Overview of Current Work

Martin Spießl

LMU Munich, Germany

Outline

- Current Work
  - Distribution of Verification Effort
  - Bridging Automatic and Interactive Verification

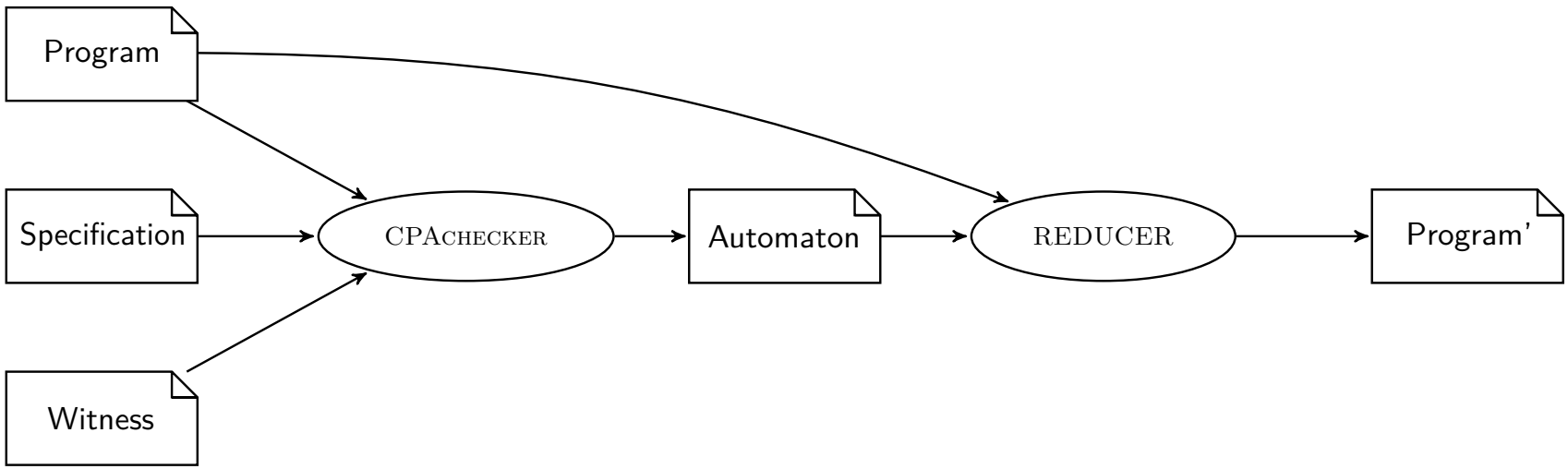
Outline

- Current Work
  - Distribution of Verification Effort
  - Bridging Automatic and Interactive Verification

Outline

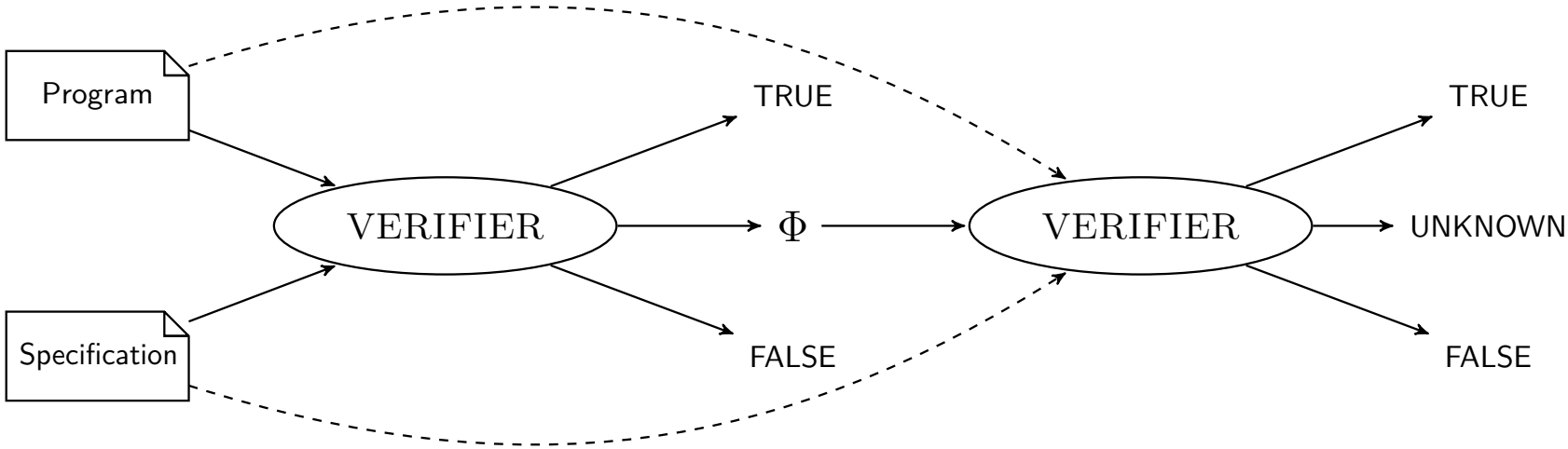
- Current Work
  - Distribution of Verification Effort
  - Bridging Automatic and Interactive Verification

