PLCverif Extension: Verifying LTL Properties via Monitor Generation

Xaver Fink, Borja Fernández Adiego – *BE-ICS-ACS*
Anastasia Mavridou, Andreas Katis – *KBR at NASA Ames Research Center*

December 9, 2024

Motivation: PLC Installations at CERN

Context: Large number of (industrial) control systems with PLCs at CERN

- Hundreds of **control systems** with thousands ($\approx$ 3050 in 2023) of **Programmable Logic Controllers** (PLC)
  - Installed in: Cryogenics, ventilation, vacuum systems ...

Figure 1: Siemens S7-1500 F PLC

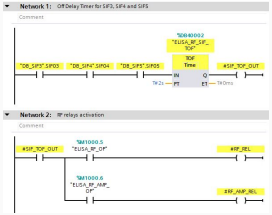
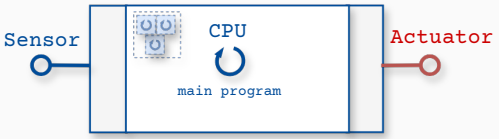


Figure 2: PLC Ladder diagram program

Motivation: PLC software development for Installations at CERN




```
(* Interlocks / FailSafe *)
IF #POutOff THEN
    IF #InterlockR THEN
        IF #PFsPulse AND NOT #PFsPosOn2 THEN
            IF #PFsPosOn THEN
                #OutOnOVSt := #PulseOn;
                #OutOffOVSt := FALSE;
                #OutOVSt_aux := TRUE;
            ELSE
                #OutOnOVSt := FALSE;
                #OutOffOVSt := #PulseOff;
                #OutOVSt_aux := FALSE;
            END_IF;
        ELSE
            #OutOnOVSt := (#PFsPosOn AND NOT #PFsPosOn2) OR (#PFsPosOn AND #PFsPosOn2);
            #OutOffOVSt := (NOT #PFsPosOn AND NOT #PFsPosOn2) OR (#PFsPosOn AND #PFsPosOn2);
            #OutOVSt_aux := #PFsPosOn;
        END_IF;
    END_IF;
ELSE
    IF #InterlockR THEN
        #OutOnOVSt := #PFsPosOn;
        #OutOVSt_aux := #PFsPosOn;
    END_IF;
END_IF;
```

Figure 3: PLC Structured Text program

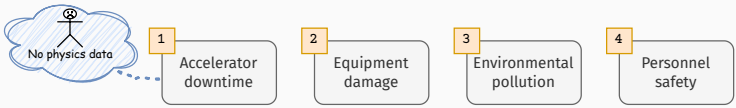
PLC Scan Cycle simplified behaviour:

1. Reading the actual values from sensors to the Input Image Memory
2. Interpreting and executing the PLC program
3. Writing the computed values from the Output Image Memory to the actuators

Motivation: Critical PLC Installations at CERN

Critical applications: Cryogenics, ventilation, vacuum systems.

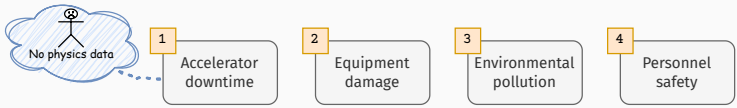
PLC software failures have significant negative consequences



Motivation: Critical PLC Installations at CERN

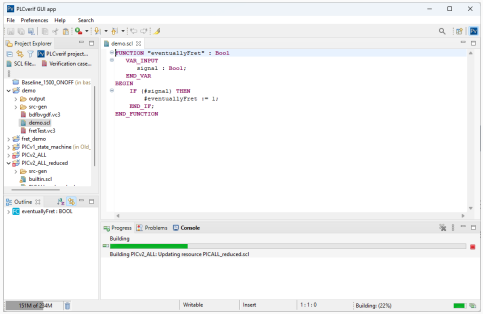
Critical applications: Cryogenics, ventilation, vacuum systems.

