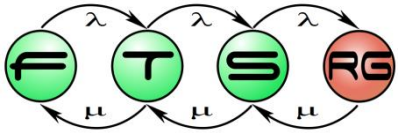
THETA: a Framework for Abstraction Refinement-Based Model Checking

Tamás Tóth[1], Ákos Hajdu[1,2], András Vörös[1,2], Zoltán Micskei[1], István Majzik[1]

[1]Budapest University of Technology and Economics
Department of Measurement and Information Systems

[2]MTA-BME Lendület Cyber-Physical Systems Research Group

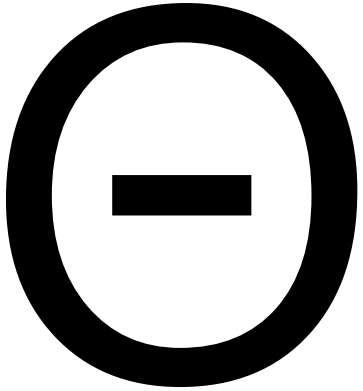
FMCAD 2017, Vienna, Austria, 05.10.2017.



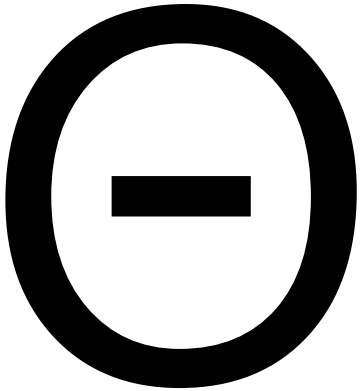
Introduction

- Motivation: a framework for
  - Abstraction refinement-based algorithms
  - Easy development, evaluation and combination
  - Supporting various formalisms
  - Applicable where systems have different aspects (e.g. CPS)

- Our solution: Theta
  - Open source: github.com/FTSRG/theta



Theta – Characteristics



Generic

Various kinds of formal models

Configurable

Different algorithms and strategies

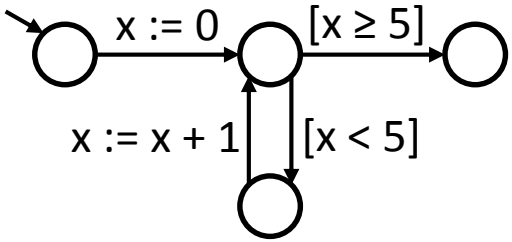
Modular

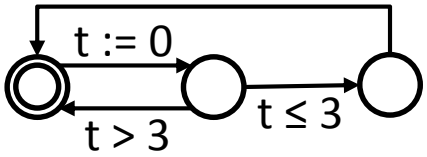
Reusable and combinable modules

Generic – Formalisms

- Symbolic transition systems
  - Low level formalism
  - Based on SMT formulas
- Control flow automata
  - Programs as graphs
  - Edges annotated with statements
- Timed automata
  - Clock variables
  - Operations over clocks
- Support for new formalisms
  - Reusable components, e.g. expressions

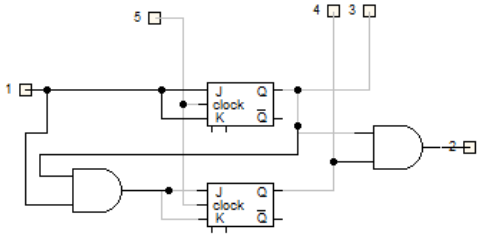
$$I := x = 0 \wedge y = 0$$
$$T := x' = y + 1 \wedge y' = 2 * y$$





Generic – Language frontends

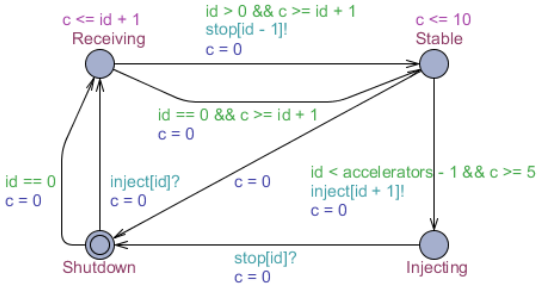
- Symbolic transition systems [FORTE'16]
  - AIGER format
  - Intermediate language for PLCs



