Theta: Portfolio of CEGAR-based analyses with dynamic algorithm selection (Competition Contribution)

Zsófia Ádám[1], Levente Bajczi[1], Mihály Dobos-Kovács[1], Ákos Hajdu, Vince Molnár[1]

[1] Budapest University of Technology and Economics, Budapest, Hungary

The research contributions of the authors from the Budapest Univ. of Tech. and Econ. were funded by the EC and NKFIH through the Arrowhead Tools project (EU grant No. 826452, NKFIH grant 2019-2.1.3-NEMZ ECSEL-2019-00003), and by the UNKP-21-2 New National Excellence Program of ITM from the NRDI Fund.

MŰEGYETEM 1782

BME FACULTY OF ELECTRICAL ENGINEERING AND INFORMATICS

ftsrg Critical Systems Research Group

Theta by ftsrg Critical Systems Research Group

- A **generic**, **modular** and **configurable** model checking framework

Theta by ftsrg Critical Systems Research Group

- A **generic**, **modular** and **configurable** model checking framework
- **Abstraction refinement**-based algorithms *(CEGAR)*

Theta by ftsrg Critical Systems Research Group

- A **generic**, **modular** and **configurable** model checking framework
- **Abstraction refinement**-based algorithms *(CEGAR)*
- Supporting different frontends for **C programs,** Statecharts, Petri nets, PLC, AIGER, Timed Automata

Theta – C frontend module

- ANTLR for parsing
- C to **XCFA**
  - eXtended CFA – with processes and procedures

- **Direct** transformation
  - **Simplification passes** on XCFA,
  - instead of using a robust framework, like LLVM
    (see FormaliSE'22 paper „C for yourself")

Theta – New features of the analysis

CEGAR – highly configurable

- Abstract domains,
- Refinements,
- Initial precisions,
- etc.

Analysis of **multithreaded** programs

SMT-LIB support

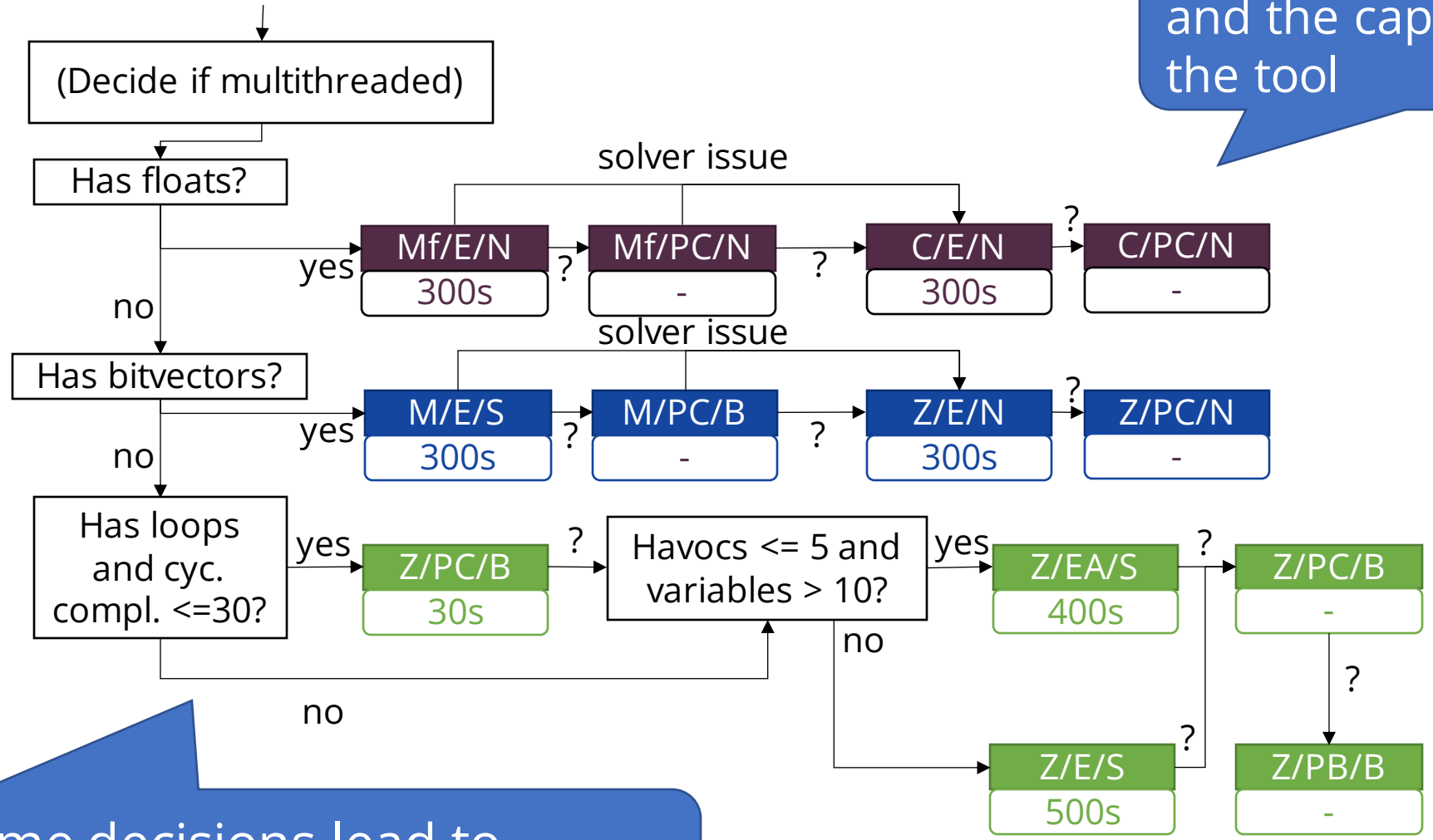
Runtime monitoring, if there is **refinement progress**