PLC software failures have significant negative consequences

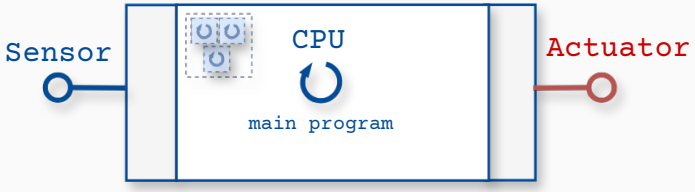


PLCverif: PLC verification

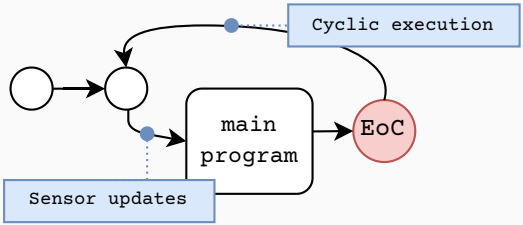
- Formal verification tool for PLC programs actively developed and used in BE-ICS PLC development
- Complements traditional testing for systems of higher criticality



Motivation: PLC model for verification



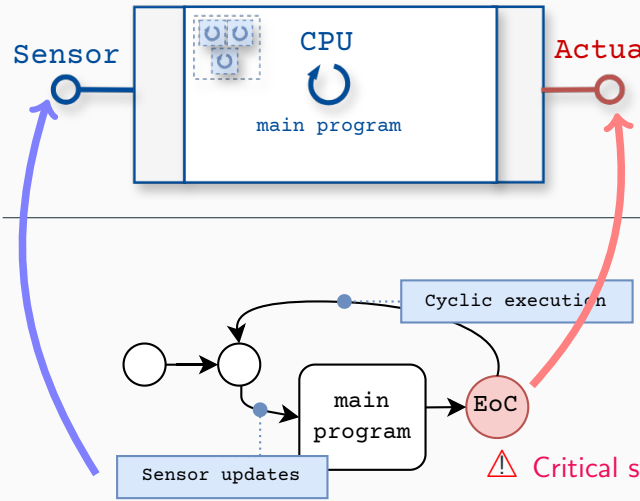
PLC behavior (simplified)



PLCverif abstraction

- Infinite program loop
- Sensor signals modelled as non-deterministic variables
- Critical system state at actuator updates

Motivation: PLC model for verification



PLC behavior (simplified)

PLCverif abstraction

- Infinite program loop
- Sensor signals modelled as non-deterministic variables
- Critical system state at actuator updates

Motivation: PLCverif pipeline

Main structure: Pipeline from PLC source code → Verification report

PLC program | Specification | CFA representation + reductions + property parsing | Verification | Reporting

```
FUNCTION FC1 : VOID
    VAR_INPUT
        in1 : BOOL;
        in2: BOOL;
        in3 : BOOL;
        in4: BOOL;
    END_VAR
    VAR_OUTPUT
        out1 : BOOL;
        out2 : BOOL;
    END_VAR
BEGIN
    // logic for out1
    IF in1 OR NOT in2 THEN
        out1 := NOT in3 OR in4;
    ELSE
        out1 := FALSE;
    END_IF;

    // logic for out2
    out2 := (in1 OR NOT in2) AND
            (NOT in3 or in4);

END_FUNCTION
```

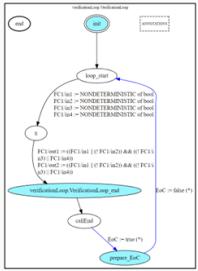
IEC 61131-3 languages
SCL, STL, Ladder

Assertion

Pattern

FRETish

via FRET



$$\mathbf{H}((RF\_CMD \wedge \mathbf{Z}(RF\_CMD)) \rightarrow ELISA\_RF\_PRESSURE\_SWS!)$$

