(li193)FCGV(li194)}}$$

```
LC[i1, i2, i3, i4] GA[i1, i2, i3, i4]

FCCanonicalizeDummyIndices[%, LorentzIndexNames -> {mu, nu, rho, si}]
```

$$\bar{\gamma}^{\text{i1}}.\bar{\gamma}^{\text{i2}}.\bar{\gamma}^{\text{i3}}.\bar{\gamma}^{\text{i4}}\bar{\epsilon}^{\text{i1 i2 i3 i4}}$$

$$\bar{\gamma}^{\text{mu}}.\bar{\gamma}^{\text{nu}}.\bar{\gamma}^{\text{rho}}.\bar{\gamma}^{\text{si}}\bar{\epsilon}^{\text{mu nu rho si}}$$

13.104 Mandelstam

Mandelstam is an option for **DiracTrace**, **OneLoop**, **OneLoopSum**, **Tr** and **TrickMandelstam**. A typical setting is **Mandelstam -> {s, t, u, m1^2+m2^2+m3^2+m4^2}**, which implies $s+t+u = m_1^2+m_2^2+m_3^2+m_4^2$. If other than four-particle processes are calculated, the setting should be **Mandelstam -> {}**.

13.104.1 See also

Overview, DiracTrace, OneLoop, OneLoopSum, Tr, TrickMandelstam, SetMandelstam.

13.104.2 Examples

13.105 MultiLoop

MultiLoop is an option for **FCLoopIsolate**. When set to **True**, **FCLoopIsolate** will isolate only such loop integrals, that depend on all of the given loop momenta. Integrals that depend only on some of the loop momenta will be treated as non-loop terms and remain non-isolated.

13.105.1 See also

Overview, FCLoopIsolate.

13.105.2 Examples

13.106 NoSave

NoSave is an option of **CheckDB**. If set to **True**, no results will ever be saved to disk. It is there to allow evaluating notebooks using **CheckDB** without having to worry about overwriting old results (**SetOptions[CheckDB, NoSave->True]**). Default value: **False**.

13.106.1 See also

Overview, CheckDB.

13.106.2 Examples

13.107 NumberOfPolarizations

NumberOfPolarizations is an option for **DoPolarizationSums**. It specifies the number of polarizations to sum over in the expression. This is relevant only for expressions that contain terms free of polarization vectors. This may occur e.g. if the scalar products involving polarization vectors have already been assigned some particular values. In this case the corresponding terms will be multiplied by the corresponding number of polarizations.

The default value is **Automatic** which means that the function will attempt to recognize the correct value automatically by extracting the dimension **dim** of the polarization vectors and putting **(dim-2)** for massless and **(dim-1)** for massive vector bosons. Notice that if the input expression is free of polarization vectors, the setting **Automatic** will fail, and the user must specify the correct dimension by hand.

13.107.1 See also

Overview, DoPolarizationSums.

13.107.2 Examples

```
PolarizationVector[p, mu] ComplexConjugate[PolarizationVector[p, mu]]
```

$$\bar{\varepsilon}^{*\mathrm{mu}}(p)\bar{\varepsilon}^{\mathrm{mu}}(p)$$

Here the setting Automatic is sufficient.

```
FCClearScalarProducts[];

ScalarProduct[p, p] = 0;

PolarizationVector[p, mu] ComplexConjugate[PolarizationVector[p, mu]] + xyz


DoPolarizationSums[%, p, n]
```

$$\bar{\varepsilon}^{*\mathrm{mu}}(p)\bar{\varepsilon}^{\mathrm{mu}}(p) + \mathrm{xyz}$$

DoPolarizationSums: The input expression contains terms free of polarization vectors. Those will be multiplied wit

$$2\ \mathrm{xyz} - 2$$

Here it is not

```
DoPolarizationSums[xyz, p, n]
```

DoPolarizationSums: The option NumberOfPolarizations is set to Automatic but since the input expression contains no polarization vectors, the function cannot determine the number of polarizations to sum over in the expression automatically. Please set NumberOfPolarizations to the appropriate value e.g. 2 or D- 2 etc. by hand.

$Aborted

Setting the number of polarizations by hand fixes the issue

```
DoPolarizationSums[xyz, p, n, NumberOfPolarizations -> 2]
```

DoPolarizationSums: The input expression contains terms free of polarization vectors. Those will be multiplied wit

$$2\ \mathrm{xyz}$$

13.108 NotMomentum

NotMomentum is an option of **FCCanonicalizeDummyIndices**. It specifies a list of momenta for which no canonicalization should be done.

13.108.1 See also

Overview, FCCanonicalizeDummyIndices.

13.108.2 Examples

13.109 OtherLoopMomenta

OtherLoopMomenta is an option of **ToPaVe**. It takes a list of loop momenta other than **q** that appear in the expression. Knowing about these momenta prevents **ToPaVe** from erroneously converting multi-loop integrals into **PaVe** scalar functions.

This is of course relevant only for multi-loop calculations. At 1-loop you don't need to specify this option explicitly.

13.109.1 See also

Overview, ToPaVe.

13.109.2 Examples

13.110 OutgoingMomenta

OutgoingMomenta is an option of **FCFAConvert**. It specifies how the outgoing momenta in the diagram should be named. The number and order of momenta in the list of momenta should exactly match those in **InsertFields** of FeynArts.

13.110.1 See also

Overview, FCFAConvert.

13.110.2 Examples

13.111 PairCollect

PairCollect is an option for **DiracTrace** specifying if the result is collected with respect to **Pair**s.

13.111.1 See also

Overview, DiracTrace, Pair.

13.111.2 Examples

13.112 PartialDRelations

PartialDRelations is an option for **ExpandPartialD**. It is a list of rules applied by **ExpandPartialD** at the end.

13.112.1 See also

Overview, FCPartialD, ExpandPartialD, LeftPartialD, LeftRightPartialD, RightPartialD.

13.112.2 Examples

```
QuantumField[A, {\[Mu]}] . QuantumField[B, {\[Mu]}] . LeftPartialD[\[Nu]]

ExpandPartialD[%, PartialDRelations -> {A -> C}]
```

$$A_\mu.B_\mu.\overleftarrow{\partial}_\nu$$

$$C_\mu.\left(\left(\partial_\nu B_\mu\right)\right)+\left(\left(\partial_\nu C_\mu\right)\right).B_\mu$$

13.113 PatchModelsOnly

PatchModelsOnly is an option of **FAPatch**. When set to **True**, **FAPatch** will patch only the model files in the **"Models"** directory of the FeynArts installation. This is useful when you have added new custom models, e.g. generated by FeynRules.

13.113.1 See also

Overview, FAPatch.

13.113.2 Examples

13.114 PauliIndexNames

PauliIndexNames is an option for **FCFAConvert**, **FCCanonicalizeDummyIndices** and other functions. It renames the generic dummy Pauli indices to the indices in the supplied list.

13.114.1 See also

Overview, FCFAConvert, FCCanonicalizeDummyIndices, DiracIndexNames, LorentzIndexNames.

13.114.2 Examples

```
PCHN[CSI[a], i, j] PCHN[CSI[b], j, k]

FCCanonicalizeDummyIndices[%]
```

$$\left(\overline{\sigma}^{a}\right)_{ij}\left(\overline{\sigma}^{b}\right)_{jk}$$

$$\left(\overline{\sigma}^{a}\right)_{i\text{FCGV(pi251)}}\left(\overline{\sigma}^{b}\right)_{\text{FCGV(pi251)}k}$$

```
PCHN[CSI[a], i, j] PCHN[CSI[b], j, k]

FCCanonicalizeDummyIndices[%, PauliIndexNames -> {l}]
```

$$\left(\overline{\sigma}^{a}\right)_{ij}\left(\overline{\sigma}^{b}\right)_{jk}$$

$$\left(\overline{\sigma}^{a}\right)_{il}\left(\overline{\sigma}^{b}\right)_{lk}$$

13.115 PauliReduce

PauliReduce is an option for **PauliTrick** and other functions. It specifies whether a chain of Pauli matrices should be reduced to at most one matrix by rewriting every pair of matrices in terms of commutator and anticommutator.

13.115.1 See also

Overview, PauliTrick, PauliSimplify.

13.115.2 Examples

```
CSI[i, j, k]
```

```
PauliSimplify[%]
```

$$\overline{\sigma}^i.\overline{\sigma}^j.\overline{\sigma}^k$$

$$\overline{\sigma}^i.\overline{\sigma}^j.\overline{\sigma}^k$$

```
CSI[i, j, k]
```

```
PauliSimplify[%, PauliReduce -> True]
```

$$\overline{\sigma}^i.\overline{\sigma}^j.\overline{\sigma}^k$$

$$\overline{\sigma}^i \bar{\delta}^{jk} - \overline{\sigma}^j \bar{\delta}^{ik} + \overline{\sigma}^k \bar{\delta}^{ij} + i\bar{\epsilon}^{ijk}$$

13.116 PauliTraceEvaluate

PauliTraceEvaluate is an option for **PauliTrace**, **PauliSimplify** and some other functions. If set to **False**, Pauli traces remain unevaluated.

