New developments in FORM

Takahiro Ueda

Nikhef Theory Group

FORM website: https://www.nikhef.nl/~form/
Source code: https://github.com/vermaseren/form

Outline

Introduction: background

FORM: design principles

New release: FORM 4.2
FORM is a toolkit for formula manipulation. Symbolic computation instead of numerical computation. Often used in theoretical particle physics though not restricted to any specific field.
FORM is

a toolkit for formula manipulation
FORM is

a toolkit for formula manipulation

Symbolic computation instead of numerical computation

\[ A = \pi r^2 \]

vs.

\[ A = 3.141593 \times 5.000000 \times 5.000000 \]

\[ = 78.53982 \]
**FORM is**

a toolkit for formula manipulation

Symbolic computation instead of numerical computation

\[ A = \pi r^2 \]

vs.

\[ A = 3.141593 \times 5.000000 \times 5.000000 \]

\[ = 78.53982 \]

Often used in theoretical particle physics though not restricted to any specific field
Why symbolic?

Exact

More information about the mathematical structure

\[ B := A \]

\[ n = \pi n r / \text{two.osf} \]

\[ B = ( / \text{seven.osf} / \text{eight.osf} / \text{five.osf} / \text{three.osf} / \text{nine.osf} / \text{eight.osf} / \text{two.osf} ) / \text{one.osf} / \text{zero.osf} = / \text{eight.osf} / \text{nine.osf} / \text{three.osf} / \text{zero.osf} / \text{nine.osf} / \text{seven.osf} / \text{eight.osf} \times / \text{one.osf} / \text{zero.osf} / \text{one.osf} / \text{eight.osf} \]

But, see Laporta’s talk

PSLQ to recover analytical result

Theory is described in math (symbolic)
Why symbolic?

Exact
Why symbolic?

Exact

More information about the mathematical structure

\[ B := A^n = \pi^n r^{2n} \]

vs.

\[ B = (78.53982)^{10} = 8.930978 \times 10^{18} \]

but, see Laporta’s talk
PSLQ to recover analytical result
Why symbolic?

Exact

More information about the mathematical structure

\[ B := A^n = \pi^n r^{2n} \]

vs.

\[ B = (78.53982)^{10} = 8.930978 \times 10^{18} \]

☞ but, see Laporta’s talk
PSLQ to recover analytical result

Theory is described in math (symbolic)
Theories in QFTs

Framework of Quantum Field Theory
Theories in QFTs

Framework of Quantum Field Theory

Theory = Lagrangian $\mathcal{L}$

(model)
Theories in QFTs

Quantum electrodynamics (QED)

\[ \mathcal{L} = -\frac{1}{4} F_{\mu\nu} F^{\mu\nu} + \bar{\psi} (i\gamma^\mu - m) \gamma_\mu \psi \]

photon (light)  electron (matter)
Theories in QFTs

Quantum electrodynamics (QED)

\[ \mathcal{L} = -\frac{1}{4} F_{\mu\nu} F^{\mu\nu} + \bar{\psi}(i\gamma^\mu - m)\psi \]

- photon (light)
- electron (matter)
Theories in QFTs

Quantum chromodynamics (QCD)

\[ \mathcal{L} = -\frac{1}{4} F_{\mu\nu}^{a} F^{\mu\nu, a} + \bar{\psi}(i\slashed{D} - m)\psi \]

- gluon
- quark
Theories in QFTs

Quantum chromodynamics (QCD)

\[ \mathcal{L} = -\frac{1}{4} F^a_{\mu\nu} F^{\mu\nu, a} + \bar{\psi} (iD - m) \psi \]

- gluon self-interaction
- quark-gluon interaction

- gluon
- quark
- + Faddeev–Popov ghost
Theories in QFTs

The Standard Model (SM)

\[ 
\mathcal{L} = - \frac{1}{4} F_{\mu \nu} F^{\mu \nu} 
\]

\[ + i \bar{\psi} D^\mu \psi + \text{h.c.} \]

\[ + \bar{\chi}_i y_{ij} \chi_j \phi + \text{h.c.} \]

\[ + \partial_\mu \phi \partial^\mu \phi - V(\phi) \]