Intemediate model (IM)
Reduction techniques

CBMC

Otheta

NuSMV

ESBMC

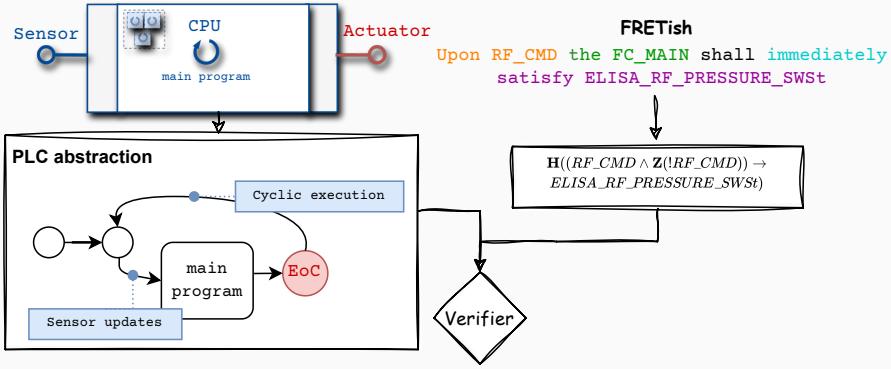
Variety of external tools

Counterexample

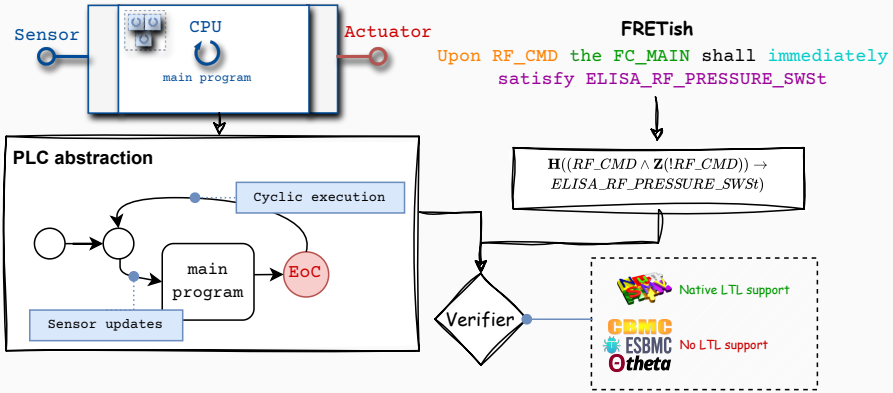
	Variable	End of **Cycle 1**
INPUT BOOL	**FC1.in1**	true
INPUT BOOL	**FC1.in2**	false
INPUT BOOL	**FC1.in3**	true
INPUT BOOL	**FC1.in4**	false
OUTPUT BOOL	**FC1.out1**	false
OUTPUT BOOL	**FC1.out2**	false

User Interaction | PLCverif Internal

Problem Setup



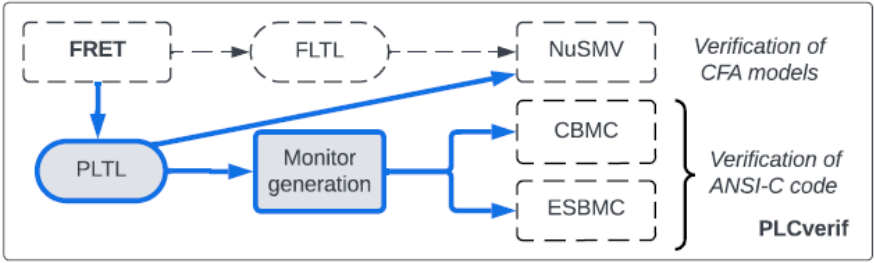
Problem Setup



Problem: not all verification backends support Linear Temporal Logic (LTL) specifications ⚠

A prior work integrated the tool FRET into PLCverif[1]

→ ⚠ LTL property verification was only possible via NuSMV



Our contribution:

1. Extend PLCverif with an algorithm for monitor-based verification of LTL properties via Bounded Model Checking (BMC) assertion-verification
2. Validate the algorithm on two CERN case studies

[1] Adam et al., 2023

Problem Definition

[1]Adiego et al., 2014

Goal: LTL-property verification via PLCverif with arbitrary verification backends

→ via problem instance modification, **without** verification algorithm modification