13.116.1 See also

Overview, PauliTrace, PauliSimplify.

13.116.2 Examples

13.117 PaVeAutoOrder

PaVeAutoOrder is an option of **PaVe** and other functions that work with **PaVe** functions. When set to **True**, the arguments of the PaVe functions will be automatically ordered by using known symmetries between those arguments.

13.117.1 See also

Overview, PaVe, PaVeReduce.

13.117.2 Examples

13.118 PaVeAutoReduce

PaVeAutoReduce is an option of **PaVe** and other functions that work with **PaVe** functions. When set to **True**, for some special cases **PaVe** functions will be automatically reduced to simpler expressions. Otherwise, PaVe functions will not be further simplified unless explicitly evaluated by **PaVeReduce**.

13.118.1 See also

Overview, PaVe, PaVeReduce.

13.118.2 Examples

13.119 PaVeIntegralHeads

PaVeIntegralHeads is an option for **FCLoopIsolate**, **FCLoopSplit** and other functions. It gives a list of heads that denote Passarino-Veltman integrals

13.119.1 See also

Overview, FCLoopIsolate, FCLoopSplit.

13.119.2 Examples

13.120 PaVeOrderList

PaVeOrderList is an option for **PaVeOrder** and **PaVeReduce**, specifying in which order the arguments of **D0** are to be permuted.

13.120.1 See also

Overview, PaVeOrder, PaVeReduce, PaVe, D0.

13.120.2 Examples

13.121 PostFortranFile

PostFortranFile is an option for **Write2** which may be set to a file name (or a list of file names) or a string, which will be put at the end of the generated Fortran file.

13.121.1 See also

Overview, Write2, PreFortranFile.

13.121.2 Examples

13.122 Prefactor

Prefactor is an option for **OneLoop** and **OneLoopSum**. If set as option of **OneLoop**, the amplitude is multiplied by **Prefactor** before calculation; if specified as option of **OneLoopSum**, after calculation in the final result as a global factor.

13.122.1 See also

Overview, OneLoop, OneLoopSum.

13.122.2 Examples

PreferredIntegrals is an option for **FCLoopFindIntegralMappings** and other related functions. It allows to specify a list of GLIs onto which the occurring loop integrals should be preferably mapped.

13.122.3 See also

Overview, FCLoopFindIntegralMappings.

13.122.4 Examples

PreferredTopologies is an option for **FCLoopFindTopologies**, **FCLoopFindTopologyMappings** and other related functions. It allows to specify a list of topologies onto which the occurring topologies should be preferably mapped.

13.122.5 See also

Overview, FCLoopFindTopologies, FCLoopFindTopologyMappings.

13.122.6 Examples

13.123 PreFortranFile

PreFortranFile is an option for **Write2** which may be set to a file name (or a list of file names) or a string, which will be put at the beginning of the generated Fortran file.

13.123.1 See also

Overview, Write2, PostFortranFile.

13.123.2 Examples

13.124 PreservePropagatorStructures

PreservePropagatorStructures is an option for **DotSimplify**. If set to **True**, numerators of fermionic propagators like **(GS[p]+m)** that appear in chains of Dirac matrices will not be expanded.

13.124.1 See also

Overview, DotSimplify.

13.124.2 Examples

13.125 QuarkMass

QuarkMass is an option of **Amplitude** and **CounterTerm**.

13.125.1 See also

Overview, Amplitude, CounterTerm.

13.125.2 Examples

13.126 ReduceGamma

ReduceGamma is an option of **OneLoop**. If set to True all **GA[6]** and **GA[7]** (i.e. all chirality projectors) are reduced to **GA[5]**.

13.126.1 See also

Overview, OneLoop.

13.126.2 Examples

13.127 ReduceToScalars

ReduceToScalars is an option for **OneLoop** and **OneLoopSum** that specifies whether the result will be reduced to scalar **A0**, **B0**, **C0** and **D0** scalar integrals.

13.127.1 See also

Overview, OneLoop, OneLoopSum.

13.127.2 Examples

13.128 Rename

Rename is an option for **Contract**. If set to **True**, dummy indices in **Eps** are renamed, using **$MU[i]**.

13.128.1 See also

Overview, Contract.

13.128.2 Examples

```
LC[\[Mu], \[Nu], \[Rho], \[Sigma]] LC[\[Alpha], \[Nu], \[Rho], \[Sigma]]

Contract[%, EpsContract -> False, Rename -> True]
```

$$\bar{\epsilon}^{\alpha\nu\rho\sigma}\bar{\epsilon}^{\mu\nu\rho\sigma}$$

$$\bar{\epsilon}^{\alpha\ \$\mathrm{MU}(4)\$\mathrm{MU}(5)\$\mathrm{MU}(6)}\bar{\epsilon}^{\mu\ \$\mathrm{MU}(4)\$\mathrm{MU}(5)\$\mathrm{MU}(6)}$$

13.129 SameSideExternalEdges

SameSideExternalEdges is an option for **FCGraphCuttableQ**, **FCGraphFindPath** and other functions. It specifies external edges located on the same side of the graph so that a path that starts and ends on edges from this set does not prevent us from cutting the integral.

13.129.1 See also

Overview, FCGraphCuttableQ, FCGraphFindPath.

13.129.2 Examples

13.130 SchoutenAllowNegativeGain

SchoutenAllowZeroGain is an option for **FCSchoutenBruteForce** and other functions that attempt to simplify the input expression by applying Schouten's identity. It is similar to **SchoutenAllowZeroGain** with the difference that even transformations that increase the total number of terms might be applied in an attempt to arrive at a shorter expressions at a later stage.

13.130.1 See also

Overview, FCSchoutenBruteForce.

13.130.2 Examples

13.131 SchoutenAllowZeroGain

SchoutenAllowZeroGain is an option for **FCSchoutenBruteForce** and other functions that attempt to simplify the input expression by applying Schouten's identity. When set to **True**, the algorithm would apply Schouten's identity to the given expression even if this does not decrease the total number of terms in the expression. This is sometimes useful when the algorithm gets stuck and cannot find further transformation that would make the expression shorter.

13.131.1 See also

Overview, FCSchoutenBruteForce.

13.131.2 Examples

13.132 SelectGraphs

SelectGraphs is an option for **OneLoopSum** indicating that only a slected set of graphs of the list provided to **OneLoopSum** is to be calculated. Possible settings are: **SelectGraphs -> {i, j, ...}** or **SelectGraphs -> {a, {b, c}, ...}** which indicates the graphs to be taken from the list provided to **OneLoopSum**. In the second setting the list **{b, c}** indicates that all amplitudes from **b** to **c** should be taken.

13.132.1 See also

Overview, OneLoopSum.

13.132.2 Examples

13.133 SetDimensions

SetDimensions is an option for **ScalarProduct**, **CartesianScalarProduct** and various **FCLoopBasis*** functions.

For scalar products it specifies the dimensions for which the scalar products will be set when **ScalarProduct** or **CartesianScalarProduct** are used with the equality sign, e.g. in **ScalarProduct[a, b] = m^2**. By default, the scalar products are set for 4 and D dimensions. By changing this option the user can add other dimensions or remove the existing ones.

In case of the **FCLoopBasis*** functions this option specifies the dimensions of the loop and external momenta to be taken into account when extracting the propagator basis.

13.133.1 See also

Overview, ScalarProduct, CartesianScalarProduct.

13.133.2 Examples

13.134 SmallVariables

SmallVariables is an option for OneLoop. **SmallVariables->{Melectron}** i.e. will substitute **SmallVariable[Melectron]** for all **Melectron**'s in the calculation.

13.134.1 See also

Overview, OneLoop.

13.134.2 Examples

13.135 SplitSymbolicPowers

SplitSymbolicPowers is an option for **FCFeynmanParametrize** and other functions. When set to **True**, propagator powers containing symbols will be split into a nonnegative integer piece and the remaining piece. This leads to a somewhat different form of the resulting parametric integral, although the final result remains the same. The default value is **False**.

13.135.1 See also

Overview, FCFeynmanParametrize.