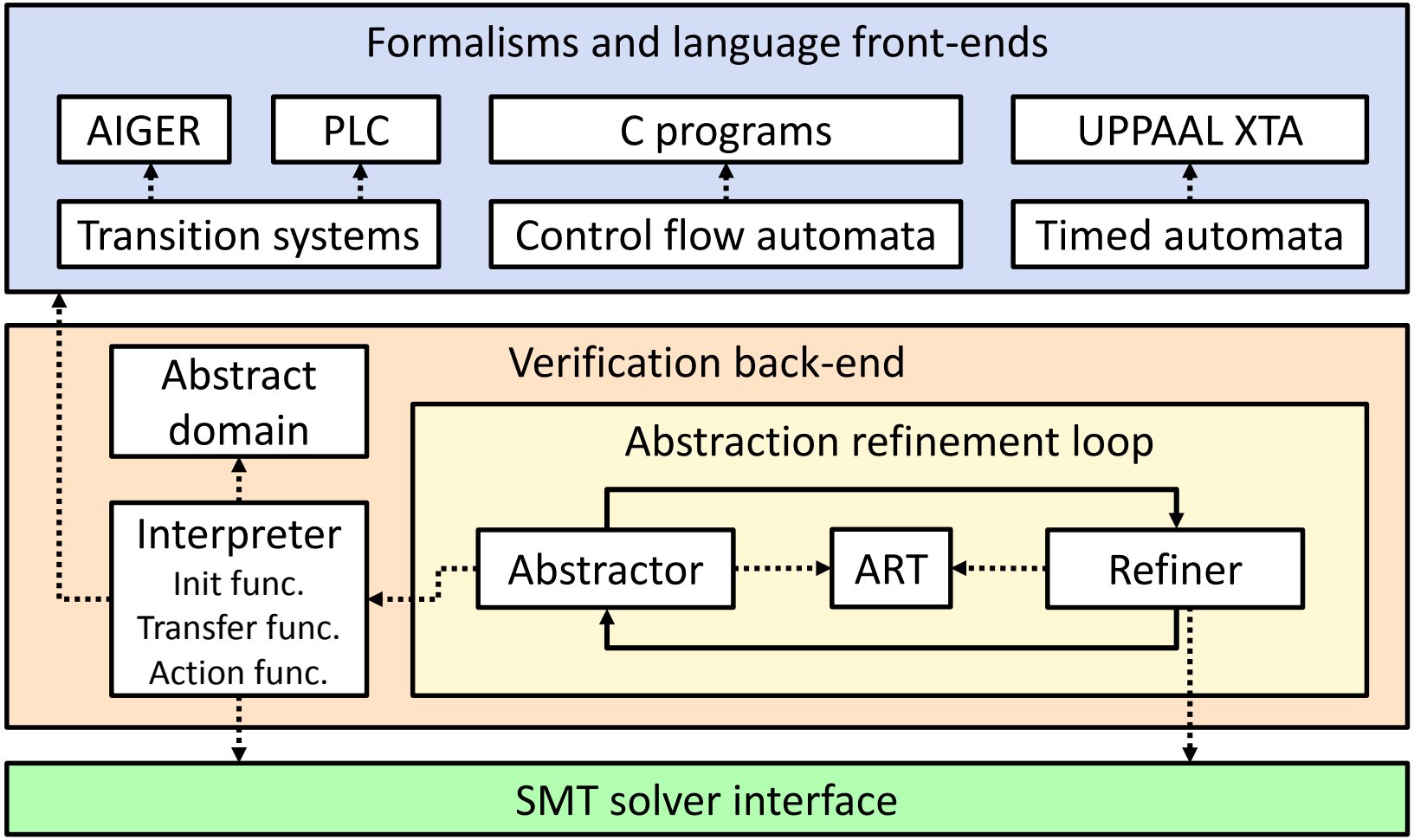
- Control flow automata [VPT'17]
  - Subset of C
  - Size reduction techniques

```
extern int nondet_int();
int main() {
    int a = nondet_int();
    int b = nondet_int();
    int c;
    while (a != 0) {
        c = a;
        a = b % a;
        b = c;
    }
    assert(b != 0);
}
```

