

Non-termination Witnesses and Their Validation

Zsófia Ádám, Paulína Ayaziová, Levente Bajczi, Dirk Beyer, **Marek Jankola**,
Marian Lingsch-Rosenfeld, and Jan Strejček

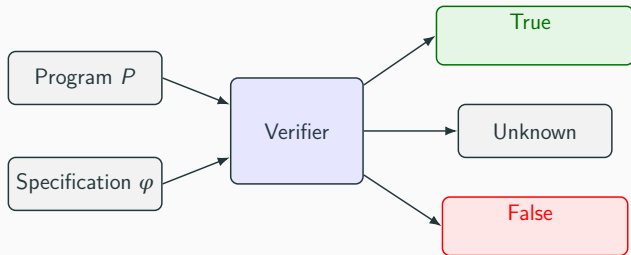


ASE 2025 @ Seoul, South Korea
BME Budapest, Hungary
LMU Munich, Germany
Masaryk University Brno, Czech Rep.

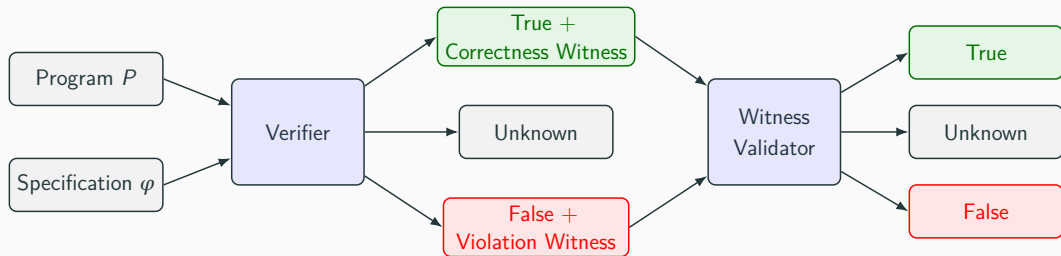


M U N I

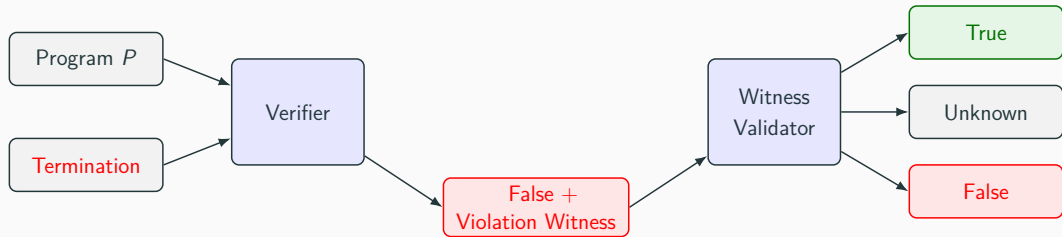
Motivation



Motivation



Motivation



Highlights

- An extension of the existing witness format to support **non-termination**
- Our extension has clear syntax and semantics

Highlights

- An extension of the existing witness format to support **non-termination**
- Our extension has clear syntax and semantics
- Tools that **construct** non-termination witnesses
 - TRANSVER [3]
 - THETA [6]
 - SYMBIOTIC [5]

Highlights

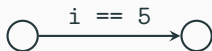
- An extension of the existing witness format to support **non-termination**
- Our extension has clear syntax and semantics
- Tools that **construct** non-termination witnesses
 - TRANSVER [3]
 - THETA [6]
 - SYMBIOTIC [5]
- Tools that **validate** non-termination witnesses
 - CPACHECKER [2]
 - THETA [6]
 - WITCH [1]

Highlights

- An extension of the existing witness format to support **non-termination**
- Our extension has clear syntax and semantics
- Tools that **construct** non-termination witnesses
 - TRANSVER [3]
 - THETA [6]
 - SYMBIOTIC [5]
- Tools that **validate** non-termination witnesses
 - CPACHECKER [2]
 - THETA [6]
 - WITCH [1]
- Large-scale evaluation of validation of the constructed witnesses

Example

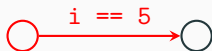
```
1  int main() {
2      int i = nondet();
3      while (i > 0) {
4          if (i != 5) {
5              i = i-1;
6          }
7      }
8      return 0;
9  }
```



```
- entry_type: "violation_sequence"
  metadata: ...
  content:
  - segment:
      - waypoint:
          type: "assumption"
          constraint:
              value: "i == 5"
              format: "c_expression"
          location:
              file_name: Ex02_sim.c
              line: 3
          action: "follow"
      - segment:
          - waypoint:
              type: "branching"
              constraint:
                  value: "true"
              location:
                  file_name: Ex02_sim.c
                  line: 3
              action: "cycle"
```