13.135.2 Examples

```
SFAD[{p, m^2, r + 2}]
```

$$(p^2 - m^2 + i\eta)^{-r-2}$$

```
v1 = FCFeynmanParametrize[SFAD[{p, m^2, r - 1}], {p}, Names -> x,
FCReplaceD -> {D -> 4 - 2 Epsilon}]
```

$$\left\{1, \frac{m^6(-1)^{r-1}\left(m^2\right)^{-\varepsilon-r}\Gamma(\varepsilon+r-3)}{\Gamma(r-1)}, \{\}\right\}$$

```
v2 = FCFeynmanParametrize[SFAD[{p, m^2, r - 1}], {p}, Names -> x,
FCReplaceD -> {D -> 4 - 2 Epsilon},
   SplitSymbolicPowers -> True]
```

$$\left\{1, \frac{m^6(-1)^{r-1}(r-1)\left(m^2\right)^{-\varepsilon-r}\Gamma(\varepsilon+r-3)}{\Gamma(r)}, \{\}\right\}$$

Both parametrizations lead to the same results (as expected)

```
Series[v1[[2]], {Epsilon, 0, 1}] // Normal

Series[v2[[2]], {Epsilon, 0, 1}] // Normal

% - %% // Simplify // FunctionExpand
```

$$\frac{\varepsilon m^6(-1)^r\left(m^2\right)^{-r}\Gamma(r-3)\left(\log\left(m^2\right)-\psi^{(0)}(r-3)\right)}{\Gamma(r-1)} - \frac{m^6(-1)^r\left(m^2\right)^{-r}\Gamma(r-3)}{\Gamma(r-1)}$$

$$\frac{\varepsilon m^6(-1)^r(r-1)\left(m^2\right)^{-r}\Gamma(r-3)\left(\log\left(m^2\right)-\psi^{(0)}(r-3)\right)}{\Gamma(r)} - \frac{m^6(-1)^r(r-1)\left(m^2\right)^{-r}\Gamma(r-3)}{\Gamma(r)}$$

$$0$$

13.136 SubLoop

SubLoop is an option for **OPE1Loop**. If set to **True**, sub 1-loop tensor integral decomposition is performed.

13.136.1 See also

Overview, OPE1Loop.

13.136.2 Examples

SubtopologyMarker is an option for **FCLoopFindTopologies**, **FCLoopFindTopologyMappings** and other topology related functions. It denotes the symbol that is used to specify that the given topology is a subtopology of another topology and has been obtained by removing some of the original propagators (i.e. there are no momenta shifts involved)

This information must be put into the very last list of the FCTopology object describing the corresponding subtopology. The syntax is **marker->topoID** where **topoID** is the ID of the larger topology.

Setting **SubtopologyMarker** to **False** means that the information about subtopologies will not be added when generating subtopologies and will be ignored by routines related to topology mappings.

13.136.3 See also

Overview, FCLoopFindTopologies, FCLoopFindTopologyMappings, FCLoopFindSubtopologies.

13.136.4 Examples

13.137 SUNFJacobi

SUNFJacobi is an option for **SUNSimplify**, indicating whether the Jacobi identity should be used.

13.137.1 See also

Overview, SUNF, SUNSimplify.

13.137.2 Examples

```
SUNF[a, b, c] SUNF[e, f, c] // SUNSimplify[#, SUNFJacobi -> False] &
```

$$f^{cef}\, f^{abc}$$

```
SUNF[a, b, c] SUNF[e, f, c] // SUNSimplify[#, SUNFJacobi -> True] &
```

$$f^{bcf}\, f^{ace} - f^{acf}\, f^{bce}$$

13.138 SUNIndexNames

SUNIndexNames is an option for **FCFAConvert**, **FCCanonicalizeDummyIndices** and other functions. It renames the generic dummy $SU(N)$ indices in the adjoint representation to the indices in the supplied list.

13.138.1 See also

Overview, FCFAConvert, FCCanonicalizeDummyIndices.

13.138.2 Examples

13.139 SUNFIndexNames

SUNFIndexNames is an option for **FCFAConvert**, **FCCanonicalizeDummyIndices** and other functions. It renames the generic dummy $SU(N)$ indices in the fundamental representation to the indices in the supplied list.

13.139.1 See also

Overview, FCFAConvert, FCCanonicalizeDummyIndices.

13.139.2 Examples

13.140 SUNTraceEvaluate

SUNTraceEvaluate is an option for **SUNTrace**, **SUNSimplify** and some other functions. If set to **False**, color traces remain unevaluated. **Automatic** implies evaluation of traces with 2 or 3 color matrices. Setting this option to **True** will force every trace to be evaluated. For more details, see the documentation for **SUNTrace** and **SUNSimplify**.

13.140.1 See also

Overview, SUNSimplify, SUNTrace

13.140.2 Examples

13.141 SUNNToCACF

SUNNToCACF is an option of **SUNSimplify** and **CalcColorFactor**. If set to **True**, the Casimir operator eigenvalues **CA** ($= n_c$) and **CF** ($= (n_c^2 - 1)/(2n_c)$) are introduced.

13.141.1 See also

Overview, CalcColorFactor, SUNSimplify, Trick, SUNN, CA, CF.

13.141.2 Examples

```
SUNSimplify[SUNDelta[SUNIndex[a], SUNIndex[a]], SUNNToCACF -> True]
```

$$2C_A C_F$$

13.142 TraceDimension

TraceDimension is an option for **FeynCalc2FORM**. If set to **4** then **trace** is used, if set to **n** then **tracen** is employed.

13.142.1 See also

Overview, FeynCalc2FORM.

13.142.2 Examples

13.143 TraceOfOne

TraceOfOne is an option for **Tr** and **DiracTrace**. Its setting determines the value of the unit trace.

13.143.1 See also

Overview, DiracSimplify, DiracTrace.

13.143.2 Examples

```
DiracTrace[1]
```

```
DiracSimplify[%]
```

$$\text{tr}(1)$$

$$4$$

```
DiracTrace[1, TraceOfOne -> tr1]

DiracSimplify[%]
```

$$\mathrm{tr}(1)$$

$$\mathrm{tr1}$$

13.144 Transversality

Transversality is an option for **Polarization** and **PolarizationVector**. Setting it to **True** will make all scalar products of a polarization vector with its momentum vanish.

13.144.1 See also

Overview, Polarization, PolarizationVector.

13.144.2 Examples

13.145 TransversePolarizationVectors

TransversePolarizationVectors is an option of **FCFAConvert**. It specifies which polarization vectors should be defined as transverse. A particle is specified by its 4-momentum.

13.145.1 See also

Overview, FCFAConvert.

13.145.2 Examples

13.146 UndoChiralSplittings

UndoChiralSplittings is an option of **FCPrepareFAAmp**. When set to **True**, it attempts to undo splittings of couplings into left and right handed pieces, e.g. $aP_L\gamma^\mu + aP_R\gamma^\mu$ will be replaced by $a\gamma^\mu$.

13.146.1 See also

Overview, FCPrepareFAAmp.

13.146.2 Examples

13.147 UsePaVeBasis

UsePaVeBasis is an option of **TID**. When set to True, tensor reduction is always performed in terms of the Passarino-Veltman coefficient functions (e.g. B_1, B_{11}, C_{001} etc.) even if those can be reduced to the scalar functions A_0, B_0, C_0, D_0. By default this is done automatically only for tensor integrals with vanishing Gram determinants.

This option may be useful, if you are doing computations where the kinematics may later lead to vanishing Gram determinants or if you plan to evaluate all the Passarino-Veltman coefficient functions numerically (e.g. with LoopTools or Collier)

13.147.1 See also

Overview, TID.

13.147.2 Examples

13.148 UseTIDL

UseTIDL is an option of **Tdec**. When set to **True**, **Tdec** will check if the integral you want to decompose is already stored in **TIDL**, the built-in Tensor Integral Decomposition Library. If yes, then the result will be fetched immediately.

13.148.1 See also

Overview, Tdec.

13.148.2 Examples

13.149 UseWriteString

UseWriteString is an option for **FCPrint**. If set to **True**, the expression is printed via **WriteString** instead of **Print**.

13.149.1 See also

Overview, FCPrint.

13.149.2 Examples

13.150 VirtualBoson

VirtualBoson is an option for **PolarizationSum** and **DoPolarizationSums**. If set to True, FeynCalc will not complain when you apply the gauge trick (i.e. replace the polarization sum by $-g^{\mu\nu}$ for a particle that is not massless. This is useful when computing processes that involve a virtual photon as an external state.

13.150.1 See also

Overview, PolarizationSum, DoPolarizationSums.

13.150.2 Examples

13.151 West

West is an option for **DiracTrace** and several other functions that deal with traces of Dirac matrices. The option applies only to the computation of D-dimensional traces with an odd number of γ^5 in the Breitenlohner-Maison-t'Hooft-Veltman (BMHV) scheme. With **West -> True** (default setting), such traces are computed according to formula (A.5) from Comp. Phys. Comm 77 (1993) 286-298, which is also known as West's formula. For more details, see the documentation of DiracTrace.

13.151.1 See also

Overview, DiracTrace.

13.151.2 Examples

13.152 WriteOut

WriteOut is an option for **OneLoop**. If set to **True**, the result of **OneLoop** will be written to a file called "name.res", where name is the first argument of **OneLoop**.

13.152.1 See also

Overview, OneLoop.

13.152.2 Examples

13.153 WriteOutPaVe

WriteOutPaVe is an option for **PaVeReduce** and **OneLoopSum**. If set to a string, the results of all Passarino-Veltman **PaVe**'s are stored in files with names generated from this string and the arguments of **PaVe**.