- Timed automata [FORMATS'17]
  - UPPAAL XTA

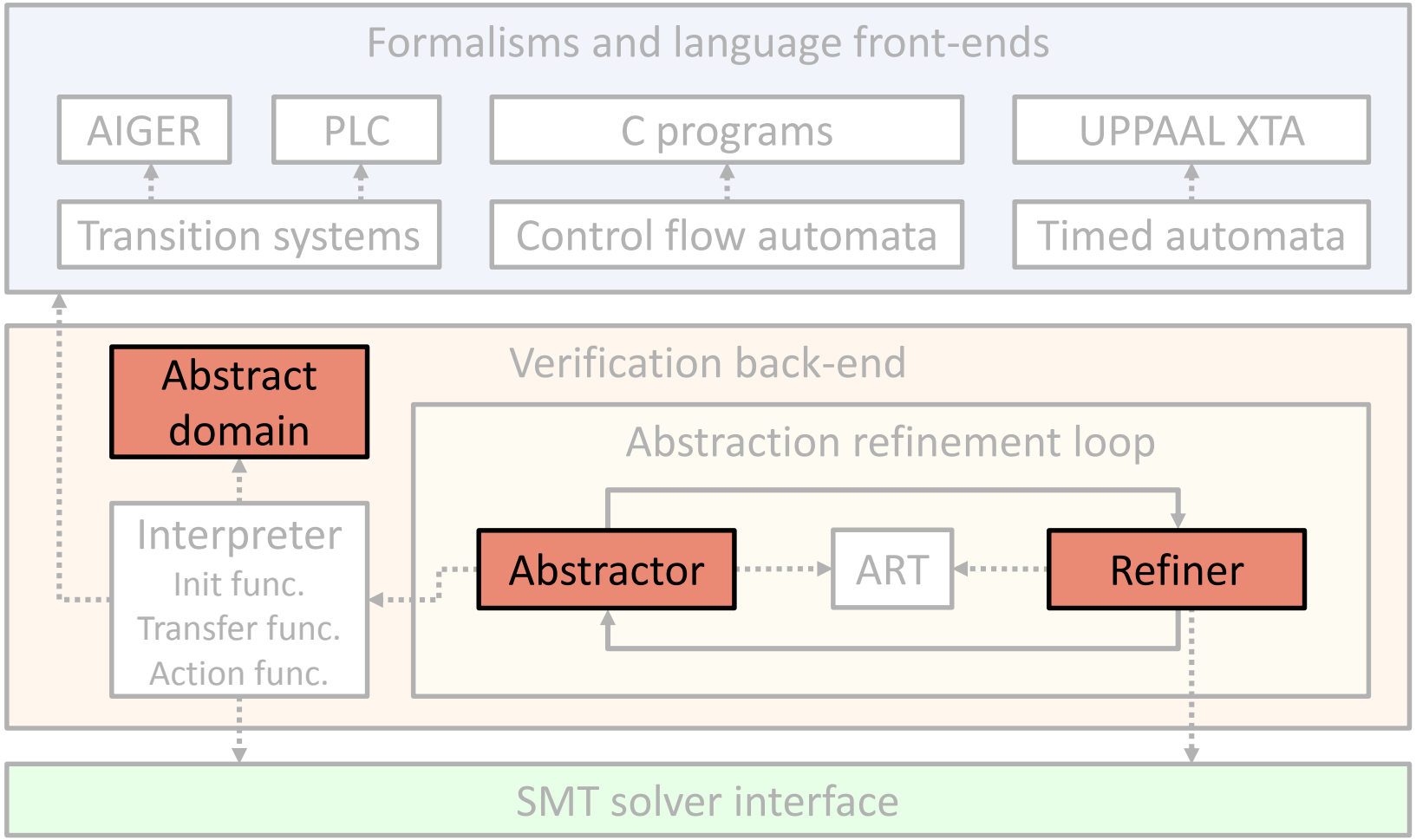


Modular – Architecture



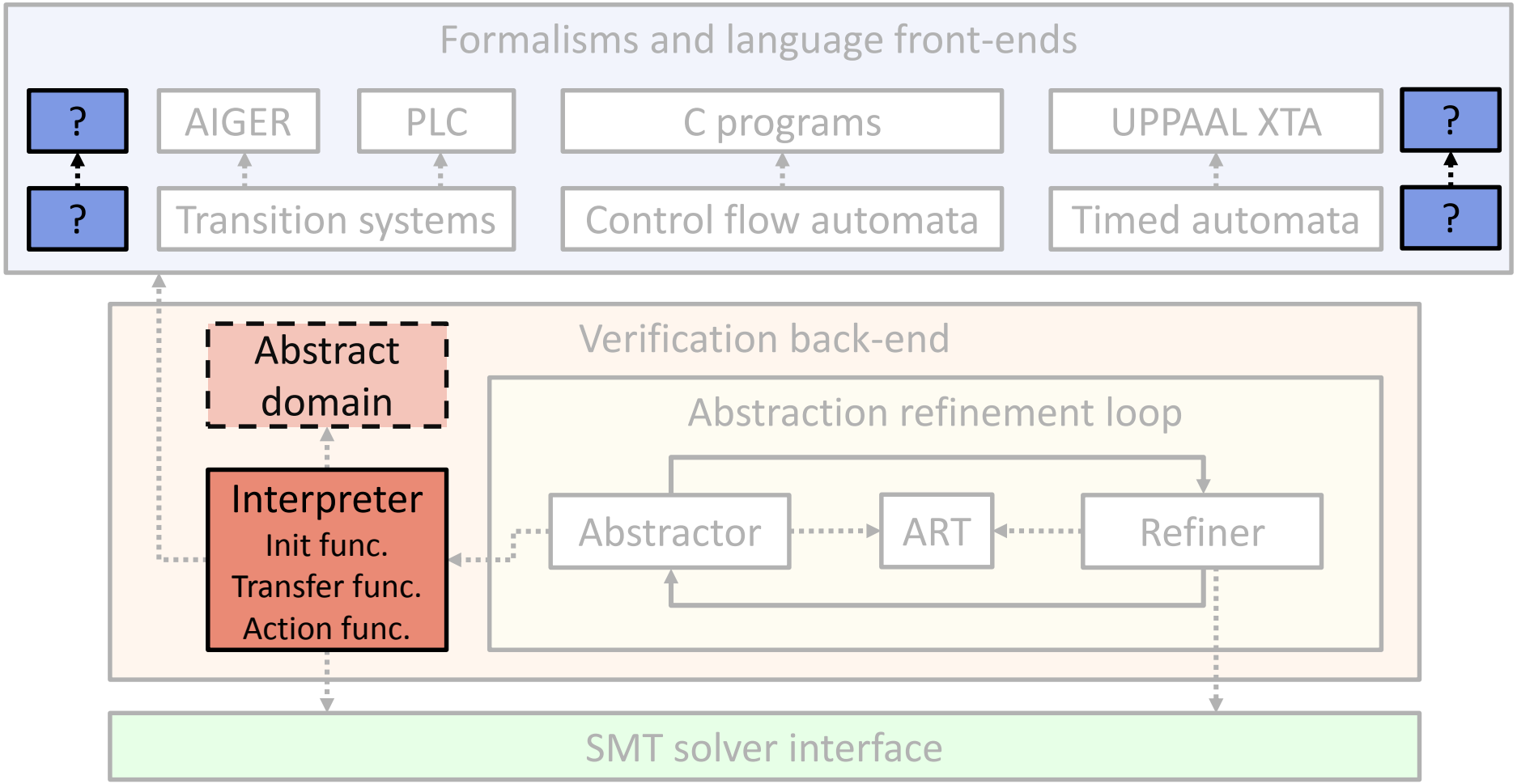
Modular – Extensibility

- New algorithms



Modular – Extensibility

- New formalisms



Configurable – Parameters

Abstract domain
- Predicate
- Explicit value
- Zone
- Location
- Composition

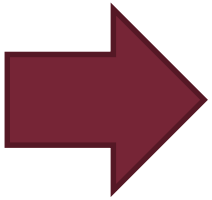
