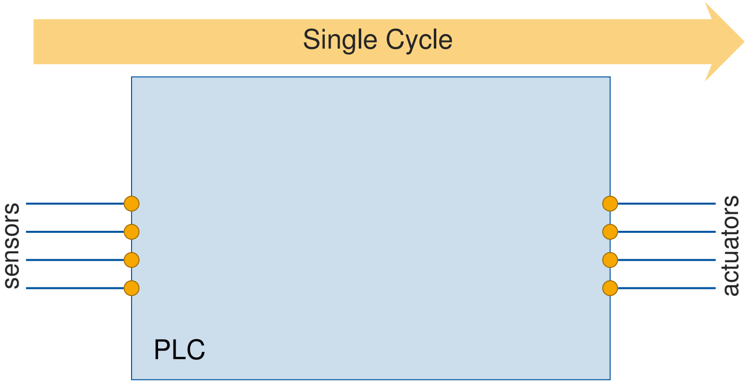
Verification of Reactive Programs from Industrial Automation

Dimitri Bohlender

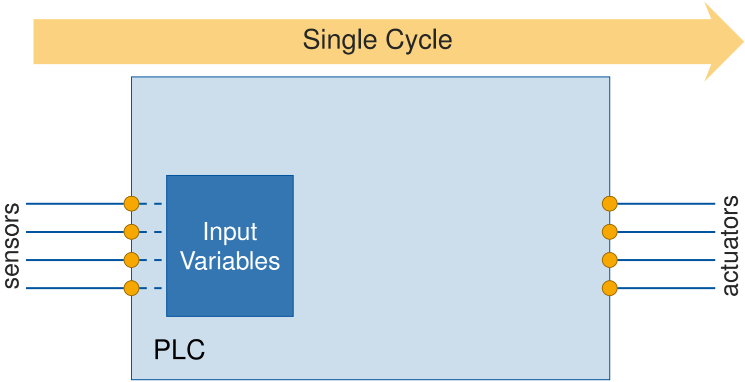
Programmable Logic Controller (PLC)

- Tailored to the domain of industrial automation
- Realise reactive systems, repeatedly executing the same task



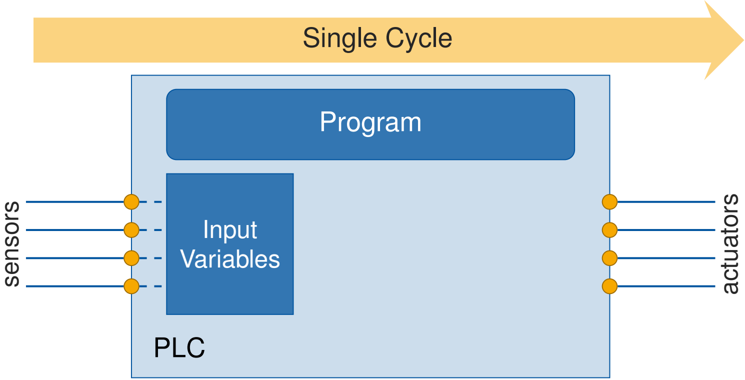
Programmable Logic Controller (PLC)

- Tailored to the domain of industrial automation
- Realise reactive systems, repeatedly executing the same task



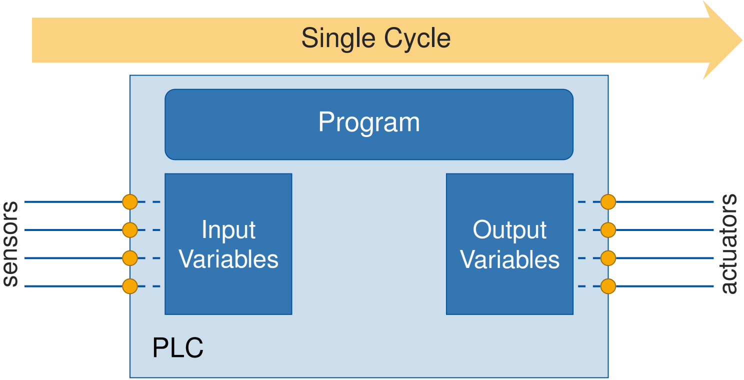
Programmable Logic Controller (PLC)

- Tailored to the domain of industrial automation
- Realise reactive systems, repeatedly executing the same task



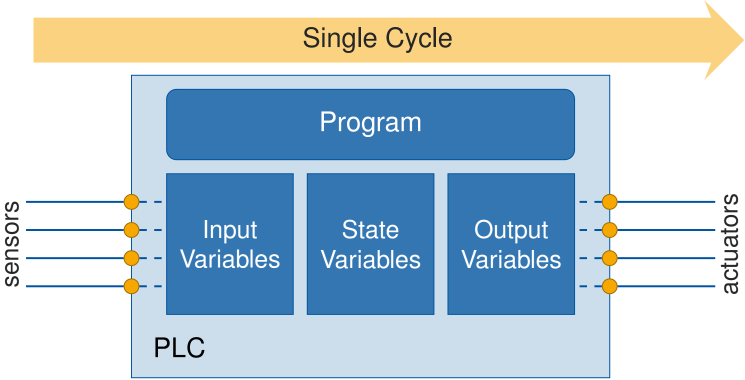
Programmable Logic Controller (PLC)