Example

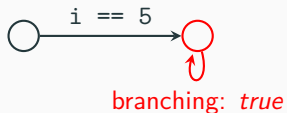
```
1  int main() {  
2      int i = nondet();  
3      while (i > 0) {  
4          if (i != 5) {  
5              i = i-1;  
6          }  
7      }  
8      return 0;  
9  }
```



```
- entry_type: "violation_sequence"  
  metadata: ...  
  content:  
    - segment:  
      - waypoint:  
        type: "assumption"  
        constraint:  
          value: "i == 5"  
          format: "c_expression"  
        location:  
          file_name: Ex02_sim.c  
          line: 3  
          action: "follow"  
      - segment:  
        - waypoint:  
          type: "branching"  
          constraint:  
            value: "true"  
          location:  
            file_name: Ex02_sim.c  
            line: 3  
            action: "cycle"
```

Example

```
1  int main() {  
2      int i = nondet();  
3      while (i > 0) {  
4          if (i != 5) {  
5              i = i-1;  
6          }  
7      }  
8      return 0;  
9  }
```



```
- entry_type: "violation_sequence"  
  metadata: ...  
  content:  
    - segment:  
      - waypoint:  
        type: "assumption"  
        constraint:  
          value: "i == 5"  
          format: "c_expression"  
        location:  
          file_name: Ex02_sim.c  
          line: 3  
          action: "follow"  
      - segment:  
        - waypoint:  
          type: "branching"  
          constraint:  
            value: "true"  
          location:  
            file_name: Ex02_sim.c  
            line: 3  
            action: "cycle"
```

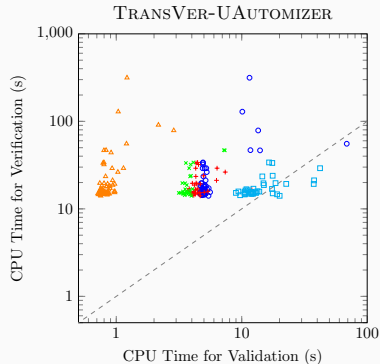
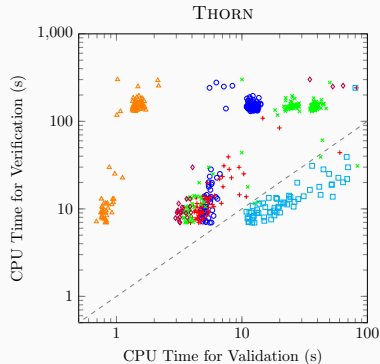
RQ1 - Effectiveness

How many constructed witnesses in new format can be validated?

Validators	EMERGENThETA 151 witnesses	ThETA 205 witnesses	ThORN 221 witnesses	TRANSVer-CPACHECKER 71 witnesses	TRANSVer-UAUTOMIZER 73 witnesses	SYMBIOTIC 658 witnesses
CPACHECKER	115	156	172	69	66	204
EMERGENThETA	62	37	61	39	41	32
ThETA	131	109	177	39	28	158
ThETA-modular	23	15	25	0	0	241
ThORN	56	36	61	39	37	43
WITCH	104	148	164	57	64	658
Total confirmed	143	173	199	71	73	658

RQ2 - Efficiency

Is validation faster than verification?



CPACHECKER ($\circ$), EMERGENTHETA ($+$), THETA ($\times$), THETA-modular ($\diamond$), THORN ($\square$), and WITCH ($\triangle$)

Conclusion

- An extension of the existing witness format to support non-termination
- Clear syntax and semantics
- The tools can validate majority of the constructed witnesses
- The validation is often faster than verification



References i

- [1] Ayaziová, P., Strejček, J.: WITCH 3: Validation of violation witnesses in the witness format 2.0 (competition contribution). In: Proc. TACAS (3). pp. 341–346. LNCS 14572, Springer (2024).
https://doi.org/10.1007/978-3-031-57256-2_18
- [2] Baier, D., Beyer, D., Chien, P.C., Jakobs, M.C., Jankola, M., Kettl, M., Lee, N.Z., Lemberger, T., Lingsch-Rosenfeld, M., Wachowitz, H., Wendler, P.: Software verification with CPACHECKER 3.0: Tutorial and user guide. In: Proc. FM. pp. 543–570. LNCS 14934, Springer (2024).
https://doi.org/10.1007/978-3-031-71177-0_30

References ii

- [3] Beyer, D., Jankola, M., Lingsch-Rosenfeld, M., Xia, T., Zheng, X.: TRANSVER: A modular program-transformation framework for reduction to reachability. In: Proc. SPIN. LNCS 15945, Springer (2025).
https://doi.org/10.1007/978-3-032-06847-7_1
- [4] Beyer, D., Wendler, P.: CPACHECKER releases. Zenodo.
<https://doi.org/10.5281/zenodo.3816620>
- [5] Jonáš, M., Kumor, K., Novák, J., Sedláček, J., Trtík, M., Zaoral, L., Ayaziová, P., Strejček, J.: SYMBIOTIC 10: Lazy memory initialization and compact symbolic execution (competition contribution). In: Proc. TACAS (3). pp. 406–411. LNCS 14572, Springer (2024).
https://doi.org/10.1007/978-3-031-57256-2_29

- [6] Telbisz, C., Bajczi, L., Szekeres, D., Vörös, A.: `THETA`: Various approaches for concurrent program verification (competition contribution). In: Proc. TACAS (3). pp. 260–265. LNCS 15698, Springer (2025).
https://doi.org/10.1007/978-3-031-90660-2_22