Refinement strategy
- Binary interp. forw.
- Binary interp. backw.
- Sequence interp.
- Unsat core

Search strategy
- BFS
- DFS
- Dist. to error
- Random

Initial precision
- Empty
- Property-based

Precision granularity
- Global
- Local

Predicate split
- Atoms
- Conjuncts
- Whole



78 configs for control flow automata

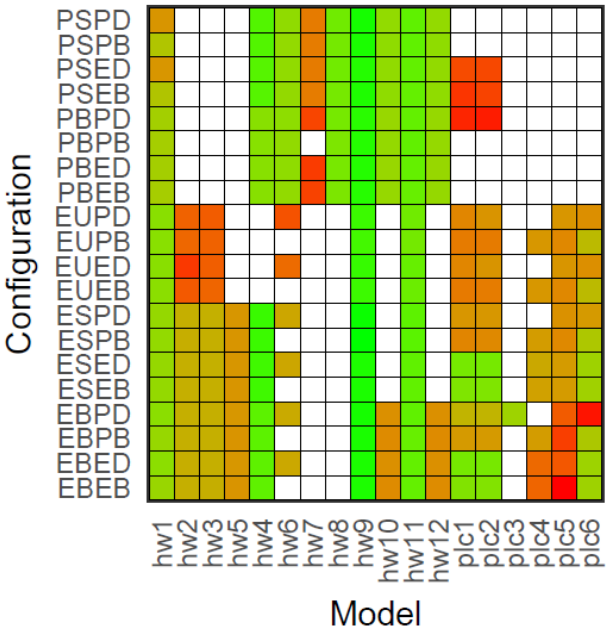
52 configs for transition systems