Utilizing these features:

Portfolio with algorithm selection and different strategies

Theta - Portfolio

Utilizing configurability and the capabilities of the tool



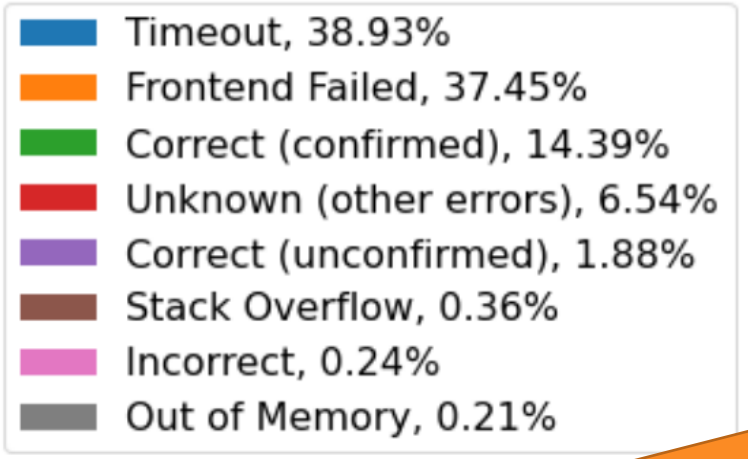
Runtime decisions lead to different configurations

Based on prior experience and earlier benchmarks

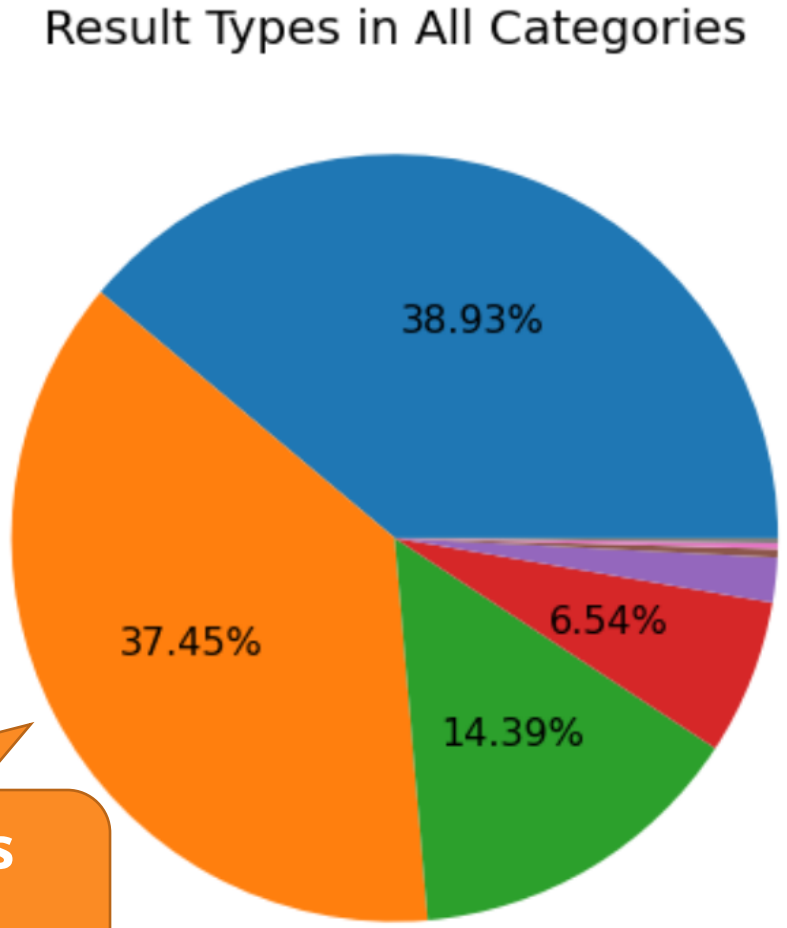
Results on SV-COMP'22

- ReachSafety and ConcurrencySafety
- Whole sub-categories **unsupported** (for now),
- Many **timeouts**,
- Over **14%** of tasks were solved correctly

14.39% Confirmed Correct
1.88% Unconfirmed Correct
0.24% Incorrect



37.45% Frontend Issues
38.93% Timeouts
7.11% Other Issues



Thank you for your attention!

Check out our poster as well!

Theta

https://github.com/ftsrg/theta/

Analysis of **multithreaded** programs

SMT-LIB support

Runtime monitoring, if there is **refinement progress**