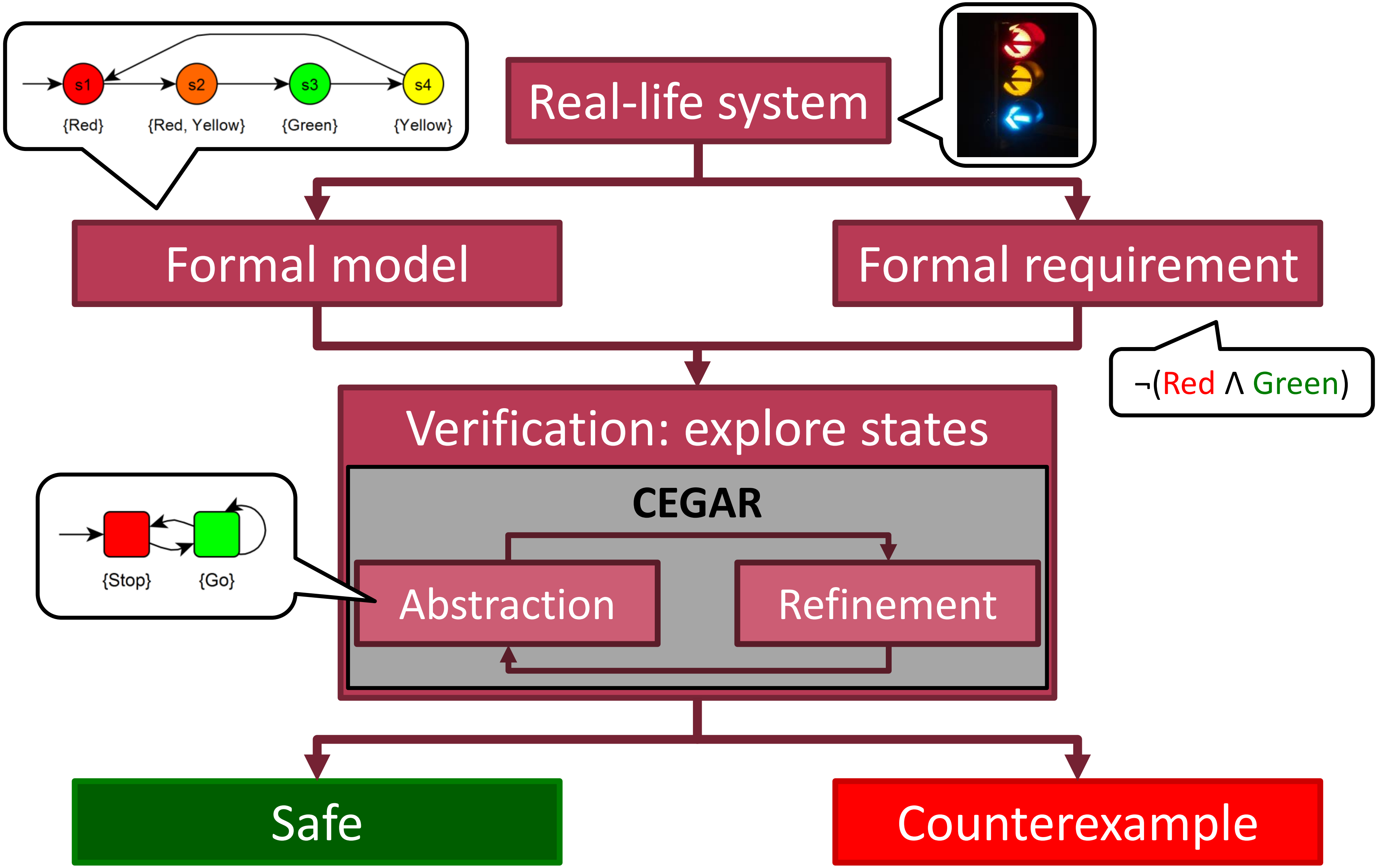
Exploratory Analysis of the Performance of a Configurable CEGAR Framework

Ákos Hajdu[1,2], Zoltán Micskei[1]

[1]Budapest University of Technology and Economics,
Department of Measurement and Information Systems
[2]MTA-BME Lendület Cyber-Physical Systems Research Group

24th Minisymposium of DMIS, 31.01.2017.