15 configs for timed automata

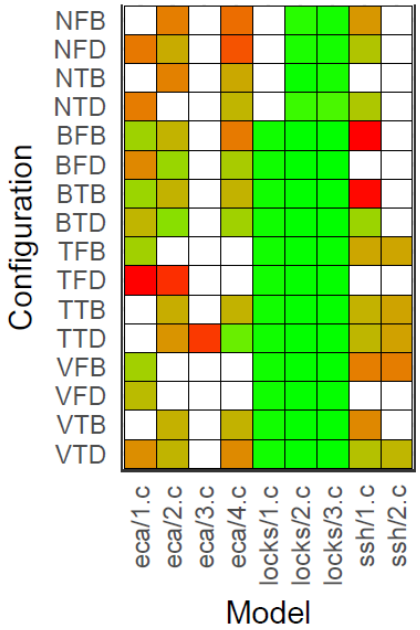
Configurable – Use Cases

- Developing and evaluating new algorithms
  - Extending predicate abstraction with explicit values [FORTE'16]
  - Lazy reachability checking of timed automata [FORMATS'17]
- Diverse results support configurability

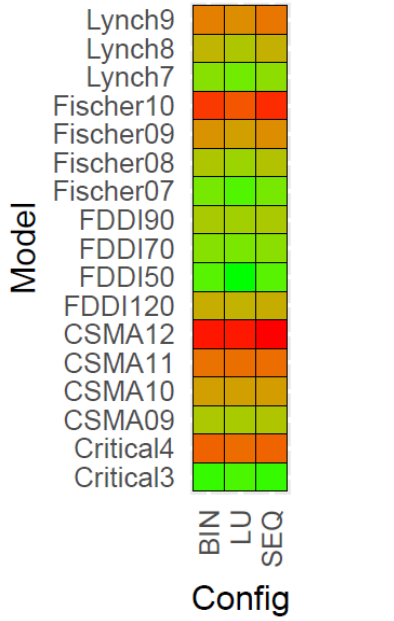
HWMCC & PLC [MiniSym'17]



