Certifying and Validating Verification

Certifying HW Verifier
Witness Translator

Our Motivation
- Explainable and trustworthy HW verification (HV)
- SW verification (SV) techniques for HW

Our Contributions
- A certifying HW framework using SV techniques
- A translator from SW witnesses to HW witnesses
- A witness validator for the Btor2 HW modeling language [6]
- Complementing HV with certified results from SV

HW-to-SW Translation via Btor2C [1]

Summary of Experimental Results

On 758 safe and 456 unsafe Btor2 verification tasks, Btor2-Cert achieved:
- Translation of all violation and 97% correctness witnesses,
- Effective and efficient validation vs. compared validators, e.g., LIV [4] and CPA-w2r [3], and
- Certified bugs in 8% of the unsafe tasks with CBMC [5] that HV overlooked

References


Try Btor2-Cert!
CPV: A Circuit-Based Program Verifier

Po-Chun Chien and Nian-Ze Lee

{po-chun.chien, nian-ze.lee}@sosy.ifi.lmu.de

Motivation

HWMCC [5]
(Input: Btor2 circuit)

ABC [7], AVR [9], ...

Applicable?

SV-COMP [1]
(Input: C program)

Software Architecture

ReachSafety property

Kratos2 [10]

Btor2 [13]

Btor2Aiger [12]

C prog.

Instrumentor

Instrumented C prog.

Software

witness [3]

Witness

translator

Btor2

witness [19]

Verdict

Circuit

AVR

KI

AVR

PDR

ABC

IMC

ABC

PDR

AVR

BMC

if Btor2-to-Aiger translation succeeds

Try CPV!

Evaluation Results at SV-COMP 2024

6th, 3rd, and 2nd place in ReachSafety, ReachSafety-ECA, ReachSafety-Hardware, respectively

Summary

- It is feasible to utilize sequential circuits as intermediate representations for software verification
- CPV can employ different hardware verifiers as the backend
- CPV competed well against other mature verifiers in SV-COMP
- Future work:
  - Support more verification properties (e.g., no-overflow and termination)
  - Export correctness witnesses
  - Incorporate more backend verifiers
  - Apply circuit optimization to improve the performance of verification

References

[1] Beyer, D.: State of the art in software verification and
    TACAS, pp. 561–579. LNCS 13243 (2022)
    Models and Verification, Johannes Kepler University (2007)
    24–40. LNCS 6174 (2010)
    FMCAD, pp. 125–134 (2011)
    (2020)
    423–436 (2023)

Artifact DOI: 10.5281/zenodo.10063681

Motivation

HWMCC [5]
(Input: Btor2 circuit)

ABC [7], AVR [9], ...

Applicable?

SV-COMP [1]
(Input: C program)

Software Architecture

ReachSafety property

Kratos2 [10]

Btor2 [13]

Btor2Aiger [12]

C prog.

Instrumentor

Instrumented C prog.

Software

witness [3]

Witness

translator

Btor2

witness [19]

Verdict

Circuit

AVR

KI

AVR

PDR

ABC

IMC

ABC

PDR

AVR

BMC

if Btor2-to-Aiger translation succeeds

Try CPV!

Evaluation Results at SV-COMP 2024

6th, 3rd, and 2nd place in ReachSafety, ReachSafety-ECA, ReachSafety-Hardware, respectively

Summary

- It is feasible to utilize sequential circuits as intermediate representations for software verification
- CPV can employ different hardware verifiers as the backend
- CPV competed well against other mature verifiers in SV-COMP
- Future work:
  - Support more verification properties (e.g., no-overflow and termination)
  - Export correctness witnesses
  - Incorporate more backend verifiers
  - Apply circuit optimization to improve the performance of verification

References

[1] Beyer, D.: State of the art in software verification and
    TACAS, pp. 561–579. LNCS 13243 (2022)
    Models and Verification, Johannes Kepler University (2007)
    24–40. LNCS 6174 (2010)
    FMCAD, pp. 125–134 (2011)
    (2020)
    423–436 (2023)