Reducing Correctness Witness Validation to Reachability



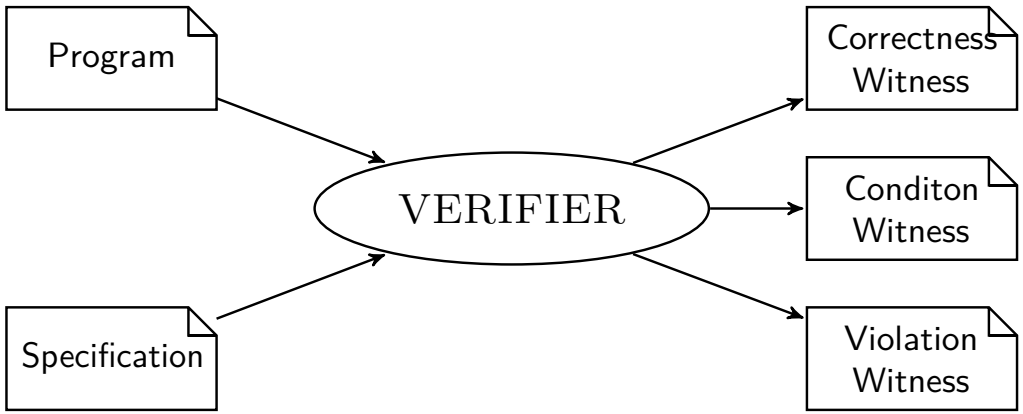
Current joint work with Maximilian Wiesholler (Bachelor student)

Conditional Model Checking[5]



- Cooperation of verification approaches can improve performance
- no standardized exchange format for conditions (yet)
- inherently sequential
- Not clear when best to switch between verifiers

Standardize Condition Format



- Exchange formats for verdicts TRUE and FALSE already exists with correctness and violation witnesses
- Should be easily extendable to encode CMC conditions
- witness format is already supported by many verifiers (due to SV-COMP)
- $\Rightarrow$ easier adaption

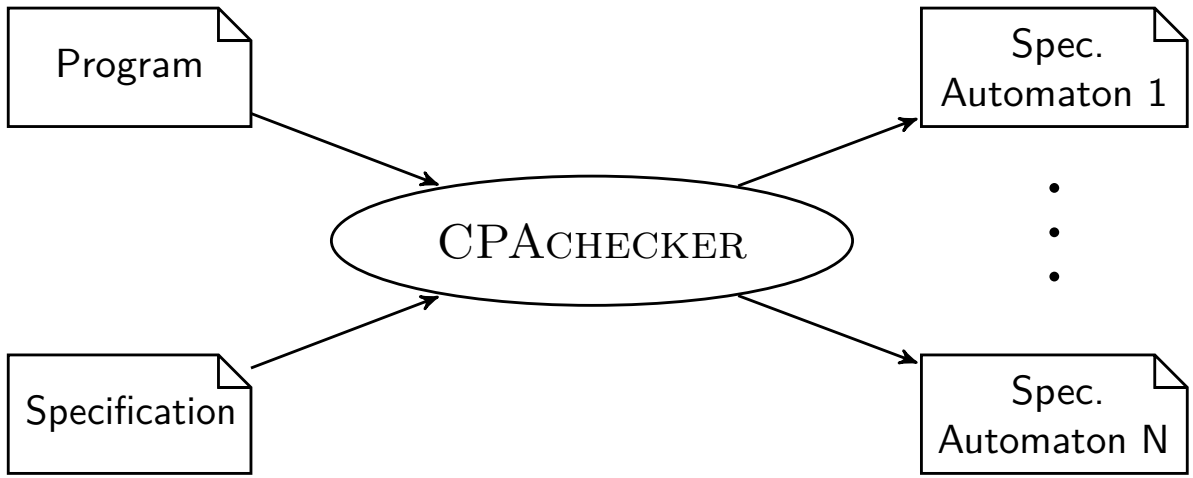
Joint work with Roman Hosseini (bachelor student)