13.153.1 See also

Overview, PaVeReduce, PaVe, OneLoopSum.

13.153.2 Examples

13.154 WriteStringOutput

UseWriteStringOutput an option for **FCPrint**. It specifies, to which stream **WriteString** should output the expression.

13.154.1 See also

Overview, FCPrint.

13.154.2 Examples

13.155 ZeroMomentumInsertion

ZeroMomentumInsertion is an option of FeynRule, Twist2GluonOperator and Twist2QuarkOperator.

13.155.1 See also

Overview, FeynRule, Twist2GluonOperator, Twist2QuarkOperator.

13.155.2 Examples

14 Misc

14.1 CalculateCounterTerm

`CalculateCounterTerm[exp, k]` calculates the residue of **`exp`**.This is a rather special function designed for some specific OPE calculations. Not a universal routine for daily use.

14.1.1 See also

Overview, OPE2TID.

14.1.2 Examples

14.2 GO

`GO` is equivalent to **`Twist2GluonOperator`**.

14.2.1 See also

Overview, Twist2GluonOperator.

14.2.2 Examples

`Integratedx[x, low, up]` is a variable representing the integration operator **`Integrate[#, {x, low, up}]&`**.

14.2.3 See also

Overview, CalculateCounterTerm, TimedIntegrate, HypergeometricIR, HypInt, TimedIntegrate.

14.2.4 Examples

14.3 $MIntegrate

`$MIntegrate` is a list of integrations done by Mathematica inside **`OPEIntegrateDelta`**.

14.3.1 See also

Overview, OPEIntegrateDelta.

14.3.2 Examples

```
$MIntegrate
```

{}

14.4 $OPEWard

`$OPEWard` is experimental.

14.4.1 See also

Overview

14.4.2 Examples

```
$OPEWard
```

False

14.5 OPE

`OPE` is a convenience variable to separate **`OPE`** insertions.

`OPE` is also an option of several input functions like **`GluonPropagator`**.

14.5.1 See also

Overview, OPE1Loop.

14.5.2 Examples

14.6 OPE1Loop

`OPE1Loop[q1, amp]` and **`OPE1Loop[{q1, q2}, amp]`** do sub-loop decomposition.

14.6.1 See also

Overview, OPESum.

14.6.2 Examples

14.7 OPE2TID

`OPE2TID[exp, k1, k2, p]` does a special tensor integral decomposition of exp.

14.7.1 See also

Overview, OPESum.

14.7.2 Examples

14.8 OPEDelta

`OPEDelta` is a lightlike axial vector as used e.g. in the operator product expansion in QCD.

14.8.1 See also

Overview, Twist2QuarkOperator.

14.8.2 Examples

```
FV[OPEDelta, \[Mu]]
```

```
Contract[% %]
```

$$\Delta^{\mu}$$

$$0$$

```
SP[OPEDelta, OPEDelta]
```

$$0$$

14.9 OPEi

`OPEi` etc. are variables with **`DataType PositiveInteger`** which are used in functions relating to the operator product expansion.

14.9.1 See also

Overview, OPEj, OPEk, OPEl, OPEn, OPEo.

14.9.2 Examples

```
OPEi
```

$$i$$

```
DataType[OPEi, OPEj, OPEk, OPEl, OPEm, OPEn, OPEo, PositiveInteger]
```

$$\{\text{True}, \text{True}, \text{True}, \text{True}, \text{True}, \text{True}, \text{True}\}$$

```
PowerSimplify[{(-1)^(2 OPEi), (-1)^(2 OPEj), (-1)^(2 OPEk), (-1)^(2 OPEl),
(-1)^(2 OPEm), (-1)^(2 OPEn), (-1)^(2 OPEo)}]
```

$$\{1, 1, 1, 1, 1, 1, 1\}$$

Re has been changed:

```
{Re[OPEi] > -3, Re[OPEi] > -2, Re[OPEi] > -1,   Re[OPEi] > 0, Re[OPEi] > 1}
```

$$\{\Re(i) > -3, \Re(i) > -2, \Re(i) > -1, \Re(i) > 0, \Re(i) > 1\}$$

```
{Re[-OPEi + OPEm] > 0, Re[-OPEi + OPEm] > 1, Re[-OPEi + OPEm] > 2}
```

$$\{\Re(m - i) > 0, \Re(m - i) > 1, \Re(m - i) > 2\}$$

```
{Re[OPEm] > -3, Re[OPEm] > -2, Re[OPEm] > -1,   Re[OPEm] > 0, Re[OPEm] > 1}
```

$$\{\Re(m) > -3, \Re(m) > -2, \Re(m) > -1, \Re(m) > 0, \Re(m) > 1\}$$

14.10 OPEInt

`OPEInt[expr, q, p, x]` calculates 1-loop OPE-type self energies.

14.10.1 See also

Overview.

14.10.2 Examples

14.11 OPEIntegrate

`OPEIntegrate[expr, q, x]` calculates a one-loop OPE-type integral.

14.11.1 See also

Overview.

14.11.2 Examples

14.12 OPEIntegrate2

OPEIntegrate2[exp, k] does special loop (tensorial) integrations. Only the residue is calculated.

14.12.1 See also

Overview, OPEIntegrate.

14.12.2 Examples

14.13 OPEIntegrateDelta

`OPEIntegrateDelta[expr, x, m]` introduces the $\delta(1-x)$ (**`DeltaFunction[1-x]`**).

The Mathematica **`Integrate`** function is called and each integration (from 0 to 1) is recorded for reference (and bug-checking) in the list **`$MIntegrate`**.

Notice that the dimension specified by the option should also be the dimension used in **`expr`**. It is replaced in OPEIntegrateDelta by (**`4+Epsilon`**).

14.13.1 See also

Overview.

14.13.2 Examples

14.14 OPEj

`OPEj` is a dummy index in OPESum.

14.14.1 See also

Overview, OPESum.

14.14.2 Examples

14.15 OPEk

`OPEk` is a dummy index in OPESum.

14.15.1 See also

Overview, OPESum.

14.15.2 Examples

14.16 OPEl

`OPEl` is a dummy index in OPESum.

14.16.1 See also

Overview, OPESum.

14.16.2 Examples

14.17 OPEm

`OPEm` is a dummy index in **`OPESum`**.

14.17.1 See also

Overview, OPESum.

14.17.2 Examples

14.18 OPEn

`OPEn` is a dummy index in **`OPESum`**.

14.18.1 See also

Overview, OPESum.

14.18.2 Examples

14.19 OPEo

`OPEo` is a dummy index in OPESum.

14.19.1 See also

Overview, OPESum.

14.19.2 Examples

14.20 OPESum

`OPESum[exp, {i, 0, m}]` denotes a symbolic sum. The syntax is the same as for **`Sum`**.

14.20.1 See also

Overview, OPESumExplicit, OPESumSimplify.

14.20.2 Examples

```
OPESum[SO[p]^OPEiSO[k]^(OPEm - OPEi - 3), {OPEi, 0, OPEm - 3}]

OPESumExplicit[%]
```

$$\sum_{i=0}^{-3+m} (\Delta \cdot p)^{\text{OPEiSO}(k)^{-3-i+m}}$$

$$\sum_{i=0}^{-3+m} (\Delta \cdot p)^{\text{OPEiSO}(k)^{-3-i+m}}$$

```
OPESum[a^ib^(j - i) c^(m - j - 4), {i, 0, j}, {j, 0, m - 4}]

OPESumExplicit[%]
```

$$\sum_{j=0}^{-4+m} (j+1) c^{-j+m-4} a^{\text{ib}^{j-i}}$$

$$\sum_{j=0}^{-4+m} (j+1) c^{-j+m-4} a^{\text{ib}^{j-i}}$$

14.21 OPESumExplicit

`OPESumExplicit[exp]` calculates **`OPESum`**s.

14.21.1 See also

Overview, OPESum, OPESumSimplify.

14.21.2 Examples

```
OPESum[A^iB^(m - i - 3), {i, 0, m - 3}]

OPESumExplicit[%]
```

$$\sum_{i=0}^{-3+m} A^{\text{iB}^{-3-i+m}}$$

$$\sum_{i=0}^{-3+m} A^{\text{iB}^{-3-i+m}}$$

```
OPESum[a^ib^(j - i) c^(m - j - 4), {i, 0, j}, {j, 0, m - 4}]

OPESumExplicit[%]
```

$$\sum_{j=0}^{-4+m} (j+1)c^{-j+m-4}a^{\text{ib}^{j-i}}$$

$$\sum_{j=0}^{-4+m} (j+1)c^{-j+m-4}a^{\text{ib}^{j-i}}$$

14.22 OPESumSimplify

`OPESumSimplify[exp]` simplifies **`OPESum`**s in **`exp`**.

14.22.1 See also

Overview, OPESum, OPESumExplicit.