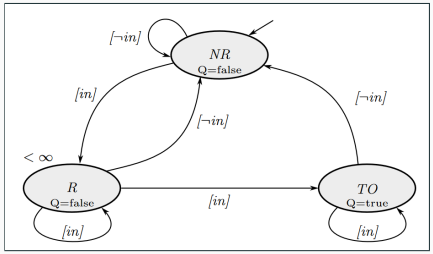


Figure 4: "ON delay timer" property modeled as state machine[1]

Prior manual approach:

1. Create state machine for property
2. Extend PLC program with the state machine
3. Verify assertion over the state machine

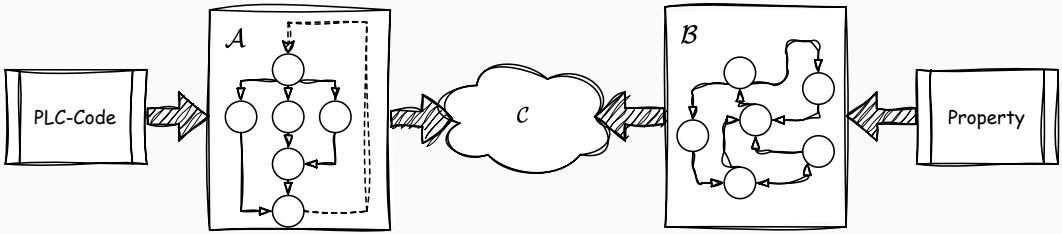
How can we generalize this?

Problem Definition: Generalized Algorithm

Goal: LTL-property verification via PLCverif with arbitrary verification backends

Planned generalized pipeline:

1. Transform PLC code to CFA $\mathcal{A}$ + LTL to monitor automaton $\mathcal{B}$
2. Merge $\mathcal{A}$ and $\mathcal{B}$ to combined CFA $\mathcal{C}$
3. Verify merged automaton $\mathcal{C}$

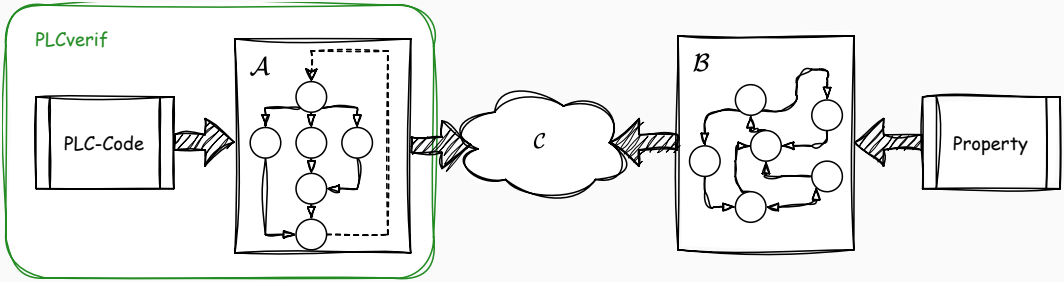


Problem Definition: Generalized Algorithm

Goal: LTL-property verification via PLCverif with arbitrary verification backends

Planned generalized pipeline:

1. Transform PLC code to CFA $\mathcal{A}$ + LTL to monitor automaton $\mathcal{B}$
2. Merge $\mathcal{A}$ and $\mathcal{B}$ to combined CFA $\mathcal{C}$
3. Verify merged automaton $\mathcal{C}$



Verification Pipeline: 1. Monitor construction

[1] Havelund & Rosu, 2002

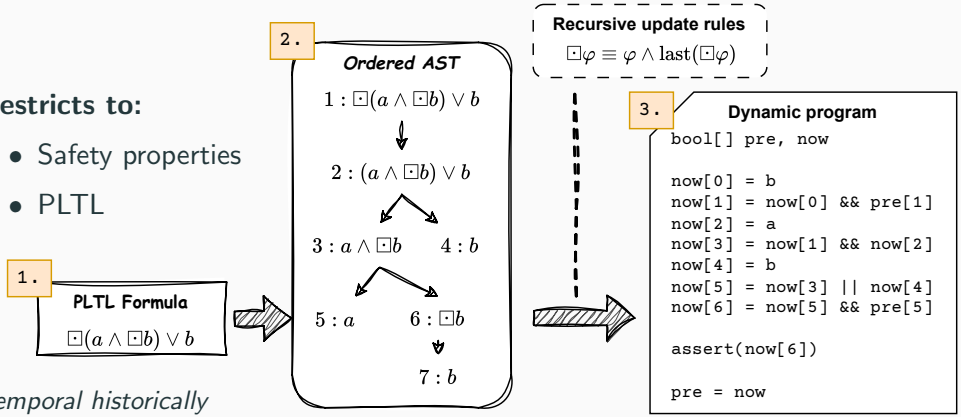
1. Transform PLC code to CFA $\mathcal{A}$ + **LTL to monitor automaton $\mathcal{B}$**