Background – Formal verification



Real-life system

Formal model

Formal requirement

¬(Red ∧ Green)

Verification: explore states

CEGAR

Abstraction

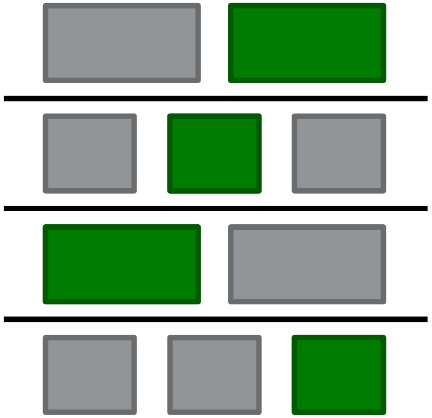
Refinement

Safe

Counterexample

Motivation

- Configurable CEGAR framework
  - Different algorithm configurations
  - Different kinds of models



- Which is the "best" configuration?
  → Preliminary experiment and evaluation

Á. Hajdu, T. Tóth, A. Vörös, and I. Majzik, *"A configurable CEGAR framework with interpolation-based refinements,"* in Formal Techniques for Distributed Objects, Components and Systems, ser. LNCS. Springer, 2016, vol. 9688, pp. 158–174.

Variables of the problem

- Input variables: model
  - System **type** (Hardware/PLC)
  - **Name**
  - Number of **variables**
  - **Size**
- Input variables: configuration
  - **Domain** of abstraction (Pred./Expl.)
  - **Refinement** strategy (Craig itp./Seq. itp./Unsat core)
  - **Initial precision** (Empty/Prop.)
  - **Search** strategy (BFS/DFS)

Variables of the problem

- Output variables
  - Is the model **safe**
  - Execution **time**
  - Number of refinement **iterations**
  - **Size** of the **ARG** (Abstract Reachability Graph)
  - **Depth** of the **ARG**
  - **Length** of the counterexample (**cex**)

Measurement procedure

- **18** input **models**
  - 12 hardware (benchmarks from HWMCC)
  - 6 PLC (from a particle accelerator)
- **20** algorithm **configurations**
- Repeated **5 times**
- **Timeout 480 s**

