What is special about PLC software model checking?

THPHA159

D. Darvas, E. Blanco Viñuela, CERN, Geneva, Switzerland
I. Majzik, BME, Budapest, Hungary
daniel.darvas@cern.ch · enrique.blanco@cern.ch · majzik@mit.bme.hu

What is software model checking?

The requirement describes an expected property with mathematical precision

Source code

Requirement

Model checker tool

Model checkers will rigorously check the requirement for each input combination

✓

! Counter-example

Counterexamples demonstrate the violation of the given requirement

Is model checking of PLC programs easier than of general-purpose programs?

Yes!

PLC programs typically have ***simpler control logic***

PLC programs are often ***more critical***, thus there is more motivation

Simpler data structures in PLCs, less data

No!

Different ***background knowledge*** of developers

Many ***different PLC languages*** and vendor-specific flavours

Complex syntax of PLC programming languages

Environment model is crucial for complete program verification

Specialities of PLC software for model checking

Syntax

No precise syntax definition
Systematic experiments are needed to figure out the exact syntax.

Mixing absolute and symbolic addressing permitted
`%M0.0` and `boolVar` may refer to the same bit.

Permissive grammars
`M4.1`, `%MX4.1`, `MX[4,1]` mean the same thing, so do 3 other ways.

Context-dependent grammars
In the STL statement '`A A`', the first `A` is an instruction, the second is a variable.
In case of '`A L 0.0`', `L` refers to a part of the memory, but in '`A L`' the `L` is a variable.

Semantics

No dynamic memory allocation
All memory has to be allocated at compilation-time, which eases the formal verification.

Special semantics
The cyclic execution semantics or the concept of function blocks is different from the semantics of typical general-purpose programming languages.

Imprecise and incomplete semantics definition
Many corner cases or details of semantics (esp. in case of Siemens STL) are not defined.

Timed behaviour
Timing is an essential part of PLC programs, support for timers is crucial.

Low-level memory manipulation
Variables may overlap each other and there are rich data conversion features.

What can we do?

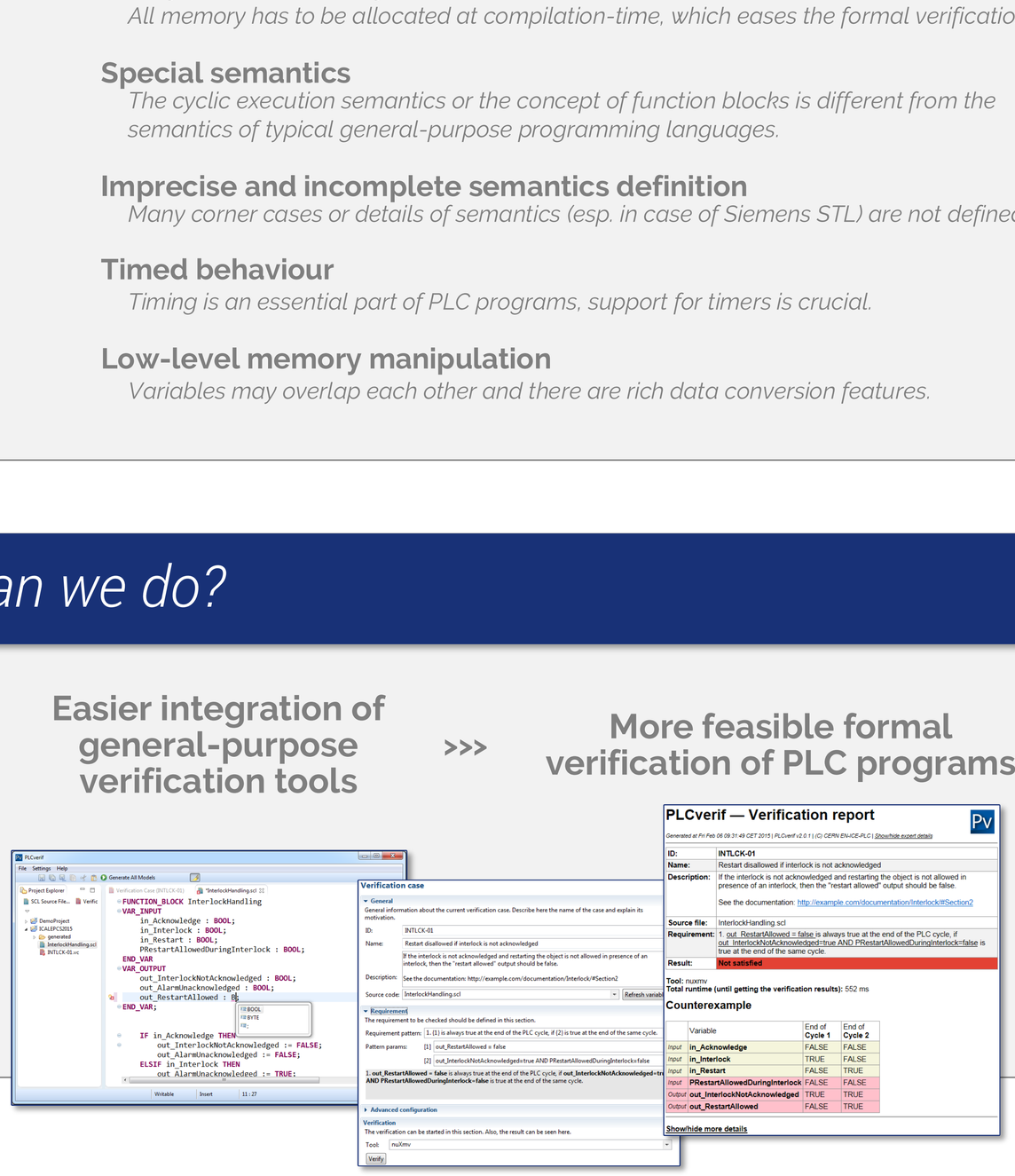
Open PLC programming language infrastructure >>> **Hiding details of PLC languages** >>> **Easier integration of general-purpose verification tools** >>> **More feasible formal verification of PLC programs**

PLCverif

Generic platform for verification of PLC programs

Our response to many of the challenges listed above

Developed at CERN



You can find this poster, the paper and more information at

http://go.cern.ch/9ZjF
http://cern.ch/plcverif

Icons: Google Material design icons, licensed under Apache v2

CERN Beams Department
Industrial Controls and Safety Systems Group (ICS)

ICALEPCS2017
Barcelona · Spain, October 8-13