Monitoring algorithm by Havelund & Rosu [1] based on dynamic programming

Key idea: PLTL evaluation of $w, i \models \varphi$ only depends on current and last state

Restricts to:

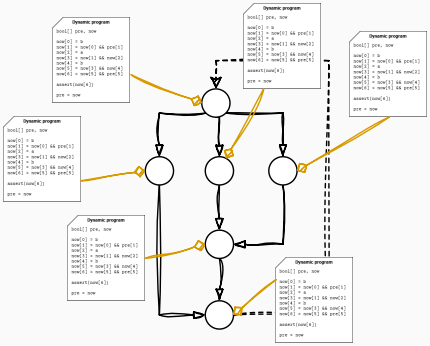
- Safety properties
- PLTL



$\boxdot$: *temporal historically*

2. Merge $\mathcal{A}$ and $\mathcal{B}$ to combined CFA $\mathcal{C}$

Naive approach: Insert dynamic program computation after each assignment



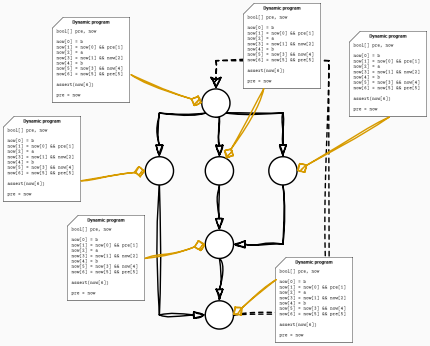
Problem: Efficiency ⚠

We tackle this problem via:

1. Sparse updates based on Cone-of-Influence
   $\rightarrow$ Uses variable interdependencies
2. State-space reduction on $\mathcal{A}$

2. Merge $\mathcal{A}$ and $\mathcal{B}$ to combined CFA $\mathcal{C}$

Naive approach: Insert dynamic program computation after each assignment



Problem: Efficiency ⚠

We tackle this problem via:

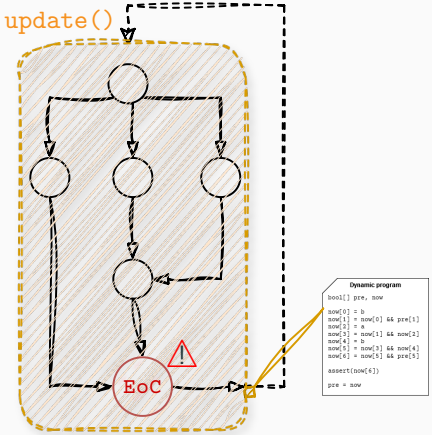
1. Sparse updates based on Cone-of-Influence
   → Uses variable interdependencies
2. State-space reduction on $\mathcal{A}$