→ **1800 measurement points**, 1120 successful

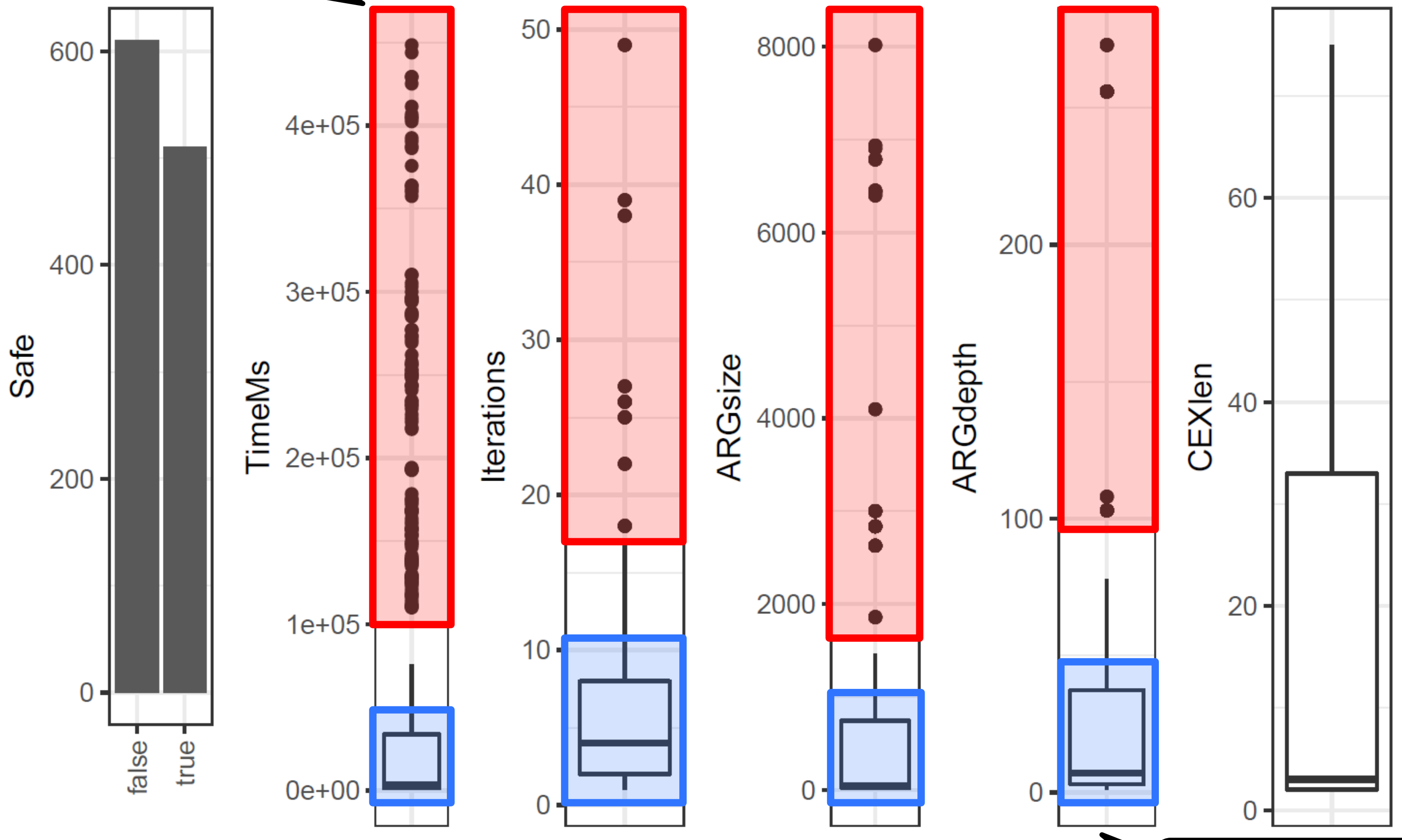
Research questions

- **RQ1**: Overall, high level properties
- **RQ2**: Effect of individual input parameters
- **RQ3**: Influence of input parameters on output

- Validity
  - External: representative input models
  - Internal: repetitions, dedicated machine