Picture from CERN Shop
Theories in QFTs

The Standard Model (SM)

quarks & leptons
a Higgs boson
QED + QCD
+ weak interaction
Theory prediction
Theory prediction

electron quantum effect?

perturbative expansion

QED coupling: $\alpha \sim \frac{1}{\text{three.osf/seven.osf}} \ll \frac{1}{\text{one.osf/three.osf}}$ but, see Kronfeld's talk non-perturbative method
Theory prediction

electron quantum effect?

\[ = \mathcal{N} \left\langle 0 \left| T \left\{ \psi(x) \bar{\psi}(y) \exp \left[ i \int d^4z \mathcal{L}_{\text{int}}(z) \right] \right\} \right| 0 \right\rangle \]
Theory prediction

\[ \mathcal{N} \langle 0 | T \{ \psi(x) \bar{\psi}(y) \exp \left[ i \int d^4 z \mathcal{L}_{\text{int}}(z) \right] \} | 0 \rangle \]

electron quantum effect?

Your Theory

perturbative expansion

\[ = \alpha + \frac{\alpha}{2} + \frac{\alpha}{3} + \ldots \]

QED coupling:

\[ \alpha \sim \frac{1}{\sqrt{\pi}} \]

but, see Kronfeld's talk
Theory prediction

Electron quantum effect?

\[ \mathcal{N} \left\langle 0 \right| T \left\{ \psi(x)\bar{\psi}(y) \exp \left[ i \int d^4z \mathcal{L}_{\text{int}}(z) \right] \right\} \left| 0 \right\rangle \]

Perturbative expansion

\[ = \begin{array}{c} \text{vac}\end{array} + \alpha \begin{array}{c} \text{1st}\end{array} + \alpha^2 \begin{array}{c} \text{2nd}\end{array} + \alpha^3 \begin{array}{c} \text{3rd}\end{array} + \ldots \]

QED coupling: \[ \alpha \sim \frac{1}{137} \ll 1 \]

But, see Kronfeld’s talk non-perturbative method.
Perturbative expansion: How to?
Perturbative expansion: How to?

- Draw Feynman diagrams

Picture from Nobelprize.org
Perturbative expansion: How to?

Draw Feynman diagrams

freely propagating electron and photon
Perturbative expansion: How to?

Draw Feynman diagrams

freely propagating electron and photon

interaction vertex

Picture from Nobelprize.org
Perturbative expansion: How to?

Draw Feynman diagrams

freely propagating electron and photon

interaction vertex

$O(\alpha)$-contribution
Perturbative expansion: How to?
Perturbative expansion: How to?

Translated into math expressions
Life is not easy

Many diagrams, possibly $\sim 1M$
Life is not easy

Many diagrams, possibly ~ 1M

Complicated structure
Life is not easy

Many diagrams, possibly $\sim 1M$

Complicated structure

$$\rightarrow \text{Tr}(\gamma^{\mu_1} \gamma^{\mu_2} \ldots \gamma^{\mu_{16}})$$

$$= 4g^{\mu_1\mu_2} g^{\mu_3\mu_4} \ldots g^{\mu_{15}\mu_{16}} + \ldots$$

$$\frac{16!}{2^{16/2} (16/2)!} = 2027025 \text{ terms in total}$$
Life is not easy

Many diagrams, possibly $\sim 1M$

Complicated structure

$$\rightarrow \text{Tr}(\gamma^{\mu_1}\gamma^{\mu_2} ... \gamma^{\mu_{16}})$$

$$= 4g^{\mu_1\mu_2}g^{\mu_3\mu_4} ... g^{\mu_{15}\mu_{16}} + \cdots$$

$$\frac{16!}{2^{16/2}(16/2)!} = 2027025 \text{ terms in total}$$

Divergent integral
Life is not easy

Many diagrams, possibly \( \sim 1\text{M} \)

