CPA-DF: A Tool for Configurable Interval Analysis to Boost Program Verification

Dirk Beyer, Po-Chun Chien, and Nian-Ze Lee
Highlights of CPA-DF

Use cases:
- as a standalone verifier for C programs
- as a **performance booster** in a portfolio-based verifier

Contributions:
- discovery of a new aspect of an existing component in CPAchecker [1]
- large-scale evaluation
- open-source data-flow analysis tool
Motivation

Cooperative Invariant Injection [1]

\[ \text{KI} \rightarrow \text{DF} \]

- \( k \)-induction
- Data-flow analysis based on intervals
Motivation

Cooperative
Invariant Injection [1]

Non-cooperative
Parallel Execution

\[ \text{KI} \quad || \quad \text{DF} \]

\[ k\text{-induction} \quad \text{data-flow analysis based on intervals} \]

SoSy-Lab, LMU Munich
CPA-DF
Configurable Program Analysis using Interval Expressions

- Interval CPA $\mathcal{I}$
  - Abstract domain: $\text{var} \rightarrow \text{interval expression}$
    - E.g. $[l_1, u_1] \cup [l_2, u_2]$  
  - Precision
    - Important variables: merge allowed only if expressions match on them
    - Widening [4]: whether to further relax an abstraction
- CPA$^+$ [2]: a reachability algorithm exploring abstract states
Interval Analysis with Precision Refinement

Verif. Task \[\rightarrow\] CPA+ (initial precision)

SoSy-Lab, LMU Munich
Interval Analysis with Precision Refinement

Verif. Task \rightarrow CPA+ \rightarrow Safety Check

(initial precision)

abstract reachable states

precision refinement

unknown not possible precision increment
Interval Analysis with Precision Refinement

Verif. Task $\rightarrow$ CPA$^+$

(initial precision)

abstract reachable states

Safety Check $\rightarrow$ safe

Precision Refinement unknown not possible precision increment

SoSy-Lab, LMU Munich CPA-DF 4/13
Interval Analysis with Precision Refinement

Verif. Task \rightarrow CPA^+ \rightarrow Safety Check

Precision Refinement

(initial precision)

abstract reachable states

safe

Precision Refinement

Precision Refinement
Interval Analysis with Precision Refinement

Verif. Task \rightarrow CPA+ \rightarrow Safety Check

Precision Refinement \not\rightarrow \text{unknown} \not\rightarrow \text{safe}

(initial precision)

abstract reachable states
Interval Analysis with Precision Refinement

Verif. Task \rightarrow CPA+ \rightarrow Precision Refinement

Precision Refinement \rightarrow Safety Check

(initial precision)

precision increment

abstract reachable states

unknown \rightarrow not possible

safe

unknown
Evaluation

We conducted experiments to answer the following research questions:

- **RQ1**: How is $\text{KI} \parallel \text{DF}$ compared to $\text{KI}\otimes\text{DF}$ in CPAchecker?
- **RQ2**: Can CPA-DF complement other verifiers in a parallel portfolio?
Benchmark Tasks and Tools

- 6386 ReachSafety tasks from SV-COMP '23 benchmark set
  - all safe, i.e. without known property violation
- Top contenders from SV-COMP '23
  - CPAchecker [3]: DF, KI, KI || DF, KI ⊤-DF
  - Esbmc [5]
  - Symbiotic [6]
Experimental Setup

- OS: Ubuntu 22.04 (64 bit)
- Machine: 3.4 GHz CPU (8 cores) and 33 GB of RAM
- Each task is limited to
  - 4 CPU cores
  - 15 min of CPU time
  - 15 GB of RAM

(reliable resource management by `BenchExec`\(^1\))

\(^1\)https://github.com/sosy-lab/benchexec
RQ1: How is $\text{KI} \parallel \text{DF}$ compared to $\text{KI} \bowtie \text{DF}$ in CPAchecker?
Parallel Portfolio vs. Invariant Injection

SoSy-Lab, LMU Munich
CPA-DF
Overall Statistics of $\text{KI} \parallel \text{DF}$ vs. $\text{KI} \leftrightarrow \text{DF}$

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>DF</th>
<th>KI</th>
<th>KI $\parallel$ DF</th>
<th>KI $\leftrightarrow$ DF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct proofs</td>
<td>1719</td>
<td>1738</td>
<td>2572</td>
<td>2673</td>
</tr>
<tr>
<td>Wrong proofs</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>Unknowns</td>
<td>6537</td>
<td>6277</td>
<td>5472</td>
<td>5356</td>
</tr>
</tbody>
</table>
RQ2: Can CPA-DF complement other verifiers in a parallel portfolio?
Boosting Program Verification with CPA-DF

SoSy-Lab, LMU Munich

### #Proofs

<table>
<thead>
<tr>
<th>Tool</th>
<th>w/o DF</th>
<th>w/ DF</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESBMC [5]</td>
<td>2962</td>
<td>3143</td>
</tr>
<tr>
<td>SYMBIOTIC [6]</td>
<td>2046</td>
<td>2904</td>
</tr>
</tbody>
</table>

SoSy-Lab, LMU Munich

CPA-DF
Conclusion

- **CPA-DF**: a data-flow analysis tool based on interval expressions
- In our evaluation,
  - running CPA-DF in parallel to KI achieved a comparable performance as running them cooperatively, and
  - CPA-DF complemented the other well-established verifiers well and boosted the overall performance.

For more info., visit: www.sosy-lab.org/research/cpa-df/