RQ1: Overall, high level properties

Many outliers

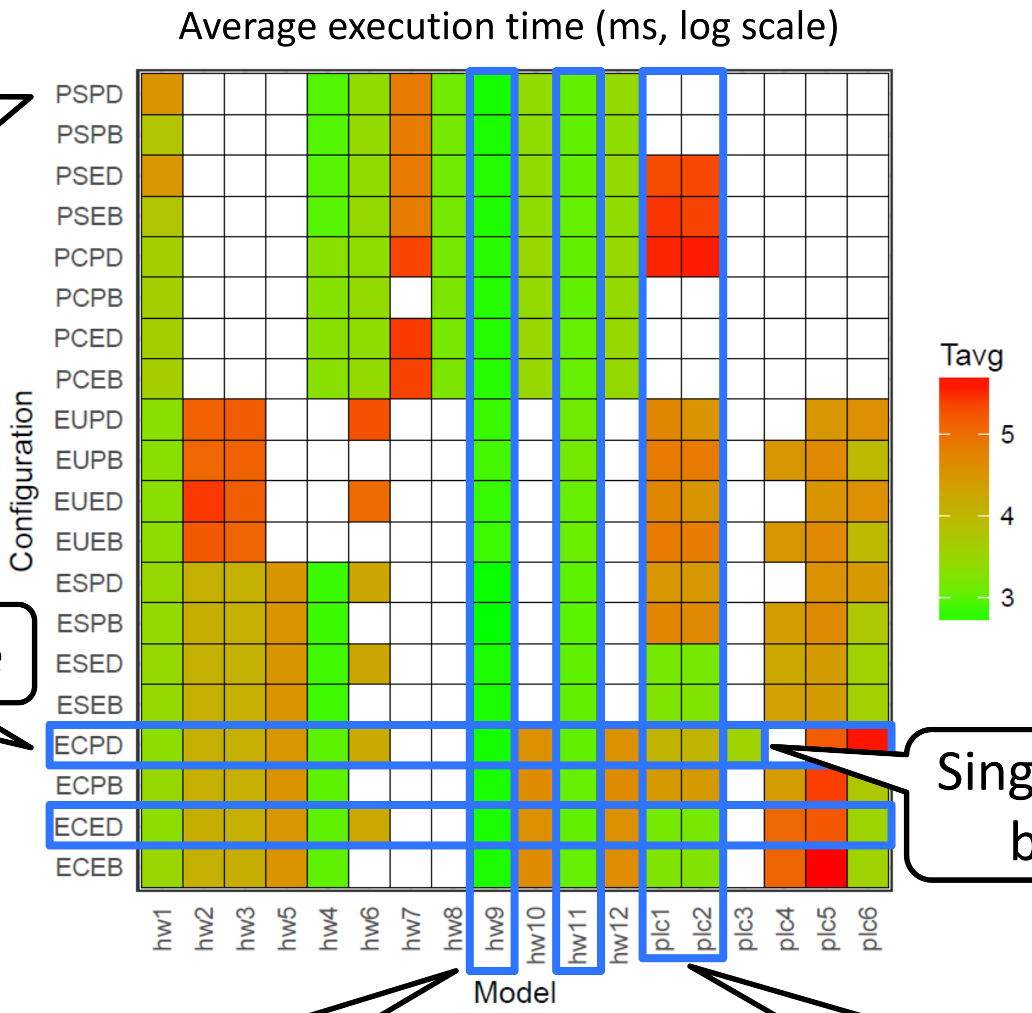


Small IQR

RQ1: Overall, high level properties

Pred
Seq. ltp.
Prop.
DFS

Average execution time (ms, log scale)