Complicated structure

\[
\rightarrow \text{Tr}(\gamma_{\mu_1} \gamma_{\mu_2} \ldots \gamma_{\mu_{16}})
= 4g_{\mu_1\mu_2} g_{\mu_3\mu_4} \ldots g_{\mu_{15}\mu_{16}} + \ldots
\]
\[
\frac{16!}{2^{16/2} (16/2)!} = 2027025 \text{ terms in total}
\]

Divergent integral

\[
\sim \infty
\]
Life is not easy

Many diagrams, possibly \( \sim 1 \text{M} \)

Complicated structure

\[
\rightarrow \text{Tr}(\gamma^{\mu_1} \gamma^{\mu_2} \ldots \gamma^{\mu_{16}}) = 4g^{\mu_1 \mu_2}g^{\mu_3 \mu_4} \ldots g^{\mu_{15} \mu_{16}} + \ldots
\]

\[
\frac{16!}{2^{16/2}(16/2)!} = 2027025 \text{ terms in total}
\]

Divergent integral

\[\sim \infty\]

Keep the divergence as a symbol until cancelled
Perturbative QFT requires symbolic manipulation with many terms. Basic algebraic operations. No need for fancy automatic integration or knowledge of special functions. Huge expression size $\sim 10^{10}$.

Q. Which computer algebra system(s) can be suited?

- FORM
- GiNaC
- Maple
- Mathematica
- Maxima
- Reduce
- SageMath

...
Summary so far

Perturbative QFT requires symbolic manipulation with many terms

Q. Which computer algebra system(s) can be suited?

F
ORM
GiNaC
Maple
Mathematica
Maxima
Reduce
SageMath

...
Summary so far

Perturbative QFT requires symbolic manipulation with many terms

Basic algebraic operations. No need for fancy automatic integration or knowledge of special functions.
Summary so far

Perturbative QFT requires symbolic manipulation with many terms

Basic algebraic operations. No need for fancy automatic integration or knowledge of special functions

Huge expression size $\sim 1.5\text{–}2\text{TB}$
Summary so far

Perturbative QFT requires symbolic manipulation with many terms

Basic algebraic operations. No need for fancy automatic integration or knowledge of special functions

Huge expression size $\sim 1.5\text{–}2\text{TB}$

Q. Which computer algebra system(s) can be suited?

**FORM** GiNaC Maple Mathematica Maxima Reduce SageMath ...
Summary so far

Perturbative QFT requires symbolic manipulation with many terms

Basic algebraic operations. No need for fancy automatic integration or knowledge of special functions
Huge expression size $\sim 1.5$–$2$TB

Q. Which computer algebra system(s) can be suited?

**FORM** GiNaC Maple Mathematica Maxima Reduce SageMath ...
Outline

Introduction: background

FORM: design principles

New release: FORM 4.2
Expression as tree

Most of computer algebra systems

\[(a + b)(c + d)\]
Expression as tree

Most of computer algebra systems

\[(a + b)(c + d)\]
Expression as tree

Most of computer algebra systems

\[(a + b)(c + d)\]

Elegant Recursive structure 😊
Expression as tree

Most of computer algebra systems

\[(a + b)(c + d)\]

Elegant
Recursive structure 😊

Random access may cause heavy swapping 😞
Expression as sequence

**FORM**

\[(a + b)(c + d)\]
Expression as sequence

**FORM always expands expressions**

\[(a + b)(c + d) = ac + ad + bc + bd\]
Expression as sequence

**FORM** always expands expressions

\[(a + b)(c + d) = ac + ad + bc + bd\]

Sequence of (sorted) terms
Each term has complete information
Local & global operations

Local operations on each term
Global operations, e.g., sorting

Input

```
+ b + ··· +
```

```
a → x + y
b → y + z
```

set of local ops.
global op.

Output

Sequential access only. Works with disks

```
/a/osf/three.osf/two.osf/seven.osf
```
Local & global operations

Local operations on each term
Global operations, e.g., sorting

Input

\[ a \rightarrow x + y \]
\[ b \rightarrow y + z \]

set of local ops.

global op.

