Accurate and Extensible Symbolic Execution of Binary Code based on Formal ISA Semantics

Sören Tempel, Tobias Brandt, Christoph Lüth, Christian Dietrich, Rolf Drechsler

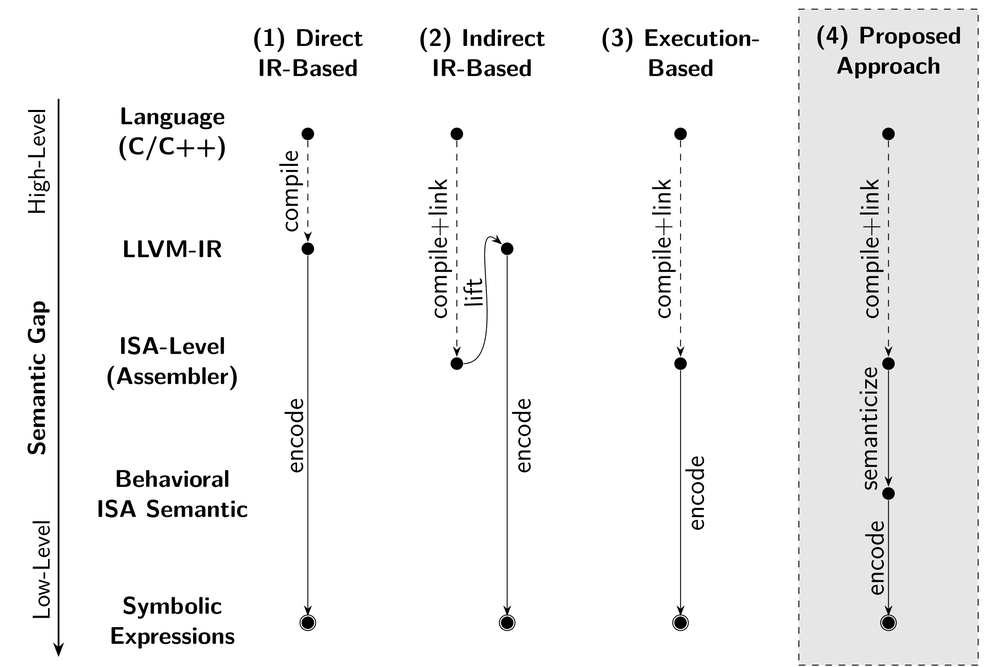
Motivation

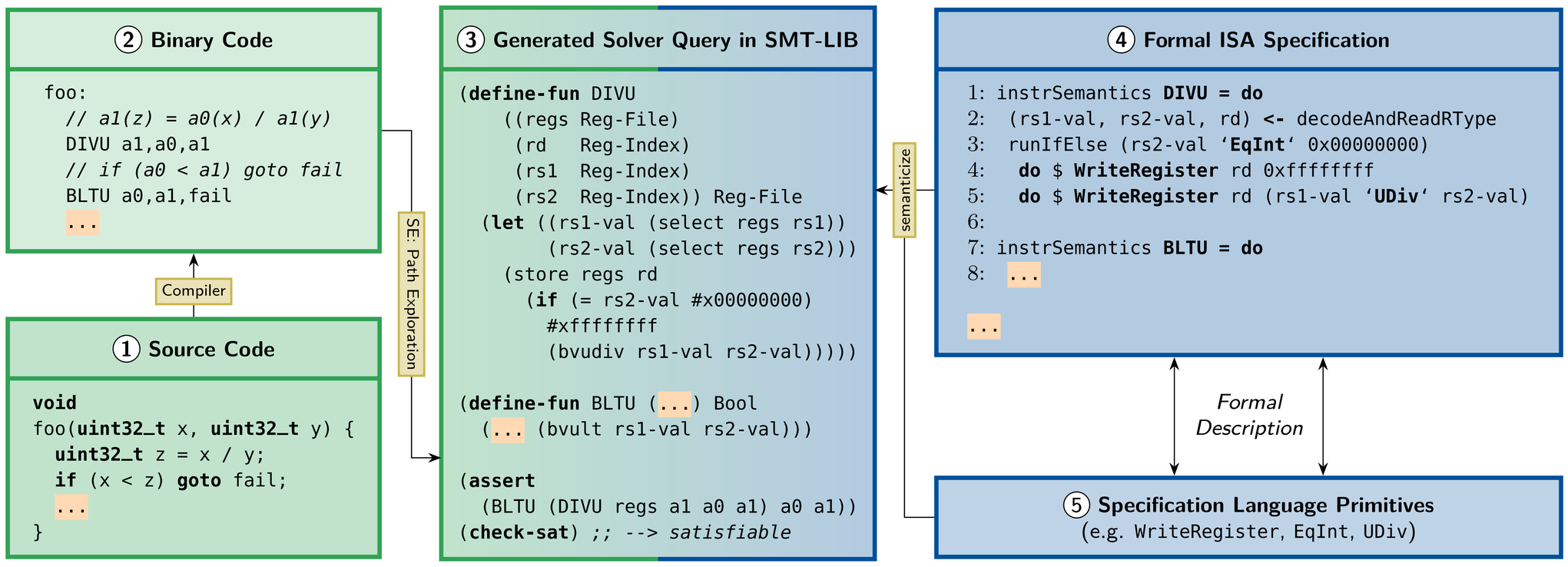
- **Goal:** Employ symbolic execution for testing binary code to ensure that we test the software exactly as it is deployed later on.
- **Challenges:** Requires symbolic software interpretation, i.e. a correct symbolic ISA implementation which must be extensible.
- **Approach:** Utilize on executable formal ISA specification for symbolic execution of binary code targeting this formalized ISA.

