Dániel Darvas (CERN / TU Budapest), István Majzik, Enrique Blanco

Conformance checking for PLC programs and specifications

SIES 2016 conference
23-25/05/2016, Kraków, Poland

Contains joint work of B. Fernández, E. Blanco, S. Bliudze, J.O. Blech, J-C. Tournier, T. Bartha, A. Vörös, I. Majzik, R. Speroni

http://go.cern.ch/9dwT
Context – CERN

- Industrial control systems: vacuum, cryogenics, CV, etc.
- Failures might have *negative impact*
- Increasing complexity without *decreasing quality*?
Context – PLCs at CERN

- Programmable Logic Controllers
  *robust industrial computers*
- Small computing capacity
- Cyclic execution (~100 ms cycle time)
- Special programming languages
- 1000+ PLCs at CERN
Context – Previous work

- **Formal verification** (model checking)
  - Exhaustive analysis
  - Useful and feasible
  - How to get requirements?

- **Formal specification**
  - Unambiguous, detailed definition of behaviour

- **Equivalence checking**
  *Does the implementation match the specification?*
Equivalence checking for PLCs

specification

input sequence

timing

implementation

IF c1 THEN
  v := TRUE;
  i := i + 1;
ELSE
  i := i - 1;
END_IF;

output sequence #1

= ?

output sequence #2
Does implementation match the specification?

− Model checker tools can check the equivalence
− Typically the answer is NOT
− Model checkers can give counterexample:

How to proceed?
Non-important violations
(Acceptable differences)

distract the users,
undermine usability,
hide real problems.

They have to be excluded from the analysis.
Conformance relations

Teaser about the paper:
• More conformance relations
• Formal definition
• Methods defined how to check them
Strict equivalence

- Exact match required
Permissive relation with **fixed delay**

- Constant shift allowed
- Signal shape required to be the same

---

**specified**

**conformant**

**non-conf.**
Permissive relation with variable delay

- **Local differences** are allowed
- Edges can disappear
Permissive relation with variable delay + edge preservation

- Like previous but with edge count preservation
Checking the conformance

PLC code -> Formal verif. model

Formal specification -> Formal verif. model

Reductions

Composite model

Model checker

Satisfied

Not satisfied

Counter-example

Conformance relation

Verification report
Usage examples

– Magnet test safety controller
  • Real, pilot project
  • Equivalence checking showed discrepancies…
  • … even after model checking!

– Re-engineering of our PLC framework
  • Strict equivalence is not required for every output
  • Different conformance levels for each output pair
  • We can focus on the important differences
Summary

- **Equivalence checking**: useful, complements testing and model checking

- **Permissive conformance** relations can make it practical by focusing on real issues

- In our paper: **5 permissive relations**
  - Responding to observed scenarios
  - Formal definition + checking method

- **New relations** can be defined if needed