- Tailored to the domain of industrial automation
- Realise reactive systems, repeatedly executing the same task



Programmable Logic Controller (PLC)

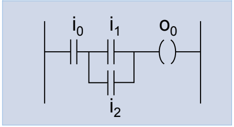
- Tailored to the domain of industrial automation
- Realise reactive systems, repeatedly executing the same task



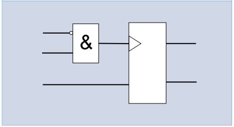
PLC Software

- Programming languages standardised in IEC 61131-3
- Combination of several languages typical

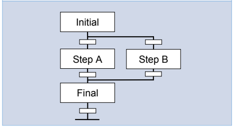
Ladder Diagram



Function Block Diagram



Sequential Function Chart



Instruction List

```
LD          input0
ADD         50
GT          100
JMPC        label
```

Structured Text

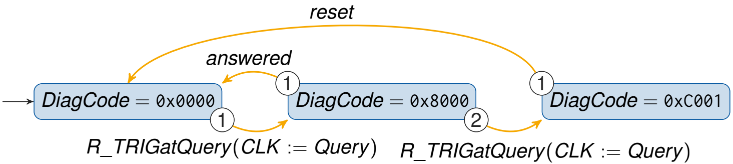
```
IF input0+50 > 100 THEN
        output0 := 1;
ELSE
        output0 := 0;
ENDIF;
```

Vendor Specific Dialects

...

Verification of Domain-Specific Specifications

- Specification automata used by the PLCopen



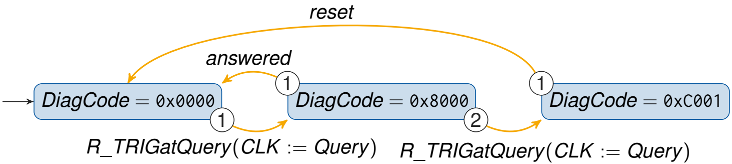
⇒ Characterisation in terms of Constrained Horn-Clauses ✓

- Analysis of Reset-Behaviour
  - Certain variables may retain their value after restart/power cut.
  - Restarting shall not affect the set of observable states, i. e.

$$Reach_{nominal}(s_0) \overset{!}{\supseteq} Reach_{reset}(s_0)$$

Verification of Domain-Specific Specifications

- Specification automata used by the PLCopen



reset

answered

$DiagCode = \texttt{0x0000}$

$DiagCode = \texttt{0x8000}$

$DiagCode = \texttt{0xC001}$

1

1

1

2

$R\_TRIGatQuery(CLK := Query)$

$R\_TRIGatQuery(CLK := Query)$

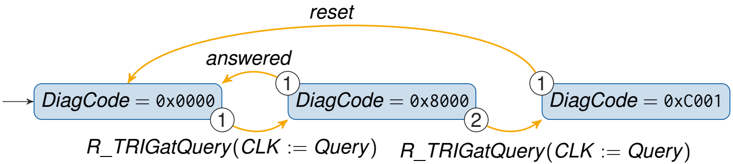
⇒ Characterisation in terms of Constrained Horn-Clauses ✓

- Analysis of Reset-Behaviour
  - Certain variables may retain their value after restart/power cut.
  - Restarting shall not affect the set of observable states, i. e.

$$Reach_{nominal}(s_0) \overset{!}{\supseteq} Reach_{reset}(s_0)$$

Verification of Domain-Specific Specifications

- **Specification automata** used by the PLCopen


reset
answered
$DiagCode = \texttt{0x0000}$
$DiagCode = \texttt{0x8000}$
$DiagCode = \texttt{0xC001}$
1
1
2
1
$R\_TRIGatQuery(CLK := Query)$
$R\_TRIGatQuery(CLK := Query)$

⇒ Characterisation in terms of Constrained Horn-Clauses ✓

- **Analysis of Reset-Behaviour**
  - Certain variables may retain their value after restart/power cut.
  - Restarting shall not affect the set of observable states, i. e.

$$Reach_{nominal}(s_0) \overset{!}{\supseteq} Reach_{reset}(s_0)$$

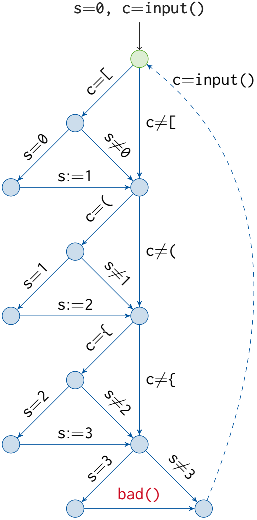
Exploiting Domain-Specifics in Existing Techniques

- Consider bug-finding via symbolic execution

⇒ CFG-based guidance fails

- Bad choices hard to identify (due to cyclicity)