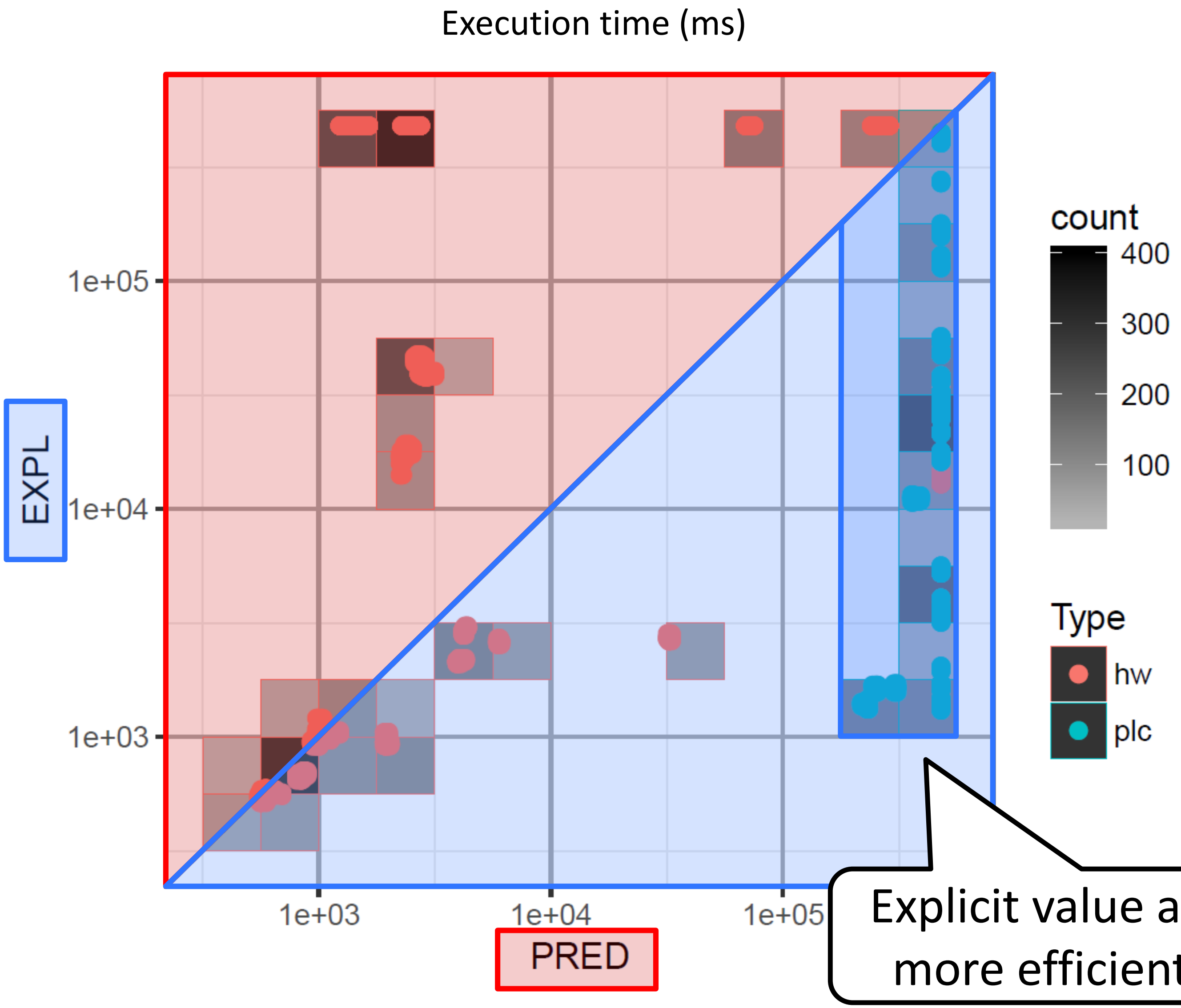
High success rate

Single configuration, but short time

Easy problems

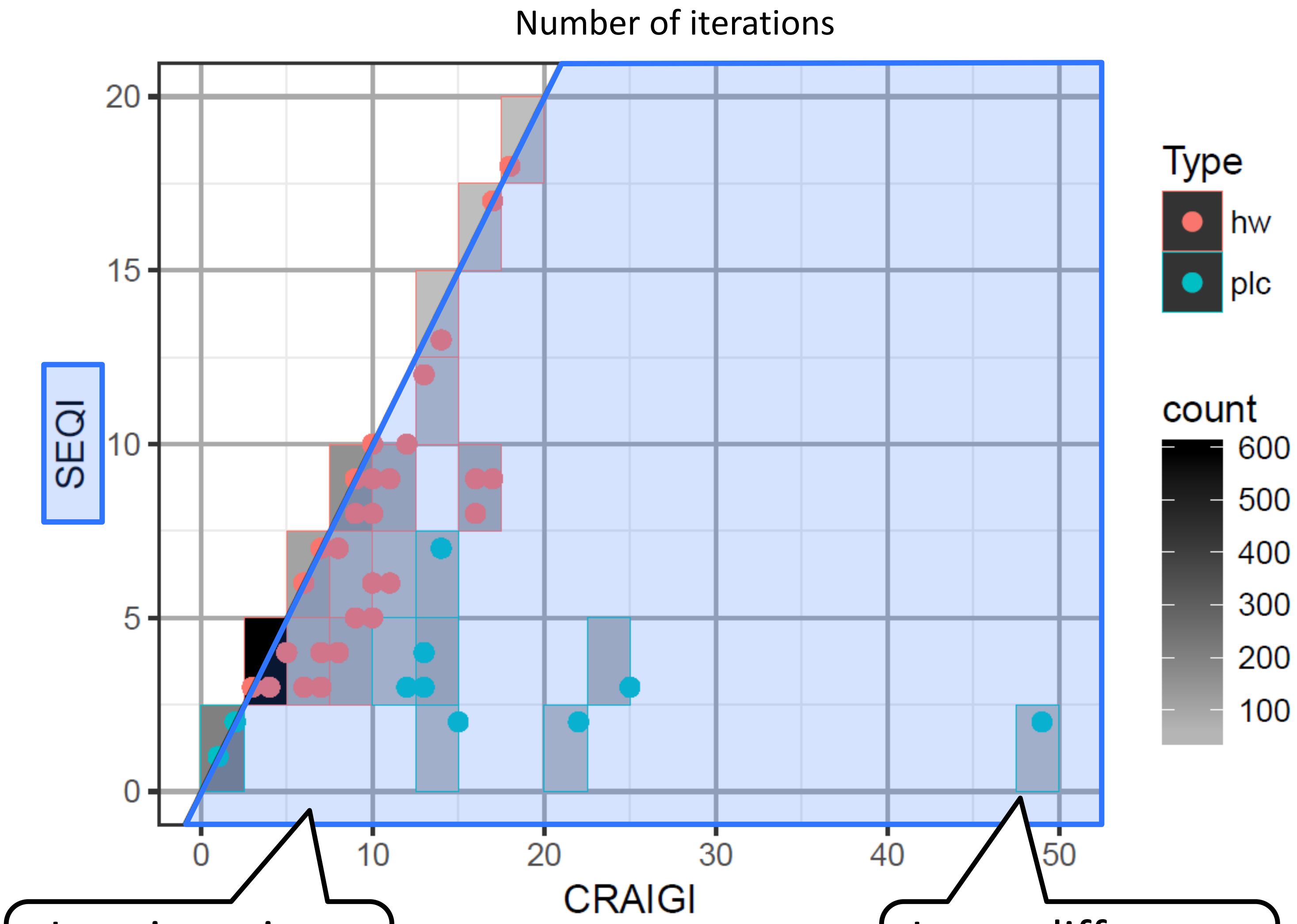
Varying difficulty