14.22.2 Examples

```
OPESum[(-SOD[p])^(OPEi + 1) SOD[p - q]^(OPEm - OPEi - 2), {OPEi, 0, OPEm}]
```

$$\sum_{i=0}^{m}(-(\Delta\cdot p))^{1+i}(\Delta\cdot(p-q))^{-2-i+m}$$

```
OPESumSimplify[%]
```

$$(\Delta\cdot p)\left(-\sum_{i=0}^{m}(-1)^i(\Delta\cdot p)^i(\Delta\cdot(p-q))^{-2-i+m}\right)$$

```
OPESumSimplify[OPESum[{OPEi, 0, OPEm}] a^OPEi]
```

$$\sum_{i=0}^{m} a^i$$

```
OPESumSimplify[OPESum[{j, 0, i}, {i, 0, m}] a^(j - i) b^i]
```

$$\sum_{i=0}^{m} (i+1)b^i a^{j-i}$$

```
% // StandardForm

(*OPESum[a^(-i + j) b^i, {i, 0, m}, {j, 0, i}]*)
```

14.23 QO

QO is equivalent to **Twist2QuarkOperator**.

14.23.1 See also

Overview, Twist2QuarkOperator.

14.23.2 Examples

14.24 SO

SO[q] is a four-dimensional scalar product of **OPEDelta** with **q**. It is transformed into **Pair[Momentum[q], Momentum[OPEDelta]** by **FCI**.

14.24.1 See also

Overview, FCI, OPEDelta, Pair, ScalarProduct, SOD.

14.24.2 Examples

```
SO[p]
```

$$\Delta \cdot p$$

```
SO[p - q]
```

$$\Delta \cdot (p - q)$$

```
SO[p] // FCI // StandardForm

(*Pair[Momentum[OPEDelta], Momentum[p]]*)
```

14.25 SOD

SOD[q] is a D-dimensional scalar product of **OPEDelta** with **q**. It is transformed into **Pair[Momentum[q, D], Momentum[OPEDelta, D]** by **FeynCalcInternal**.

14.25.1 See also

Overview, OPEDelta, Pair, ScalarProduct, SOD.

14.25.2 Examples

```
SOD[p]
```

$$\Delta \cdot p$$

```
SOD[p - q]
```

$$\Delta \cdot (p - q)$$

```
SOD[p] // FCI // StandardForm

(*Pair[Momentum[OPEDelta, D], Momentum[p, D]]*)
```

14.26 SymbolicSum2

SymbolicSum2 is similar to **SymbolicSum** (**SymbolicSum** was a Mathematica function to do symbolic summation. It was obsolete from version 3 - all functionality is now autoloaded by Sum), but extended to several double sums.

14.26.1 See also

Overview, SymbolicSum3.

14.26.2 Examples

14.27 SymbolicSum3

SymbolicSum3 is similar to **SymbolicSum** (**SymbolicSum** was a Mathematica function to do symbolic summation. It was obsolete from version 3 - all functionality is now autoloaded by **Sum**), but extended to several double sums.

14.27.1 See also

Overview, SymbolicSum2.

14.27.2 Examples

14.28 Twist2AlienOperator

Twist2AlienOperator[p, 0] : (7); Twist2AlienOperator[p1,p2,{p3,mu,a}, 0] (p1: incoming quark momentum, p3: incoming gluon (count1)).

14.28.1 See also

Overview, Twist2GluonOperator.

14.28.2 Examples

```
Twist2AlienOperator[p, 0]

Twist2AlienOperator[p1, p2, {p3, mu, a}, 0]
```

$$\frac{2\left(\frac{2}{m}-\frac{1}{m+1}-\frac{2}{m-1}\right)C_F g_s^2 S_n \gamma\cdot\Delta(\Delta\cdot p)^{m-1}}{\varepsilon}$$

$$\frac{i\left((-1)^m+1\right)g_s^3\Delta^{\text{mu}}S_n T^a.(\gamma\cdot\Delta)\left(\left(\frac{2}{m}-\frac{1}{m+1}-\frac{2}{m-1}\right)\left(\frac{(\Delta\cdot \text{p1})^{m-1}}{\Delta\cdot \text{p1}+\Delta\cdot \text{p3}}-\frac{(-(\Delta\cdot \text{p3}))^{m-1}}{\Delta\cdot \text{p1}+\Delta\cdot \text{p3}}\right)+\left(\frac{1}{m-1}-\frac{1}{m}\right)(-(\Delta\cdot\ \text{p3}))^{m-2}\right.}{\varepsilon}$$

14.29 Twist2CounterOperator

`Twist2CounterOperator[p, mu, nu, a, b, 5]` is a special routine for particular QCD calculations.

Also available: **`Twist2CounterOperator[p, 7]`**, **`Twist2CounterOperator[p1, p2, {p3, mu, a}, 1]`** (**`p1`**: incoming quark momentum, **`p3`**: incoming gluon (count1)).

14.29.1 See also

Overview, Twist2GluonOperator.

14.29.2 Examples

```
Twist2CounterOperator[p, mu, nu, a, b, 5]
```

$$-\frac{1}{2\varepsilon}\left((-1)^m+1\right)C_A g_s^2 S_n \delta^{ab}\left(\left(\frac{8}{m}-\frac{12}{m+1}+\frac{8}{m+2}-\frac{8}{m-1}\right)g^{\text{mu nu}}(\Delta\cdot p)^m\right.$$
$$+\left(\frac{8}{m}-\frac{24}{m+1}+\frac{24}{m+2}-\frac{4}{m-1}\right)p^2\Delta^{\text{mu}}\Delta^{\text{nu}}(\Delta\cdot p)^{m-2}$$
$$\left.+\left(-\frac{6}{m}+\frac{16}{m+1}-\frac{16}{m+2}+\frac{6}{m-1}\right)(\Delta\cdot p)^{m-1}\left(p^{\text{mu}}\Delta^{\text{nu}}+\Delta^{\text{mu}}p^{\text{nu}}\right)\right)$$

```
Twist2CounterOperator[p, 7]
```

$$\frac{\left((-1)^m+1\right)\left(\frac{2}{m}-\frac{1}{m+1}-\frac{2}{m-1}\right)C_F g_s^2 S_n \gamma\cdot\Delta(\Delta\cdot p)^{m-1}}{\varepsilon}$$

```
Twist2CounterOperator[p1, p2, {p3, mu, a}, 1]
```

$$-\frac{\left((-1)^m+1\right)\left(\frac{2}{m}-\frac{1}{m+1}-\frac{2}{m-1}\right)T^a g_s^3\Delta^{\text{mu}}S_n\left(C_A-2C_F\right)\gamma\cdot\Delta\left(\frac{(\Delta\cdot \text{p1})^{m-1}}{\Delta\cdot \text{p1}+\Delta\cdot \text{p2}}-\frac{(-(\Delta\cdot \text{p2}))^{m-1}}{\Delta\cdot \text{p1}+\Delta\cdot \text{p2}}\right)}{2\varepsilon}$$

14.30 Twist2GluonOperator

`Twist2GluonOperator[{p, mu, a}, {nu, b}]` or **`Twist2GluonOperator[p, {mu, a}, {nu, b}]`** or **`Twist2GluonOperator[p, mu, a, nu, b]`** yields the 2-gluon operator (**`p`** is ingoing momentum corresponding to Lorentz index **`mu`**).

`Twist2GluonOperator[{p, mu, a}, {q, nu, b}, {k, la, c}]` or **`Twist2GluonOperator[p, mu, a , q, nu, b , k, la, c]`** gives the 3-gluon operator.

`Twist2GluonOperator[{p, mu, a}, {q, nu, b}, {k, la, c}, {s, si, d}]` or **`Twist2GluonOpera`** **`p, mu, a , q, nu, b , k, la, c , s, si, d]`** yields the 4-Gluon operator.

The dimension is determined by the option **`Dimension`**. The setting of the option **`Polarization`** (unpolarized: **`0`**; polarized: **`1`**) determines whether the unpolarized or polarized Feynman rule is returned.

With the setting **`Explicit`** set to **`False`** the color-structure and the (**`1+(-1)^OPEm`**) (for polarized: **`(1-(-1)^OPEm)`**) is extracted and the color indices are omitted in the arguments of **`Twist2GluonOperator`**.

14.30.1 See also

Overview, Twist2QuarkOperator.