SV-COMP [VPT'17]



UPPAAL [FORMATS'17]

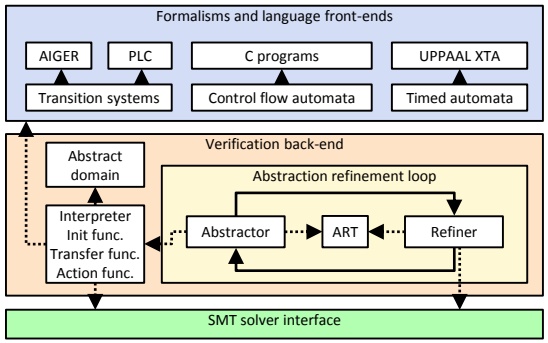


Comparison of execution time in case of different analysis configurations on various models

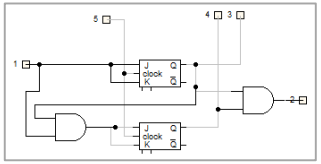
Conclusions

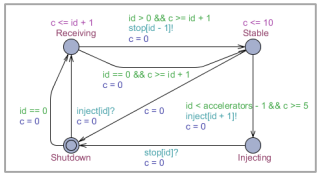
- Theta: Model checking framework
  - Generic, modular, configurable
  - Various formalisms and frontends
  - Abstraction refinement algorithms
- Current and future work
  - Extend the C frontend (LLVM)
  - Experiment with novel algorithms
  - Increase input models in experiments
  - Automatic configuration selection

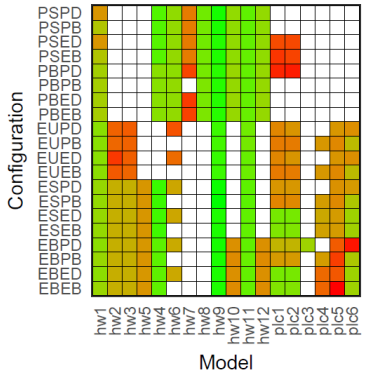
→ github.com/FTSRG/theta




```
extern int nondet_int();

int main() {

  int a = nondet_int();
  int b = nondet_int();
  int c;

  while (a != 0) {
    c = a;
    a = b % a;
    b = c;
  }

  assert(b != 0);

}
```







References

- **[FORTE'16] A Configurable CEGAR Framework with Interpolation-based Refinements**. *Hajdu, Á.; Tóth, T.; Vörös, A.; and Majzik*, I. In Formal Techniques for Distributed Objects, Components and Systems, vol. 9688 of LNCS, pages 158--174. Springer, 2016.
- **[MiniSym'17] Exploratory Analysis of the Performance of a Configurable CEGAR Framework**. *Hajdu, Á.; and Micskei, Z.* In Proceedings of the 24th PhD Mini-Symposium, pages 34--37, 2017. Budapest University of Technology and Economics, Department of Measurement and Information Systems
- **[VPT'17] Towards Evaluating Size Reduction Techniques for Software Model Checking**. *Sallai, Gy.; Hajdu, Á.; Tóth, T.; and Micskei, Z.* In Proceedings of the Fifth International Workshop on Verification and Program Transformation, vol. 253 of EPTCS, pages 75--91. Open Publishing Association, 2017.
- **[FORMATS'17] Lazy Reachability Checking for Timed Automata using Interpolants**. *Tóth, T.; and Majzik, I.* In Formal Modelling and Analysis of Timed Systems, vol. 10419 of LNCS, pages 264--280. Springer, 2017.