Sorting

\[ a \rightarrow x + c \]
\[ b \rightarrow y + c \]

\[
\begin{align*}
\text{External merge sort} & \\
\text{Sequential access only. Works with disks} &
\end{align*}
\]

/one.osf/three.osf/two.osf/seven.osf
Local & global operations

Local operations on each term
Global operations, e.g., sorting

Input

\[ a \rightarrow x + y \]
\[ b \rightarrow y + z \]

set of local ops.

Global op.

Sorting
External merge sort

Output

Sequential access only. Works with disks 😊
Parallelization

Master-worker model

Input

\[ a \rightarrow x + y \]

\[ b \rightarrow y + z \]

set of local ops.

\[ a \rightarrow x + y \]

\[ b \rightarrow y + z \]

global op.

TFORM (POSIX Threads), ParFORM (MPI)

Output

master

worker 1

worker 2

master

External merge sort

Sorting

External merge sort

Sorting

Worker 1

Worker 2

Input

\[ a \rightarrow x + y \]

\[ b \rightarrow y + z \]

set of local ops.

\[ a \rightarrow x + y \]

\[ b \rightarrow y + z \]

global op.
FORM as language

Imperative programming, term rewriting
Just write operations for every term

\[ \text{id } x^n? = n \times x^{(n-1)}; \]

Replace \( x^n \rightarrow nx^{n-1} \) for \( n \in \mathbb{Z} \), i.e., \( \frac{\partial}{\partial x} \)
Form as language

Imperative programming, term rewriting
Just write operations for every term

\[ \text{id } x^n? = n \ast x^{(n-1)}; \]

Replace \( x^n \rightarrow nx^{n-1} \) for \( n \in \mathbb{Z} \), i.e., \( \frac{\partial}{\partial x} \)

Powerful ‘preprocessor’

```plaintext
#procedure derivative(x,m)
   #do i=1,`m'
      id `x'^n? = n \ast `x'^{(n-1)};
   #enddo
#endprocedure

#call derivative(y,2)
```

Define \( \frac{\partial^m}{\partial x^m} \)

x and m are parametrized

Perform \( \frac{\partial^2}{\partial y^2} \)
On highfirst;  * term order
Symbol a,b;
Local F = (a+b)^2;
Print;
.end
On highfirst; * term order
Symbol a,b;
Local F = (a+b)^2;
Print;
.end

Run FORM as form example.frm

FORM 4.2.0 (Jul 6 2017, v4.2.0) 64-bits Run: Thu Aug 17 18:10:24 2017
On highfirst; * term order
Symbol a,b;
Local F = (a+b)^2;
Print;
.end

Time = 0.00 sec Generated terms = 3
F Terms in output = 3
Bytes used = 108

F = a^2 + 2*a*b + b^2;

0.00 sec out of 0.00 sec
On highfirst;  * term order
Symbol a,b,x;
Local F = a*x + x^2;

id x = a + b;

Print;
.sort

if (count(b,1) == 1);  * only for b^1
    multiply a/b;
endif;

Print;
.end
On highfirst;  * term order
Symbol a,b,x;
Local F = a*x + x^2;
id x = a + b;
Print;
.sort
if (count(b,1) == 1);  * only for b^1
    multiply a/b;
endif;
Print;
.end
On highfirst;  * term order
Symbol a,b,x;
Local F = a*x + x^2;
id x = a + b;
Print;
  .sort
if (count(b,1) == 1);  * only for b^1
  multiply a/b;
 endif;
Print;
  .end

Compile & Run
User controls sorting

Compile & Run
Preprocessor talks with processor

Compile-time optimization by the result of previous blocks

```c
#define HaveX "0"

* Check whether expressions have x.
if (count(x,1));
    redefine HaveX "1";
endif;

.sort

#if `HaveX'
* This part is skipped at compile-time
* when there is no x in the expressions.
#endif
```
Physics applications
FORM has been used in many big computations

1. New features of FORM
   e-Print: math-ph/0010025 | PDF
   References | BibTeX | LaTeX-US | LaTeX-EU | HarvMac | EndNote
   ADS Abstract Service; Link to GitHub
   Detailed record - Cited by 1291 records

2. FORM version 4.0
   NIKHEF-2012-004, TTP12-008, SFB-CPP-12-15
   References | BibTeX | LaTeX-US | LaTeX-EU | HarvMac | EndNote
   ADS Abstract Service
   Detailed record - Cited by 199 records