14.30.2 Examples

The setting All for Explicit performs the sums.

```
Twist2GluonOperator[{p, \[Mu], a}, {q, \[Nu], b}, {r, \[Rho], c},
Polarization -> 1, Explicit -> All]
```

$$(1-(-1)^m)\, g_s f^{abc} \left(O^{\mathrm{G3}}_{\nu\rho\mu}(q,r,p)\right)$$

14.31 Twist2QuarkOperator

Twist2QuarkOperator[p] or Twist2QuarkOperator[p,,] yields the quark-antiquark operator (p is momentum in the direction of the incoming quark). Twist2QuarkOperator[{p,q}] yields the 2-quark operator for non-zero momentum insertion (p is momentum in the direction of the incoming quark). Twist2QuarkOperator[{p1,}, {p2,}, {p3, mu, a}] or Twist2QuarkOperator[p1,,, p2,,, p3,mu,a] is the quark-antiquark-gluon operator, where p1 is the incoming quark, p2 the incoming antiquark and p3 denotes the incoming gluon momentum. Twist2QuarkOperator[{p1}, {p2}, {p3, mu, a}, {p4, nu, b}] or Twist2QuarkOperator[{p1,}, {p2,}, {p3, mu, a}, {p4, nu, b}] or Twist2QuarkOperator[p1,,, p2,,, p3,mu,a, p4, nu, b] gives the Quark-Quark-Gluon-Gluon-operator. The setting of the option Polarization (unpolarized: 0; polarized: 1) determines whether the unpolarized or polarized operator is returned.

14.31.1 See also

Overview, Twist2GluonOperator.

14.31.2 Examples

14.32 Twist3QuarkOperator

Twist3QuarkOperator[p] or **Twist3QuarkOperator[p, _, _]** yields the 2-quark operator (**p** is momentum in the direction of the fermion number flow).

Twist3QuarkOperator[{p1, ___}, {p2, ___}, {p3, mu, a}] or **Twist3QuarkOperator[p1, _, _, p2, _, _, p3, mu, a]** yields the Quark-Quark-Gluon-operator, where **p1** is the incoming quark, **p2** the incoming antiquark and **p3** denotes the (incoming) gluon momentum.

Twist3QuarkOperator[{p1, ___}, {p2, ___}, {p3, mu, a}, {p4, nu, b}] or **Twist3QuarkOpera p1, _, _, p2, _, _, p3, mu, a, p4, nu, b]** gives the Quark-Quark-Gluon-Gluon-operator. The setting of the option **Polarization** (unpolarized: **0**; polarized: **1**) determines whether the unpolarized or polarized operator is returned.

14.32.1 See also

Overview, Twist2QuarkOperator, Twist2GluonOperator.

14.32.2 Examples

```
Twist3QuarkOperator[p]
```

$$(-1)^m(\gamma\cdot\Delta).\bar{\gamma}^5(\Delta\cdot p)^{m-1}$$

14.33 Twist4GluonOperator

`Twist4GluonOperator[{oa, ob, oc, od}, {p1, la1, a1}, {p2, la2, a2}, {p3, la3, a3}, {p4, la4, a4}]` is a special routine for particular QCD calculations.

14.33.1 See also

Overview, Twist2QuarkOperator, Twist3QuarkOperator, Twist2GluonOperator.

14.33.2 Examples

```
res = Twist4GluonOperator[{oa, ob, oc, od}, {p1, la1, a1}, {p2, la2, a2},
    {p3, la3, a3}, {p4, la4, a4}];
```

This is how the first two terms look like

```
res[[1 ;; 2]]
```

$$\begin{aligned}
&\delta^{\text{a1 od}}\delta^{\text{a2 oc}}\delta^{\text{a3 ob}}\delta^{\text{a4 oa}}\left(-\bar{g}^{\text{la1 la2}}\left(\Delta\cdot\overline{\text{p1}}\right)\left(\Delta\cdot\overline{\text{p2}}\right)+\Delta^{\text{la2}}\overline{\text{p2}}^{\text{la1}}\left(\Delta\cdot\overline{\text{p1}}\right)+\Delta^{\text{la1}}\overline{\text{p1}}^{\text{la2}}\left(\Delta\cdot\overline{\text{p2}}\right)\right.\\
&\left.-\Delta^{\text{la1}}\Delta^{\text{la2}}\left(\overline{\text{p1}}\cdot\overline{\text{p2}}\right)\right)\left(-\bar{g}^{\text{la3 la4}}\left(\Delta\cdot\overline{\text{p3}}\right)\left(\Delta\cdot\overline{\text{p4}}\right)+\Delta^{\text{la4}}\overline{\text{p4}}^{\text{la3}}\left(\Delta\cdot\overline{\text{p3}}\right)+\Delta^{\text{la3}}\overline{\text{p3}}^{\text{la4}}\left(\Delta\cdot\overline{\text{p4}}\right)\right.\\
&\left.-\Delta^{\text{la3}}\Delta^{\text{la4}}\left(\overline{\text{p3}}\cdot\overline{\text{p4}}\right)\right)+\delta^{\text{a1 oc}}\delta^{\text{a2 od}}\delta^{\text{a3 ob}}\delta^{\text{a4 oa}}\left(-\bar{g}^{\text{la1 la2}}\left(\Delta\cdot\overline{\text{p1}}\right)\left(\Delta\cdot\overline{\text{p2}}\right)+\Delta^{\text{la2}}\overline{\text{p2}}^{\text{la1}}\left(\Delta\cdot\overline{\text{p1}}\right)\right.\\
&\left.+\Delta^{\text{la1}}\overline{\text{p1}}^{\text{la2}}\left(\Delta\cdot\overline{\text{p2}}\right)-\Delta^{\text{la1}}\Delta^{\text{la2}}\left(\overline{\text{p1}}\cdot\overline{\text{p2}}\right)\right)\left(-\bar{g}^{\text{la3 la4}}\left(\Delta\cdot\overline{\text{p3}}\right)\left(\Delta\cdot\overline{\text{p4}}\right)\right.\\
&\left.+\Delta^{\text{la4}}\overline{\text{p4}}^{\text{la3}}\left(\Delta\cdot\overline{\text{p3}}\right)+\Delta^{\text{la3}}\overline{\text{p3}}^{\text{la4}}\left(\Delta\cdot\overline{\text{p4}}\right)-\Delta^{\text{la3}}\Delta^{\text{la4}}\left(\overline{\text{p3}}\cdot\overline{\text{p4}}\right)\right)
\end{aligned}$$

14.34 TwoLoopSimplify

`TwoLoopSimplify[amplitude, {qu1, qu2}]` simplifies the 2-loop amplitude (**`qu1`** and **`qu2`** denote the integration momenta).

`TwoLoopSimplify[amplitude]` transforms to **`TwoLoopSimplify[amplitude, {q1, q2}]`**, i.e., the integration momenta in amplitude must be named **`q1`** and **`q2`**.

14.34.1 See also

Overview, OneLoopSimplify.

14.34.2 Examples

15 Deprecated or legacy functions

15.1 AlphaStrong

AlphaStrong is a shortcut for **`SMP["alpha_s"]`** which represents the strong coupling constant.

The shortcut **`AlphaStrong`** is deprecated, please use **`SMP["alpha_s"]`** instead!

15.1.1 See also

Overview, SMP.

15.1.2 Examples

```
AlphaStrong
```

```
% // InputForm
```

$$\alpha_s$$

```
SMP["alpha_s"]
```

```
SMP["alpha_s"]
```

$$\alpha_s$$

15.2 AlphaFS

AlphaFS is a shortcut for **`SMP["alpha_fs"]`** which represents the fine-structure constant.

The shortcut **`AlphaFS`** is deprecated, please use **`SMP["alpha_fs"]`** instead!

15.2.1 See also

Overview, SMP.

15.2.2 Examples

```
AlphaFS

% // InputForm
```

$$\alpha$$

```
SMP["alpha_fs"]
```

```
SMP["alpha_fs"]
```

$$\alpha$$

15.3 $BreitMaison

$BreitMaison is a legacy switch for the Breitenlohner-Maison-t'Hooft-Veltman scheme.

Use **FCSetDiracGammaScheme** to specify a scheme for handling Dirac matrices in dimensional regularization and **FCGetDiracGammaScheme** to check the current setting.

15.3.1 See also

Overview, FCSetDiracGammaScheme, FCGetDiracGammaScheme.

15.3.2 Examples

15.4 $Larin

$Larin is a legacy switch for the Larin-Gorishny-Atkyampo-DelBurgo scheme. The modern way is to use **FCSetDiracGammaScheme** to specify a scheme for handling Dirac matrices in dimensional regularization and **FCGetDiracGammaScheme** to check the current setting.

15.4.1 See also

Overview, FCSetDiracGammaScheme, FCGetDiracGammaScheme.

15.4.2 Examples

15.5 ChiralityProjector

ChiralityProjector[+1] denotes $1/2\left(1+\gamma^5\right)$.

ChiralityProjector[-1] denotes $1/2\left(1+\gamma^5\right)$.

The shortcut **ChiralityProjector** is deprecated, please use **GA[6]** and **GA[7]** instead!

15.5.1 See also

Overview, GA, FCI.