Contributions

- BinSym, an SE engine for RISC-V binary code based on the executable formal ISA model provided by prior work on LibRISCV.
- Case study showing its extensibility and a comparison with prior work empirically investigating correctness and performance.

Prior Work

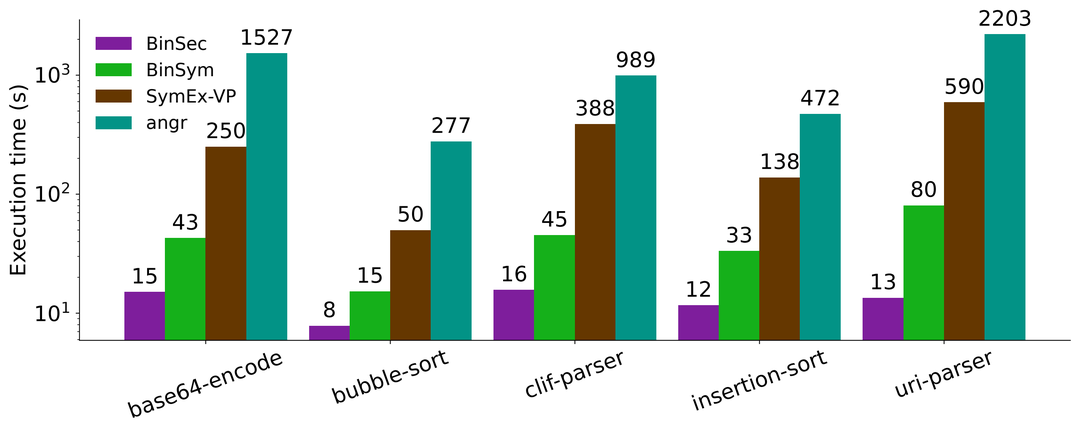




Evaluation

Comparison with prior work on symbolic RV32 binary execution:

- Found 5 previously unknown bugs in the RISC-V lifter of angr
- We further achieve competitive symbolic execution performance



Conclusion

Key Insights:

- **Complexity Reduction:** Executable formal semantics reduce the manual implementation effort and hence the margin for errors.
- **Extensibility:** Formal semantics ease extending the analysis to additional custom instructions and aid design space exploration.
- **Execution Performance:** The symbolic execution speed of our prototype implementation is competitive with existing work.

More Information:

- The RISC-V prototype implementation of our approach is written in 1000 LOC in Haskell and fully open source (scan the QR code!)
- Artifacts for our evaluation are available on Zenodo

Technische Universität Braunschweig

dfki ai Deutsches Forschungszentrum für Künstliche Intelligenz
German Research Center for Artificial Intelligence

University of Bremen