Welcome to INSPIRE, the High Energy Physics information system. Please direct questions, comments or concerns to feedback@inspirehep.net.
Physics applications

Because any choice of physics applications from 1291 citations would be biased, I just mention the keywords of our recent works:

**FORCER** and **R**

They were the driving force of the developments of **FORM** 4.2

- TU’s talk in ACAT 2016
- see Ruijl’s talk
Outline

Introduction: background

FORM: design principles

New release: FORM 4.2
FORM 4.2

V4.0  2012-03-30  Kuipers, TU, Vermaseren, Vollinga
Polyomial factorization, rational polynomials, etc.

V4.1  2013-10-25  Kuipers, TU, Vermaseren
Code output optimization, etc.

V4.2  2017-07-06  Ruijl, TU, Vermaseren
Generating all matches, automatic expansion of rational polynomials, dictionaries, spectators, etc.

v4.2 contains more than 20 new features & 50 bugfixes
FORM in GitHub ecosystem

Hosted on GitHub since 2013-09-11
(Just before v4.1)

---

FORM

build passing coverage 50%

FORM is a Symbolic Manipulation System. It reads symbolic expressions from files and executes symbolic/algebraic transformations upon them. The answers are returned in a textual mathematical representation. As its landmark feature, the size of the considered expressions in FORM is only limited by the available disk space and not by the available RAM.

FORM's original author is Jos Vermaseren of NIKHEF, the Dutch institute for subatomic physics. Other people that have made contributions can be found in the file "AUTHORS".
FORM in GitHub ecosystem

Hosted on GitHub since 2013-09-11
(Just before v4.1)

FORM

FORM is a Symbolic Manipulation System. It reads symbolic expressions from files and executes symbolic/algebraic transformations upon them. The answers are returned in a textual mathematical representation. As its landmark feature, the size of the considered expressions in FORM is only limited by the available disk space and not by the available RAM.

FORM's original author is Jos Vermaseren of NIKHEF, the Dutch institute for subatomic physics. Other people that have made contributions can be found in the file "AUTHORS".

Issue tracker (reporting bugs, feature requests, questions)

Wiki (release notes, installation notes)
FORM in GitHub ecosystem

Hosted on GitHub since 2013-09-11
(Just before v4.1)

FORM is a Symbolic Manipulation System. It reads symbolic expressions from files and executes symbolic/algebraic transformations upon them. The answers are returned in a textual mathematical representation. As its landmark feature, the size of the considered expressions in FORM is only limited by the available disk space and not by the available RAM.

FORM's original author is Jos Vermaseren of NIKHEF, the Dutch institute for subatomic physics. Other people that have made contributions can be found in the file "AUTHORS".

Every commit triggers
- Running test programs on Travis CI
- Code coverage test on Coveralls
**Form** in GitHub ecosystem

Hosted on GitHub since 2013-09-11

(Just before v4.1)

---

**Travis CI**

**Coveralls**

**FORM**

FORM is a Symbolic Manipulation System with Perl transformations upon them. The answer size of the considered expressions in this case is 50%.

FORM's original author is Jos Vermaas. Additional made contributions can be found in the development history.

---

**Downloads**

- form-4.2.0-manual-html.tar.gz (457 KB)
- form-4.2.0-manual.pdf (878 KB)
- form-4.2.0-x86_64-linux.tar.gz (3.28 MB)
- form-4.2.0-x86_64-osx.tar.gz (2.08 MB)
- form-4.2.0.tar.gz (1.32 MB)

---

Deployment from Travis CI to Github Release
FORCER and R* program need graph manipulation
Graph manipulation