RQ2: Effect of individual input parameters



Explicit value abstraction more efficient for PLCs

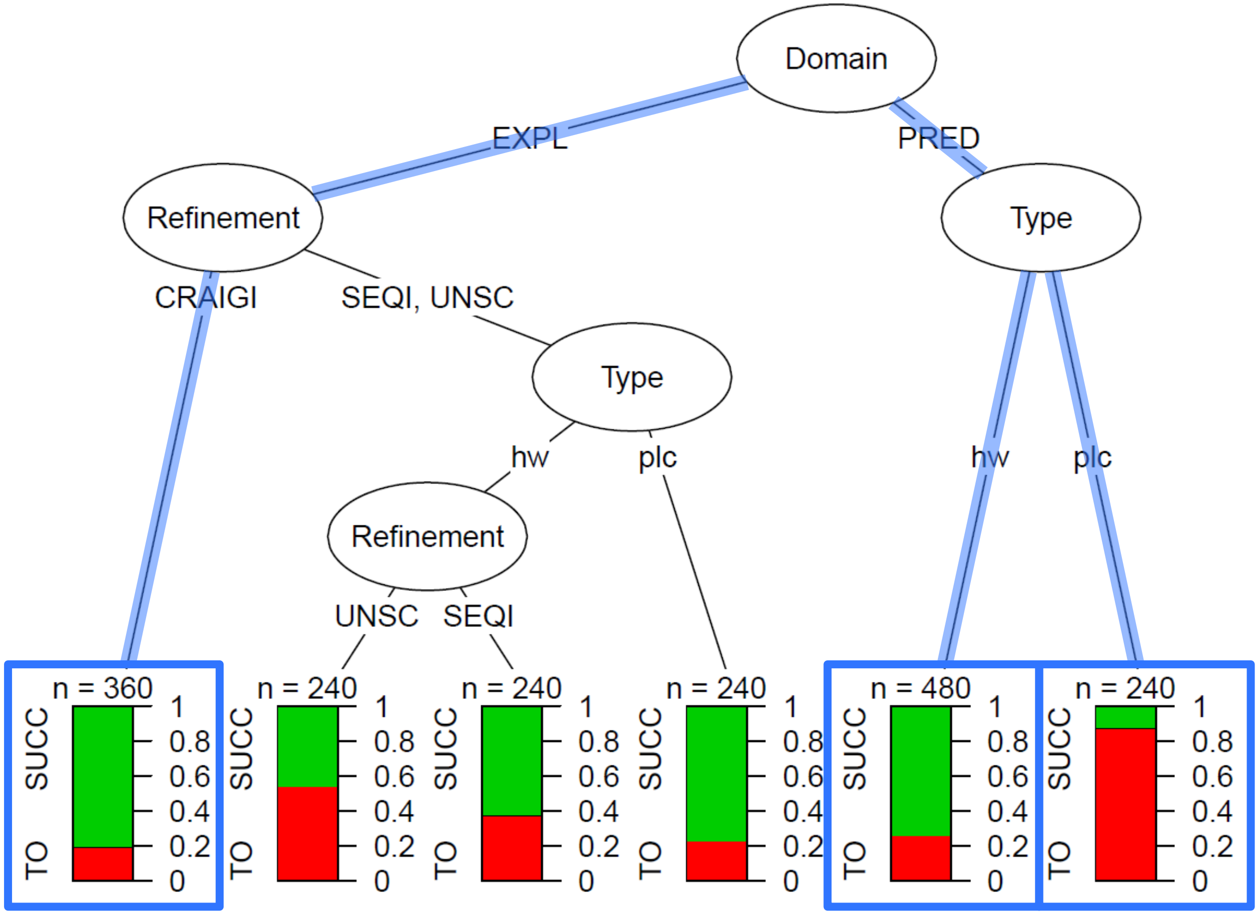
RQ2: Effect of individual input parameters