Test Vectors To Violation Witnesses

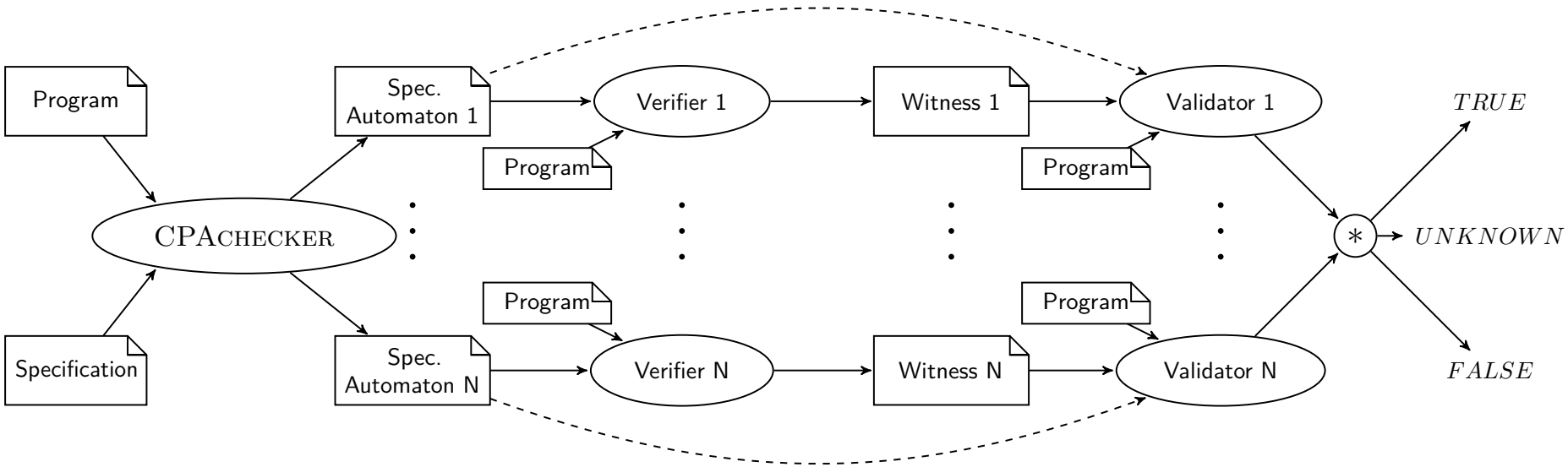
- Test witnesses code test vectors in an XML format
- We could easily extend the violation-witness format such that we can transform test witnesses into true violation witnesses