Verification Pipeline: 2. Merging

2. Merge $\mathcal{A}$ and $\mathcal{B}$ to combined CFA $\mathcal{C}$ – State-space reduction by abstraction

Key observation: PLC programs do not return intermediate cycle values

! domain-specific assumption



Idea: PLC-update as blackbox update() function

$\rightarrow$ Implicitly transforms φ into *cycle-end* $\rightarrow \varphi$

Property structure restriction

vs

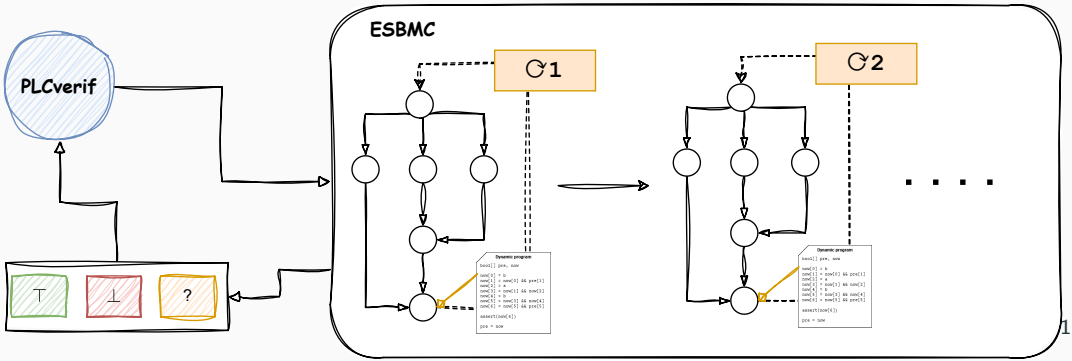
Runtime efficiency

3. Verify merged automaton $\mathcal{C}$

Idea: Hand-over to verifiers and verify assertion over the monitor

Observation: Only PLC-cycle loop is unbounded *! domain-specific*

→ *(i) fully unroll inner loops* and *(ii) perform BMC over PLC-cycle*



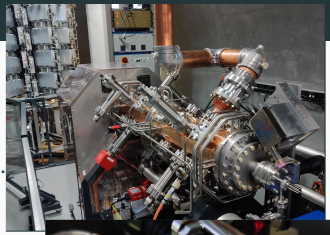
Case Studies: ELISA & UNICOS OnOff Library

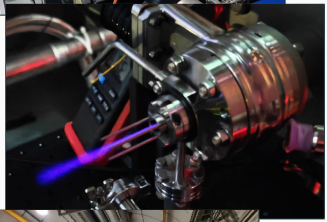
ELISA (Experimental LInear accelerator for Surface Analysis)

- PLC running a safety program: protection against voltage, etc.

UNICOS OnOff Library (industrial control system framework)

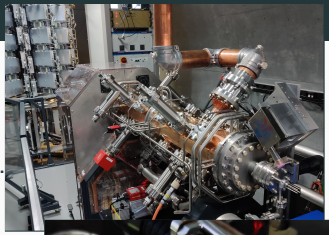
- Deployed in hundreds of industrial installations

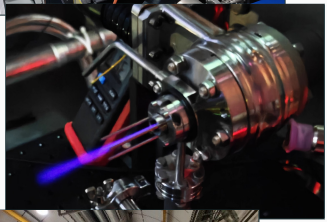




			EL2		OnOff2	
	Tool		time	result	time	result
new!	ESBMC	k-Ind	1.42	$\top$	4.57	$\bot$
		BMC	TO(810)	$\top$	3.21	$\bot$
	NuSMV	BDD	0.91	$\top$	TO	–
		BMC	TO(155)	$\top$	42.83	$\bot$

Case Studies: ELISA & UNICOS OnOff Library





ELISA (Experimental LInear accelerator for Surface Analysis)

- PLC running a safety program: protection against voltage, etc.

UNICOS OnOff Library (industrial control system framework)

- Deployed in hundreds of industrial installations

		EL2		OnOff2	
Tool		time	result	time	result
ESBMC	k-Ind	1.42	$\top$	4.57	$\bot$
	BMC	$\mathrm{TO}(810)$	$\top$	**3.21**	$\bot$
NuSMV	BDD	0.91	$\top$	TO	–
	BMC	$\mathrm{TO}(155)$	$\top$	**42.83**	$\bot$

new!

Conclusion & Future Work

Conclusions

- Extending PLCverif-FRET integration:
  - now any safety PLTL formula with any verification backend
- (PLC) Domain-specific information was used to improve verification performance on two case studies
- The experiments show up to one order of magnitude improved verification times;
  - additionally verifying previously unverifiable properties

Future/Open Work

- Experiment with other monitor types, e.g., NBA-based monitors
- Further evaluate an extension of the monitoring algorithm to metric PLTL for PLC timing modeling