15.5.2 Examples

```
{ChiralityProjector[+1], ChiralityProjector[-1]}
DiracSimplify[#, DiracSubstitute67 -> True] & /@ %
```

$$\left\{\bar{\gamma}^6, \bar{\gamma}^7\right\}$$

$$\left\{\frac{\bar{\gamma}^5}{2}+\frac{1}{2}, \frac{1}{2}-\frac{\bar{\gamma}^5}{2}\right\}$$

ChiralityProjector is scheduled for removal in the future versions of FeynCalc. The safe alternative is to use GA[6] and GA[7].

```
{GA[6], GA[7]}
```

$$\left\{\bar{\gamma}^6, \bar{\gamma}^7\right\}$$

```
FCI[GA[6]] === ChiralityProjector[+1]
```

True

```
FCI[GA[7]] === ChiralityProjector[-1]
```

True

15.6 ClearScalarProducts

ClearScalarProducts is equivalent to **FCClearScalarProducts[]**.

The shortcut **ClearScalarProducts** is deprecated, please use **FCClearScalarProducts** instead!

15.6.1 See also

Overview, FCClearScalarProducts.

15.6.2 Examples

ClearScalarProducts is scheduled for removal in the future versions of FeynCalc. The safe alternative is to use **FCClearScalarProducts**.

15.7 DiracMatrix

DiracMatrix[mu] denotes a Dirac gamma matrix with Lorentz index μ.

DiracMatrix[mu , nu , ...] is a product of γ matrices with Lorentz indices **mu , nu , ...**

DiracMatrix[5] is γ^5.

DiracMatrix[6] is $(1+\gamma^5)/2$.

DiracMatrix[7] is $(1-\gamma^5)/2$.

The shortcut **DiracMatrix** is deprecated, please use **GA** instead!

15.7.1 See also

Overview, GA, FCI.

15.7.2 Examples

```
DiracMatrix[\[Mu]]
```

$$\bar{\gamma}^{\mu}$$

This is how to enter the non-commutative product of two. The Mathematica Dot "." is used as non-commutative multiplication operator.

```
DiracMatrix[\[Mu]] . DiracMatrix[\[Nu]]
```

$$\bar{\gamma}^{\mu}.\bar{\gamma}^{\nu}$$

```
DiracMatrix[\[Alpha]] // StandardForm

(*DiracGamma[LorentzIndex[\[Alpha]]]*)
```

DiracMatrix is scheduled for removal in the future versions of FeynCalc. The safe alternative is to use **GA**.

```
GA[\[Mu]]
```

$$\bar{\gamma}^{\mu}$$

```
GAD[\[Mu]]
```

$$\gamma^{\mu}$$

```
FCI[GA[\[Mu]]] === DiracMatrix[\[Mu]]
```

True

```
FCI[GAD[\[Mu]]] === DiracMatrix[\[Mu], Dimension -> D]
```

True

15.8 DiracSlash

DiracSlash[p] is the contraction $p^{\mu}\gamma_{\mu}$ (**FV[p, mu] GA[mu]**).

Products of those can be entered in the form GS[p1, p2, . . .].

The shortcut DiracSlash is deprecated, please use GS instead!

15.8.1 See also

Overview, GS, FCI.

15.8.2 Examples

This is q-slash, i.e., $\gamma^{\mu}q_{\mu}$

```
DiracSlash[q]
```

$$\bar{\gamma} \cdot \bar{q}$$

```
DiracSlash[p] . DiracSlash[q]
```

$$(\bar{\gamma} \cdot \bar{p}) . (\bar{\gamma} \cdot \bar{q})$$

```
DiracSlash[p, q]
```

$$(\bar{\gamma} \cdot \bar{p}) . (\bar{\gamma} \cdot \bar{q})$$

DiracSlash is scheduled for removal in the future versions of FeynCalc. The safe alternative is to use GS.

```
GS[p]
```

$$\bar{\gamma} \cdot \bar{p}$$

```
GSD[p]
```

$$\gamma \cdot p$$

```
FCI[GS[p]] === DiracSlash[p]
```

True

```
FCI[GSD[p]] === DiracSlash[p, Dimension -> D]
```

True

15.9 DiracSpinor

DiracSpinor is equivalent to **Spinor**.

15.9.1 See also

Overview, Spinor.

15.9.2 Examples

15.10 FourVector

FourVector[p, mu] is the 4-dimensional vector **p** with Lorentz index **mu**.

A vector with space-time Dimension D is obtained by supplying the option **Dimension -> D**.

The shortcut **FourVector** is deprecated, please use **FV** instead!

15.10.1 See also

Overview, FV, FCI.

15.10.2 Examples

```
FourVector[p, \[Mu]]
```

$$\overline{p}^{\mu}$$

```
FourVector[p - q, \[Mu]]
```

$$(\overline{p} - \overline{q})^{\mu}$$

```
StandardForm[FourVector[p, \[Mu]]]

(*Pair[LorentzIndex[\[Mu]], Momentum[p]]*)
```

```
StandardForm[FourVector[p, \[Mu], Dimension -> D]]

(*Pair[LorentzIndex[\[Mu], D], Momentum[p, D]]*)
```

FourVector is scheduled for removal in the future versions of FeynCalc. The safe alternative is to use **FV**.

```
FV[p, \[Mu]]
```

$$\overline{p}^{\mu}$$

```
FVD[p, \[Mu]]
```

$$p^{\mu}$$

```
FCI[FV[p, \[Mu]]] === FourVector[p, \[Mu]]
```

True

```
FCI[FVD[p, \[Mu]]] === FourVector[p, \[Mu], Dimension -> D]
```

True

15.11 Gstrong

Gstrong is a shortcut for **SMP["g_s"]** which represents the strong coupling constant.

The shortcut **Gstrong** is deprecated, please use **SMP["g_s"]** instead!

15.11.1 See also

Overview, CovariantD, FieldStrength, GluonVertex.

15.11.2 Examples

```
Gstrong
```

$$g_s$$

```
Gstrong // StandardForm

(*SMP["g_s"]*)
```

15.12 IFPDOn

IFPDOn[exp_, q1, q2, ...] changes from **FeynAmpDenominator[...]** representation to the **IFPD** one (Inverse Feynman Propagator Denominator). I.e., **FeynAmpDenominator[PropagatorDenominator[a, b]]** is replaced by **1/IFPD[a, b]** and the **q1, q2, ...** are the integration momenta.

15.12.1 See also

Overview, IFPD, IFPDOff.

15.12.2 Examples

15.13 IFPDOff

IFPDOff[exp_, q1, q2, ...] changes from **IFPD** representation to **FeynAmpDenominator[...]**. The **q1, q2, ...** are the integration momenta.

15.13.1 See also

Overview, IFPD, IFPDOn.

15.13.2 Examples

```
IFPD[Momentum[p], m]
```

$$(\overline{p}^2 - m^2)$$

```
IFPD[Momentum[p], m] // StandardForm

(*IFPD[Momentum[p], m]*)
```

```
ex = IFPDOff[IFPD[Momentum[p], m], p]
```

$$\overline{p}^2 - m^2$$

```
ex // StandardForm

(*-m^2 + Pair[Momentum[p], Momentum[p]]*)
```

15.14 LeviCivita

LeviCivita[mu, nu, rho, si] is an input function for the totally antisymmetric Levi-Civita tensor. It evaluates automatically to the internal representation **Eps[LorentzIndex[mu], LorentzIndex[nu], LorentzIndex[rho], LorentzIndex[si]]** (or with a second argument in **LorentzIndex** for the **Dimension**, if the option **Dimension** of **LeviCivita** is changed).

LeviCivita[mu , nu, ...][p, ...] evaluates to **Eps[LorentzIndex[mu], LorentzIndex[nu], ..., Momentum[p], ...]**.

The shortcut **LeviCivita** is deprecated, please use **LC** instead!

15.14.1 See also

Overview, LC, FCI.

15.14.2 Examples

```
LeviCivita[\[Alpha], \[Beta], \[Gamma], \[Delta]]
```

$$\bar{\epsilon}^{\alpha\beta\gamma\delta}$$

```
LeviCivita[][p, q, r, s]
```

$$\bar{\epsilon}^{\overline{pqrs}}$$

```
LeviCivita[\[Alpha], \[Beta]][p, q]
```

$$\bar{\epsilon}^{\alpha\beta\overline{pq}}$$

```
LeviCivita[\[Alpha], \[Beta]][p, q] // StandardForm

(*Eps[LorentzIndex[\[Alpha]], LorentzIndex[\[Beta]], Momentum[p],
Momentum[q]]*)
```

LeviCivita is scheduled for removal in the future versions of FeynCalc. The safe alternative is to use **LC**.

```
LC[\[Alpha], \[Beta], \[Gamma], \[Delta]]
```

$$\bar{\epsilon}^{\alpha\beta\gamma\delta}$$

```
LC[][p, q, r, s]
```

$$\bar{\epsilon}^{\overline{pqrs}}$$

```
LC[\[Alpha], \[Beta]][p, q]
```

$$\bar{\epsilon}^{\alpha\beta\overline{pq}}$$

```
LCD[\[Alpha], \[Beta], \[Gamma], \[Delta]]
```

$$\epsilon^{\alpha\beta\gamma\delta}$$

```
LCD[][p, q, r, s]
```

$$\epsilon^{pqrs}$$

```
LCD[\[Alpha], \[Beta]][p, q]
```

$$\epsilon^{\alpha\beta pq}$$

```
FCI[LC[\[Alpha], \[Beta], \[Gamma], \[Delta]]] === LeviCivita[\[Alpha],
\[Beta], \[Gamma], \[Delta]]
```

True

```
FCI[LCD[\[Alpha], \[Beta], \[Gamma], \[Delta]]] === LeviCivita[\[Alpha],
\[Beta], \[Gamma], \[Delta], Dimension -> D]
```

True

15.15 $LoadFeynArts

$LoadFeynArts is a legacy switch for loading FeynArts. The modern way to achieve the same effect is to evaluate **$LoadAddOns={"FeynArts"}** before loading FeynCalc.