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE testcase PUBLIC "+//IDN sosy-lab.org//DTD test-forbla..
<testcase>
<input variable="x" type="int">1023</input>
<input variable="y" type="unsigned char">254</input>
</testcase>
```

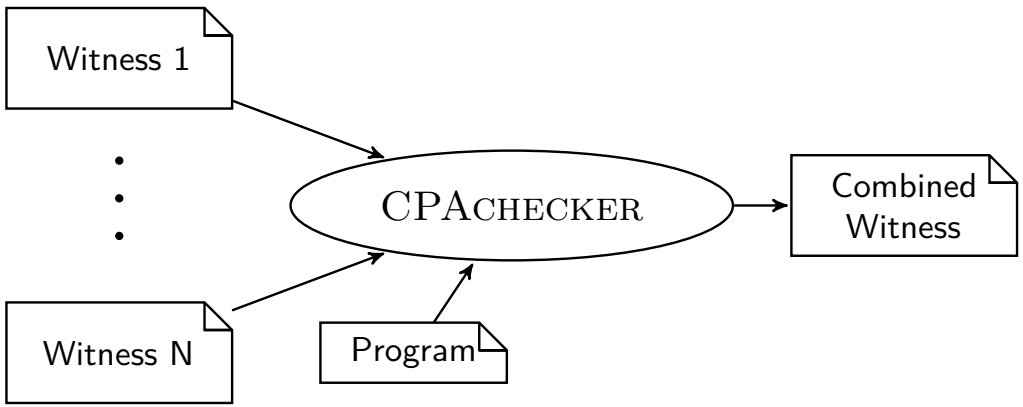
Automatic Property Splitting Using CPAchecker



Independent Verif. and Valid. of Subproperties



Witnesses can be combined and checked against original Specification



- Only works if all witnesses refer to same program
- Should work for witnesses from different verifiers

Outline

- Current Work
  - Distribution of Verification Effort
  - Bridging Automatic and Interactive Verification

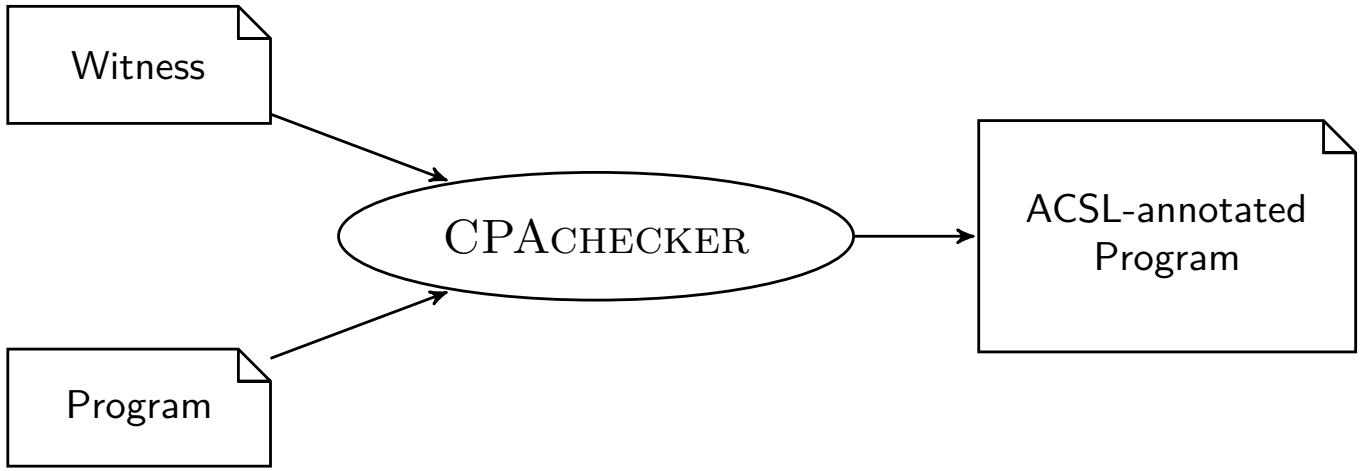
Bridging Automatic and Interactive Verification

Tools commonly used in the VerifyThis Competition

- CIVL
- Dafny
- Frama-C
- KeY
- KIV
- mCRL2
- Viper
- VerCors
- VeriFast
- Why3

- Can we exchange information with Automatic Software Verifiers like CPAchecker?
- Can we transform Automatic Software Verifiers into Interactive ones?

Information Transfer using ACSL and Witnesses



- Frama-C uses ACSL annotations in the source to store invariants and contracts[6]
- We can convert from witnesses into ACSL annotations (and back)
- Can be done for all verifiers that support witnesses, i.e., participants of SV-COMP

[1] Evren Ermis, Jochen Hoenicke, and Andreas Podelski.
Splitting via interpolants.
In *Proc. VMCAI*, LNCS 7148, pages 186–201. Springer, 2012.

[2] Ingo Brückner, Klaus Dräger, Bernd Finkbeiner, and Heike Wehrheim.
Slicing abstractions.
Fundam. Inform., 89(4):369–392, 2008.

[3] D. Beyer, A. Cimatti, A. Griggio, M. E. Keremoglu, and R. Sebastiani.
Software model checking via large-block encoding.
In *Proc. FMCAD*, pages 25–32. IEEE, 2009.

[4] Dirk Beyer, Marie-Christine Jakobs, Thomas Lemberger, and Heike Wehrheim.
Reducer-based construction of conditional verifiers.

In *Proceedings of the 40th International Conference on Software Engineering*, pages 1182–1193. ACM, 2018.

[5] D. Beyer, T. A. Henzinger, M. E. Keremoglu, and P. Wendler.
Conditional model checking: A technique to pass information between verifiers.
In *Proc. FSE*. ACM, 2012.

[6] Patrick Baudin, Pascal Cuoq, Jean-Christophe Filliâtre, Claude Marché, Benjamin Monate, Yannick Moy, and Virgile Prevosto.
ACSL: ANSI/ISO C Specification Language Version 1.10.
`http://frama-c.com/download/acsl.pdf`.