FORCER and R* program need graph manipulation

Undirected graph:

\[ p_1, Q, Q, p_2 \]
**FORCER** and **R* program need graph manipulation**

Undirected graph:

\[ \text{CFunction } vx(\text{symmetric}); \] * commuting & symmetric

\[ \text{Vector } Q, p_1, ..., p_{99}; \]

\[ \text{Local } F = vx(Q, p_1, p_2) * vx(Q, p_1, p_2); \]
Graph isomorphism

Q. Are they equivalent?

Local \( F_1 = \text{vx}(Q,p_1,p_8)\times \text{vx}(p_1,p_2,p_9)\times \text{vx}(p_2,p_3,p_{11})\times \text{vx}(p_3,p_4,p_{10})\times \text{vx}(Q,p_4,p_5)\times \text{vx}(p_5,p_6,p_9)\times \text{vx}(p_6,p_7,p_{11})\times \text{vx}(p_7,p_8,p_{10}); \)

Local \( F_2 = \text{vx}(Q,p_1,p_6)\times \text{vx}(p_1,p_2,p_{10})\times \text{vx}(p_2,p_3,p_{11})\times \text{vx}(Q,p_3,p_4)\times \text{vx}(p_4,p_5,p_9)\times \text{vx}(p_5,p_6,p_7)\times \text{vx}(p_7,p_8,p_{11})\times \text{vx}(p_8,p_9,p_{10}); \)
Graph isomorphism

CFunction map;  * for isomorphic mapping
CFunction vx(symmetric);
Vector Q,p1,...,p11,q1,...,q11;

Local F1 = vx(Q,p1,p8)*vx(p1,p2,p9)*vx(p2,p3,p11)*
vx(p3,p4,p10)*vx(Q,p4,p5)*vx(p5,p6,p9)*
vx(p6,p7,p11)*vx(p7,p8,p10);

Local F2 = vx(Q,p1,p6)*vx(p1,p2,p10)*vx(p2,p3,p11)*
vx(Q,p3,p4)*vx(p4,p5,p9)*vx(p5,p6,p7)*
vx(p7,p8,p11)*vx(p8,p9,p10);

id vx(Q,q1?,q8?)*vx(q1?,q2?,q9?)*vx(q2?,q3?,q11?)*
  vx(q3?,q4?,q10?)*vx(Q,q4?,q5?)*vx(q5?,q6?,q9?)*
  vx(q6?,q7?,q11?)*vx(q7?,q8?,q10?)
  = map(q1,...,q11);
Print;
.end
Graph isomorphism

```c
CFun cction map;  * for isomorphic mapping
CFun cction vx(symmetric);
Vector Q,p1,...,p11,q1,...,q11;

Local F1 = vx(Q,p1,p8)*vx(p1,p2,p9)*vx(p2,p3,p11)*
            vx(p3,p4,p10)*vx(Q,p4,p5)*vx(p5,p6,p9)*
            vx(p6,p7,p11)*vx(p7,p8,p10);

Local F2 = vx(Q,p1,p6)*vx(p1,p2,p10)*vx(p2,p3,p11)*
            vx(Q,p3,p4)*vx(p4,p5,p9)*vx(p5,p6,p7)*
            vx(p7,p8,p11)*vx(p8,p9,p10);

id vx(Q,q1?,q8?)*vx(q1?,q2?,q9?)*vx(q2?,q3?,q11?)*
    vx(q3?,q4?,q10?)*vx(Q,q4?,q5?)*vx(q5?,q6?,q9?)*
    vx(q6?,q7?,q11?)*vx(q7?,q8?,q10?)
    = map(q1,...,q11);
Print;
.end
```

Yes. Got the isomorphic mapping

Pattern with q1?,...,q11?
Graph automorphism

\[ p_1 \quad p_2 \quad Q \quad p_5 \quad Q \quad p_3 \quad p_4 \]

**id all** generates all matches

\[(v_4.2)\]

\begin{verbatim}
CFunction map;
CFunction vx(symmetric);
Vector Q,p1,...,p5,q1,...,q5;

Local F = vx(Q,p1,p4)*vx(p1,p2,p5)*vx(Q,p2,p3)*
    vx(p3,p4,p5);

id all,vx(Q,q1?,q4?)*vx(q1?,q2?,q5?)*vx(Q,q2?,q3?)*
    vx(q3?,q4?,q5?)
    = map(q1,...,q5);
Print +s;
.end
\end{verbatim}
Graph automorphism

\[
F = \begin{align*}
&+ \text{map}(p_1, p_2, p_3, p_4, p_5) \\
&+ \text{map}(p_2, p_1, p_4, p_3, p_5) \\
&+ \text{map}(p_3, p_4, p_1, p_2, p_5) \\
&+ \text{map}(p_4, p_3, p_2, p_1, p_5) \\
&; 
\end{align*}
\]

