Cooperative Verification

CPAchecker-Workshop 2019, Chiemsee, 2019-10-01

Dirk Beyer

LMU Munich, Germany
Automatic Software Verification

Program Specification

Verifier

Program Specification Result (True/False)

Witness
Competitions in Software Verification and Testing

- SV-COMP: off-site, automatic tools, controlled [1]
- Test-Comp: off-site, automatic tools, controlled [3]
SV-COMP (Automatic Tools 2012)
SV-COMP (Automatic Tools 2014, cumulative)
SV-COMP (Automatic Tools 2016, cumulative)
SV-COMP (Automatic Tools 2017, cumulative)
SV-COMP (Automatic Tools 2019, cumulative)
What is the best verifier?

Many different kinds of programs seem to require many different good tools with different strengths
SV-COMP (Automatic Tools)

**ReachSafety**
1. VeriAbs
2. CPA-Seq
3. PeSCo

**MemSafety**
1. Symbiotic
2. PredatorHP
3. CPA-Seq

**ConcurrencySafety**
1. Yogar-CBMC
2. Lazy-CSeq
3. CPA-Seq

**NoOverflows**
1. UAutomizer
2. UTaipan
3. CPA-Seq

**Termination**
1. UAutomizer
2. AProVE
3. CPA-Seq

**SoftwareSystems**
1. CPA-BAM-BnB
2. CPA-Seq
3. VeriAbs

**FalsificationOverall**
1. CPA-Seq
2. PeSCo
3. ESBMC-kind

**Overall**
1. CPA-Seq
2. PeSCo
3. UAutomizer

https://sv-comp.sosy-lab.org/2019/results
Which techniques are used?

<table>
<thead>
<tr>
<th>Participant</th>
<th>CEGAR</th>
<th>Predicate Abstraction</th>
<th>Symbolic Execution</th>
<th>Bounded Model Checking</th>
<th>k-Induction</th>
<th>Property-Directed Reach.</th>
<th>Explicit-Value Analysis</th>
<th>Numeric-Interval Analysis</th>
<th>Shape Analysis</th>
<th>Separation Logic</th>
<th>Bit-Precise Analysis</th>
<th>ARG-Based Analysis</th>
<th>Lazy Abstraction</th>
<th>Interpolation</th>
<th>Automata-Based Analysis</th>
<th>Concurrency Support</th>
<th>Ranking Functions</th>
<th>Evolutionary Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>2LS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AProVE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CBMC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CBMC-Path</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPA-BAM-BnB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPA-Lockator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPA-Seq</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEPTH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIVINE-EXPLICIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIVINE-SMT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ESBMC-kind</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JAYHORN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JYBCM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JPF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lazy-CSbread</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Map2Check</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PeSCo</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pinaka</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PredatorHP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Skink</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYMBIOTIC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UAutomizer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UKoja</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UTktool</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

https://doi.org/10.1007/978-3-030-17502-3_9
Algorithms

17  Bounded Model Checking
13  CEGAR
  8  Predicate Abstraction
  5  k-Induction
  4  Symbolic Execution
  3  Automata-Based Analysis
  2  Property-Directed Reachability (IC3)
Abstract Domains

24 Bit-Precise Analysis
10 Explicit-Value Analysis
9 Numerical Interval Analysis
4 Shape Analysis
1 Separation Logic
Testing

- Fuzzing (VeriFuzz [12], based on AFL)
- Symbolic execution (KLEE [11])
- Software model checking (CoVeriTest [9], → Poster)
Cooperative Verification
Approaches for Combinations

Verification Approach
- Basic
- Combination
  - Black Box
    - Portfolio
    - Selection
    - Cooperative
  - White Box
    - Conceptual Integration
Conditional Model Checking

Program $P$ → Conditional Verifier A $P \models \varphi$?

Conditional Verifier A $P \models \varphi$? → Conditional Verifier B $P \models \varphi$?

Condition $\psi$ → TRUE under condition $\psi$

TRUE

FALSE

Proc. FSE 2012 [8]
Reducer-Based Construction

Verifier B

Conditional Verifier B

Proc. ICSE 2018 [10]
Reducer-Based Construction

Verifier B

Conditional Verifier B

Construction

Reducer (preprocessor)
▶ Builds standard input (C program)
▶ Representing a subset of paths
▶ Contains at least all non-verified paths

+ Verifier-unspecific approach
+ Many conditional verifiers possible

Proc. ICSE 2018 [10]
Reducer-Based Construction

Reducer (preprocessor)
- Builds standard input (C program)
- Representing a subset of paths
- Contains at least all non-verified paths

Proc. ICSE 2018 [10]
Reducer-Based Construction

Reducer (preprocessor)
- Builds standard input (C program)
- Representing a subset of paths
- Contains at least all non-verified paths
+ Verifier-unspecific approach
+ Many conditional verifiers possible

Proc. ICSE 2018 [10]
Software Verification with Witnesses

Proc. FSE 2015, 2016 [6, 5]
Witness-Based Result Validation

- Validate untrusted results
- Easier than full verification
Stepwise Refinement

Program
Specification
Result (True/False)

Witness

Witness Testifier

Program
Specification
Result (True/False)

Witness
Execution-based Witness Validation

Proc. TAP 2018 [7]
Made “Generating Tests from Counterexamples” more practical
(Proc. ICSE 2004, [4])
Witness Creation

Program

Specification

Verification Task

Verifier

False

Bug found

True

Proof found

Violation Witness

Correctness Witness

BLAST

CBMC

CPAchecker

ESBMC

SMACK

ULTIMATE AUTOMIZER

...
Graphical Visualization of the Coop Framework

- **Conditional Verifier**
- **Verifier**
- **Validator**
- **Reducer**
- **Test Extractor**
- **Test Executor**
- **Conditional Verifier constructed from Reducer and Verifier**
- **Execution-Based Validation constructed from Verifier, Test Extractor, and Test Executor**
Conclusion

- Software verification: successful past, bright future
- Competitions solve several problems
- Cooperating combinations are the future