Less iterations with seq. itp.

Large difference for some PLCs

RQ3: Influence of input parameters on output



Explicit domain with Craig itp. good in general

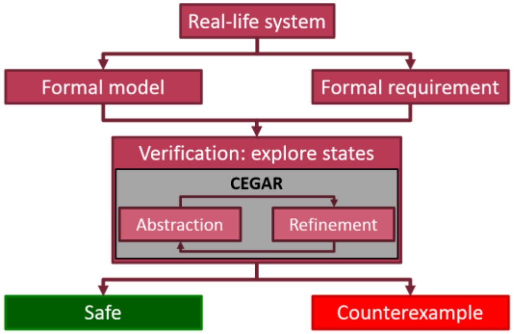
Predicate domain good for hardware

Predicate domain bad for PLCs

Conclusions

- CEGAR framework
  - Different configurations
  - Different systems
- Preliminary results
  - Different configurations are more suitable for different tasks
  - Connections between input and output variables
- Future work
  - Improving the framework
  - Further analysis, heuristics

→ inf.mit.bme.hu/en/members/hajdua



Real-life system

Formal model

Formal requirement

Verification: explore states

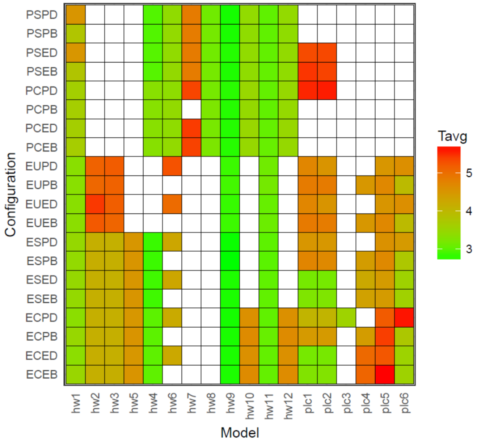
CEGAR

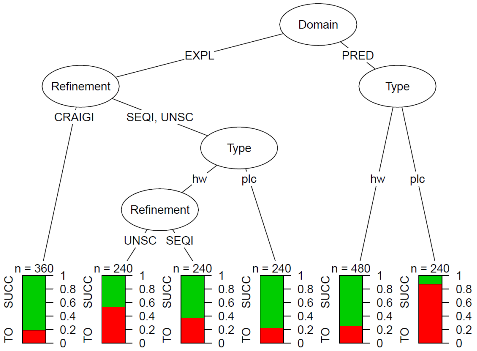
Abstraction

Refinement

Safe

Counterexample


Configuration
PSPD
PSPB
PSED
PSEB
PCPD
PCPB
PCED
PCEB
EUPD
EUPB
EUED
EUEB
ESPD
ESPB
ESED
ESEB
ECPD
ECPB
ECED
ECEB
hw1
hw2
hw3
hw5
hw4
hw6
hw7
hw8
hw9
hw10
hw11
hw12
plc1
plc2
plc3
plc4
plc5
plc6
Model
Tavg


Domain
EXPL
PRED
Refinement
Type
CRAIGI
SEQI, UNSC
Type
hw
plc
hw
plc
Refinement
UNSC
SEQI
n = 360
n = 240
n = 240
n = 240
n = 480
n = 240
SUCC
TO