Exhibits \(Z_2 \times Z_2\) symmetry
Creative (ab)use of features

\[
\text{id } vx(Q,q1?,q8?) \ast vx(q1?,q2?,q9?) \ast vx(q2?,q3?,q11?) \ast \\
vx(q3?,q4?,q10?) \ast vx(Q,q4?,q5?) \ast vx(q5?,q6?,q9?) \ast \\
vx(q6?,q7?,q11?) \ast vx(q7?,q8?,q10?) \\
= \text{map}(q1,\ldots,q11);
\]

\[
\text{id } all \ast vx(Q,q1?,q4?) \ast vx(q1?,q2?,q5?) \ast vx(Q,q2?,q3?) \ast \\
vx(q3?,q4?,q5?) \\
= \text{map}(q1,\ldots,q5);
\]

The patterns are rather static. Better way?
Creative (ab)use of features

“Dictionary” at compile-time (v4.2)

```plaintext
#$graph1 = F1;
#opendictionary wild
    #do i=1,11
        #add p'\ i': "q\ i'"
    #endo
do
closedictionary
#usedictionary wild($)
    id `#$graph1' = map(q1,...,q11);
closedictionary
```

Dictionary: e.g.,
mu1 → “\mu_1”

Translates

```plaintext
Local F1 = vx(Q,p1,p8)*vx(p1,p2,p9)*vx(p2,p3,p11)*
    vx(p3,p4,p10)*vx(Q,p4,p5)*vx(p5,p6,p9)*
    vx(p6,p7,p11)*vx(p7,p8,p10);
```

into

```plaintext
id vx(Q,q1?,q8?)*vx(q1?,q2?,q9?)*vx(q2?,q3?,q11?)*
    vx(q3?,q4?,q10?)*vx(Q,q4?,q5?)*vx(q5?,q6?,q9?)*
    vx(q6?,q7?,q11?)*vx(q7?,q8?,q10?)
= map(q1,...,q11);
```
Creative (ab)use of features

Replacement on the LHS at run-time \((v4.2)\)

$\text{graph1} = F1;
\text{id } \text{graph1 } * \text{replace_}(<p1,q1?>,...,<p11,q11?>)
  = \text{map}(q1,...,q11);

\begin{align*}
\text{Local } F1 &= \text{vx}(Q,p1,p8)*\text{vx}(p1,p2,p9)*\text{vx}(p2,p3,p11)*\text{vx}(p3,p4,p10)*\text{vx}(Q,p4,p5)*\text{vx}(p5,p6,p9)*\text{vx}(p6,p7,p11)*\text{vx}(p7,p8,p10); \\
\text{id } \text{vx}(Q,q1?,q8?)*\text{vx}(q1?,q2?,q9?)*\text{vx}(q2?,q3?,q11?)*\text{vx}(q3?,q4?,q10?)*\text{vx}(Q,q4?,q5?)*\text{vx}(q5?,q6?,q9?)*\text{vx}(q6?,q7?,q11?)*\text{vx}(q7?,q8?,q10?)
  &= \text{map}(q1,...,q11);
\end{align*}
Conclusion
Conclusion

**FORM** is designed for symbolic manipulation of very big expressions, required by perturbative QFT
FORM is designed for symbolic manipulation of very big expressions, required by perturbative QFT

Used in many cutting edge computations
Conclusion

**FORM** is designed for symbolic manipulation of very big expressions, required by perturbative QFT

Used in many cutting edge computations

New features in **FORM** 4.2 help users to make ever smarter programs
Conclusion

**FORM** is designed for symbolic manipulation of very big expressions, required by perturbative QFT

Used in many cutting edge computations

New features in **FORM** 4.2 help users to make ever smarter programs

Questions? Or ?