- Implicit state machine (over s)
- Typical pattern in PLC program modules



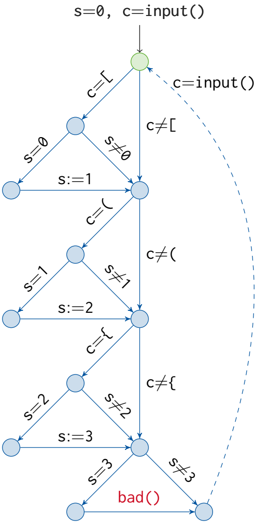
Exploiting Domain-Specifics in Existing Techniques

- Consider bug-finding via symbolic execution

⇒ CFG-based guidance fails

- Bad choices hard to identify (due to cyclicity)

- Implicit state machine (over s)
- Typical pattern in PLC program modules



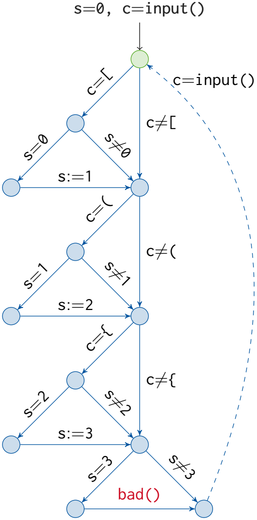
Exploiting Domain-Specifics in Existing Techniques

- Consider bug-finding via symbolic execution

⇒ CFG-based guidance fails

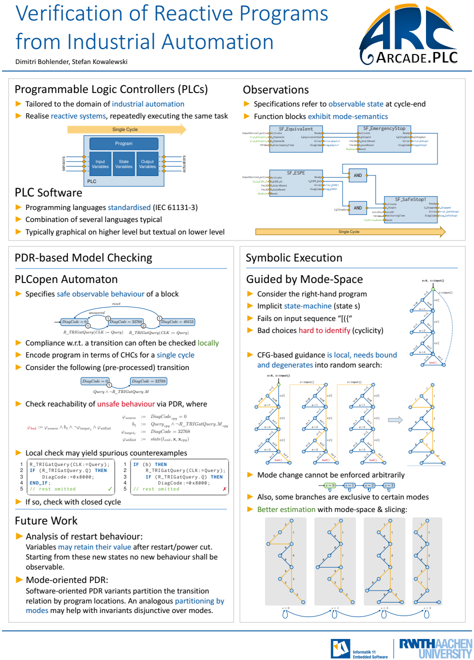
- Bad choices hard to identify (due to cyclicity)

- Implicit state machine (over s)
- Typical pattern in PLC program modules



- Interested? Offended?

⇒ Drop by this poster for more details



Verification of Reactive Programs from Industrial Automation

Dimitri Bohlender, Stefan Kowalewski

ARCADE.PLC

Programmable Logic Controllers (PLCs)

- Tailored to the domain of industrial automation
- Realise reactive systems, repeatedly executing the same task

Single Cycle
Program
sensors
Input Variables
State Variables
Output Variables
actuators
PLC

PLC Software

- Programming languages standardised (IEC 61131-3)
- Combination of several languages typical
- Typically graphical on higher level but textual on lower level

Observations

- Specifications refer to observable state at cycle-end
- Function blocks exhibit mode-semantics

PDR-based Model Checking

PLCopen Automaton

- Specifies safe observable behaviour of a block

- Compliance w.r.t. a transition can often be checked locally
- Encode program in terms of CHCs for a single cycle
- Consider the following (pre-processed) transition

- Check reachability of unsafe behaviour via PDR, where

$$\varphi_{bad} := \varphi_{source} \wedge b_1 \wedge \neg\varphi_{target_1} \wedge \varphi_{atExit}$$

$$\begin{aligned} \varphi_{source} &:= DiagCode_{cpy} = 0 \\ b_1 &:= Query_{cpy} \wedge \neg R\_TRIGatQuery.M_{cpy} \\ \varphi_{target_1} &:= DiagCode = 32768 \\ \varphi_{atExit} &:= state(l_{exit}, \mathbf{x}, \mathbf{x}_{cpy}) \end{aligned}$$

- Local check may yield spurious counterexamples

```
R_TRIGatQuery(CLK:=Query);
IF (R_TRIGatQuery.Q) THEN
    DiagCode:=0x8000;
END_IF;
// rest omitted                ✓
```

```
IF (b) THEN
    R_TRIGatQuery(CLK:=Query);
    IF (R_TRIGatQuery.Q) THEN
        DiagCode:=0x8000;
// rest omitted                ✗
```

- If so, check with closed cycle

Future Work

- Analysis of restart behaviour:
  Variables may retain their value after restart/power cut. Starting from these new states no new behaviour shall be observable.
- Mode-oriented PDR:
  Software-oriented PDR variants partition the transition relation by program locations. An analogous partitioning by modes may help with invariants disjunctive over modes.

Symbolic Execution

Guided by Mode-Space

- Consider the right-hand program
- Implicit state-machine (state s)
- Fails on input sequence “[({“
- Bad choices hard to identify (cyclicity)
- CFG-based guidance is local, needs bound and degenerates into random search:

- Mode change cannot be enforced arbitrarily

- Also, some branches are exclusive to certain modes
- Better estimation with mode-space & slicing:

Informatik 11 Embedded Software

RWTH AACHEN UNIVERSITY