15.15.1 See also

Overview, $LoadAddOns.

15.15.2 Examples

```
$LoadFeynArts
```

False

15.16 $LoadPhi

$LoadPhi is a legacy switch for loading Phi. The modern way to achieve the same effect is to evaluate **$LoadAddOns={"Phi"}** before loading FeynCalc.

15.16.1 See also

Overview, $LoadAddOns.

15.16.2 Examples

```
$LoadPhi
```

False

15.17 $LoadTARCER

$LoadTARCER is a legacy switch for loading TARCER. The modern way to achieve the same effect is to evaluate **$LoadAddOns={"TARCER"}** before loading FeynCalc.

15.17.1 See also

Overview, $LoadAddOns.

15.17.2 Examples

```
$LoadTARCER
```

False

15.18 MetricTensor

MetricTensor[mu, nu] is the metric tensor. The default dimension is 4.

The shortcut **MetricTensor** is deprecated, please use **MT** instead!

15.18.1 See also

Overview, FCI, MT, MTD.

15.18.2 Examples

```
MetricTensor[\[Alpha], \[Beta]]

Contract[% %]
```

$$\bar{g}^{\alpha\beta}$$

$$4$$

```
MetricTensor[\[Alpha], \[Beta], Dimension -> D]

Contract[% %]
```

$$g^{\alpha\beta}$$

$$D$$

```
StandardForm[MetricTensor[a, b]]

(*Pair[LorentzIndex[a], LorentzIndex[b]]*)
```

MetricTensor is scheduled for removal in the future versions of FeynCalc. The safe alternative is to use **MT**.

```
MT[\[Mu], \[Nu]]
```

$$\bar{g}^{\mu\nu}$$

```
MTD[\[Mu], \[Nu]]
```

$$g^{\mu\nu}$$

```
FCI[MT[\[Mu], \[Nu]]] === MetricTensor[\[Mu], \[Nu]]

FCI[MTD[\[Mu], \[Nu]]] === MetricTensor[\[Mu], \[Nu], Dimension -> D]
```

True

True

15.19 OneLoop

OneLoop[q, amplitude] calculates the 1-loop Feynman diagram amplitude. The argument **q** denotes the integration variable, i.e., the loop momentum. **OneLoop[name, q, amplitude]** has as first argument a name of the amplitude. If the second argument has head **FeynAmp** then **OneLoop[q, FeynAmp[name, k, expr]]** and **OneLoop[FeynAmp[name, k, expr]]** tranform to **OneLoop[name, k, expr]**. **OneLoop** is deprecated, please use **TID** instead!

15.19.1 See also

Overview, ToPaVe, ToPaVe2, A0, A00, B0, B1, B00, B11, C0, D0.

15.19.2 Examples

```
-I/Pi^2 FAD[{q, m}]

OneLoop[q, %]
```

$$-\frac{i}{\pi^2\left(q^2-m^2\right)}$$

$$\mathrm{A}_0\left(m^2\right)$$

```
I ((el^2)/(16 Pi^4 (1 - D))) FAD[{q, mf}, {q - k, mf}] DiracTrace[(mf +
GSD[q - k]) . GAD[\[Mu]] . (mf + GSD[q]) . GAD[\[Mu]]]

OneLoop[q, %]
```

$$\frac{i\,\text{el}^2\,\text{tr}\left((\gamma\cdot(q-k)+\text{mf}).\gamma^{\mu}.(\text{mf}+\gamma\cdot q).\gamma^{\mu}\right)}{16\pi^4(1-D)\left(q^2-\text{mf}^2\right).\left((q-k)^2-\text{mf}^2\right)}$$

$$\frac{\text{el}^2\left(-\frac{8\,\text{mf}^2\,\text{B}_0\left(\overline{k}^2,\text{mf}^2,\text{mf}^2\right)}{1-D}+\frac{2(2-D)\overline{k}^2\,\text{B}_0\left(\overline{k}^2,\text{mf}^2,\text{mf}^2\right)}{1-D}+\frac{4D\,\text{A}_0\left(\text{mf}^2\right)}{1-D}-\frac{8\,\text{A}_0\left(\text{mf}^2\right)}{1-D}\right)}{16\pi^2}$$

15.20 OneLoopSum

OneLoopSum[FeynAmp[...], FeynAmp[...] , ...] will calculate a list of Feynman amplitudes by replacing FeynAmp step by step by **OneLoop**.

15.20.1 See also

Overview, OneLoop.

15.20.2 Examples

15.21 PartialFourVector

PartialFourVector is equivalent to **FourDivergence**.

The shortcut **PartialFourVector** is deprecated, please use **FourDivergence** instead!

15.21.1 See also

Overview, FourDivergence.

15.21.2 Examples

PartialFourVector is scheduled for removal in the future versions of FeynCalc. The safe alternative is to use **FourDivergence**.

15.22 PropagatorDenominatorExplicit

PropagatorDenominatorExplicit is equivalent to **FeynAmpDenominatorExplicit**.

The shortcut **PropagatorDenominatorExplicit** is deprecated, please use **FeynAmpDenominatorExplicit** instead!

15.22.1 See also

Overview, FeynAmpDenominatorExplicit.

15.22.2 Examples

15.23 ScalarProductCancel

ScalarProductCancel[exp, q1, q2, ...] cancels scalar products with propagators.

ScalarProductCancel[exp] cancels simple cases.

ScalarProductCancel is deprecated, please use the more powerful **ApartFF** instead.

15.23.1 See also

Overview, ApartFF, FCClearScalarProducts, ExpandScalarProduct, Pair, SP, SPC, SPD.

15.23.2 Examples

```
SPD[q, p] FAD[{q, m}, {q - p, 0}]

ScalarProductCancel[%, q]
```

$$\frac{p \cdot q}{(q^2 - m^2) \,.(q - p)^2}$$

$$\frac{m^2 + p^2}{2q^2 \,.((q - p)^2 - m^2)} - \frac{1}{2\,(q^2 - m^2)}$$

```
SPD[q2, p] SPD[q1, p] FAD[{q1, m}, {q2, m}, q1 - p, q2 - p, q2 - q1] //FCI

SPC[%, q1, q2, FDS -> True]
```

$$\frac{(p\cdot \text{q1})(p\cdot \text{q2})}{\left(\text{q1}^2-m^2\right).\left(\text{q2}^2-m^2\right).(\text{q1}-p)^2.(\text{q2}-p)^2.(\text{q2}-\text{q1})^2}$$

$$\frac{\left(m^2+p^2\right)^2}{4\left(\text{q1}^2-m^2\right).\left(\text{q2}^2-m^2\right).(\text{q1}-p)^2.(\text{q1}-\text{q2})^2.(\text{q2}-p)^2}$$
$$+\frac{m^2+p^2}{2\,\text{q1}^2.\text{q2}^2.\left((\text{q1}-p)^2-m^2\right).(\text{q1}-\text{q2})^2}-\frac{m^2+p^2}{2\left(\text{q1}^2-m^2\right).\left(\text{q2}^2-m^2\right).(\text{q1}-p)^2.(\text{q1}-\text{q2})^2}$$
$$-\frac{1}{2\left(\text{q1}^2-m^2\right).(\text{q1}-\text{q2})^2.(\text{q2}-p)^2}+\frac{1}{4\left(\text{q1}^2-m^2\right).\left(\text{q2}^2-m^2\right).(\text{q1}-\text{q2})^2}$$

15.24 SPC

SPC is an abbreviation for **ScalarProductCancel**.

15.24.1 See also

Overview, ScalarProductCancel.

15.24.2 Examples

15.25 ScalarProductExpand

ScalarProductExpand is equivalent to **ExpandScalarProduct**.

The shortcut **ScalarProductExpand** is deprecated, please use **ExpandScalarProduct** instead!

15.25.1 See also

Overview, ExpandScalarProduct.

15.25.2 Examples

ScalarProductExpand is scheduled for removal in the future versions of FeynCalc. The safe alternative is to use **ExpandScalarProduct**.

Index