PeRIPLO
Proof tRansformer and Interpolator for Propositional LOgic

Simone Fulvio Rollini

Formal Verification Lab, University of Lugano

July 14th, 2013
The PeRIPLO Framework
Outline

1. The PeRIPLO Framework
2. Proof Compression
1. The PeRIPLo Framework
2. Proof Compression
3. Interpolation
Outline

1. The PeRIPLO Framework
2. Proof Compression
3. Interpolation
Tool Description

- Open-source tool built on MiniSAT 2.2.0
• Open-source tool built on MiniSAT 2.2.0

• Born from OpenSMT for SAT-based model checking
Tool Description

• Open-source tool built on MiniSAT 2.2.0

• Born from OpenSMT for SAT-based model checking

• Features
  • SAT-solving
• Open-source tool built on MiniSAT 2.2.0

• Born from OpenSMT for SAT-based model checking

• Features
  • SAT-solving
  • Proof compression
PeRIPLO
Tool Description

- Open-source tool built on MiniSAT 2.2.0
- Born from OpenSMT for SAT-based model checking
- Features
  - SAT-solving
  - Proof compression
  - Interpolants generation (single and collections)
PeRIPLO

Tool Description

• Open-source tool built on MiniSAT 2.2.0

• Born from OpenSMT for SAT-based model checking

• Features
  • SAT-solving
  • Proof compression
  • Interpolants generation (single and collections)

• On demand development
Usage

- Interface:
  - Configuration file
  - Application Programming Interface
  - Input:
    - Propositional formula (SMT-LIB2 format)
  - Output:
    - Sat/Unsat
    - Refutation
    - Interpolants
    - Various statistics
Usage

- Interface:
  - Configuration file

S.F. Rollini (USI)
Usage

- Interface:
  - Configuration file
  - Application Programming Interface

- Input:
  - Propositional formula (SMT-LIB2 format)

- Output:
  - Sat/Unsat
  - Refutation
  - Interpolants
  - Various statistics
Usage

- **Interface:**
  - Configuration file
  - Application Programming Interface

- **Input:**

- **Output:**
  - Sat/Unsat
  - Refutation
  - Interpolants
  - Various statistics
Usage

- **Interface:**
  - Configuration file
  - Application Programming Interface

- **Input:**
  - Propositional formula (SMT-LIB2 format)
Usage

- **Interface:**
  - Configuration file
  - Application Programming Interface

- **Input:**
  - Propositional formula (SMT-LIB2 format)

- **Output:**
Usage

- **Interface:**
  - Configuration file
  - Application Programming Interface

- **Input:**
  - Propositional formula (SMT-LIB2 format)

- **Output:**
  - Sat/Unsat
Usage

- **Interface:**
  - Configuration file
  - Application Programming Interface

- **Input:**
  - Propositional formula (SMT-LIB2 format)

- **Output:**
  - Sat/Unsat
  - Refutation
Usage

- **Interface:**
  - Configuration file
  - Application Programming Interface

- **Input:**
  - Propositional formula (SMT-LIB2 format)

- **Output:**
  - Sat/Unsat
  - Refutation
  - Interpolants
• **Interface:**
  - Configuration file
  - Application Programming Interface

• **Input:**
  - Propositional formula (SMT-LIB2 format)

• **Output:**
  - Sat/Unsat
  - Refutation
  - Interpolants
  - Various statistics
Outline

1. The PeRIpLO Framework
2. Proof Compression
3. Interpolation
• Propositional satisfiability
- Propositional satisfiability
- Resolution proof system
• Propositional satisfiability

• Resolution proof system

• Set of clauses \( \{ \overline{op}, \overline{p}, \overline{op} \} \)
SAT

- Propositional satisfiability
- Resolution proof system
- Set of clauses \( \{\text{op}, \overline{p}, \overline{\text{op}}\} \)
- Resolution proof DAG
Compression Framework

- SAT-solving
Compression Framework

- SAT-solving
  - DPLL CDCL
  - Generic
Compression Framework

- SAT-solving
  - DPLL CDCL
  - Generic

- Post-processing approach
Compression Framework

- SAT-solving
  - DPLL CDCL
  - Generic

- Post-processing approach

- Compression algorithms
Compression Framework

- SAT-solving
  - DPLL CDCL
  - Generic

- Post-processing approach

- Compression algorithms
  - Structural hashing at proof chains level [C10]
  - Lower unit clauses [FMP11]
  - Local Transformation Framework [BRST10,RBS10]
  - Structural hashing at proof level
  - Removal pivots redundancies along paths [BFHSS08,FMP11]
begin
  LowerUnits();
  for \( i = 1 \) to number of iterations do
    StructuralHashing();
    RecyclePivotsWithIntersection();
    for \( i = 1 \) to number of traversals do
      ReduceAndExpose();
    end
  end
end
## Experimental Evaluation

**SAT Challenge 2012, SATLIB, CMU BMC**

<table>
<thead>
<tr>
<th></th>
<th>#Bench</th>
<th>RedNodes</th>
<th>RedCore</th>
<th>RedEdges</th>
<th>TranTime(s)</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LU</strong></td>
<td>180</td>
<td>1.49%</td>
<td>0.00%</td>
<td>1.89%</td>
<td>2.89</td>
<td>0.09</td>
</tr>
<tr>
<td><strong>SH</strong></td>
<td>180</td>
<td>6.17%</td>
<td>0.00%</td>
<td>6.89%</td>
<td>2.43</td>
<td>0.08</td>
</tr>
<tr>
<td><strong>RPI</strong></td>
<td>180</td>
<td>25.74%</td>
<td>1.17%</td>
<td>28.12%</td>
<td>7.15</td>
<td>0.20</td>
</tr>
<tr>
<td><strong>RE 3</strong></td>
<td>180</td>
<td>3.95%</td>
<td>0.07%</td>
<td>4.73%</td>
<td>13.23</td>
<td>0.31</td>
</tr>
<tr>
<td><strong>LU+SH+RPI</strong></td>
<td>180</td>
<td>31.04%</td>
<td>1.09%</td>
<td>34.13%</td>
<td>13.05</td>
<td>0.32</td>
</tr>
<tr>
<td><strong>LU+SH+RPI+RE</strong></td>
<td>180</td>
<td>37.85%</td>
<td>1.51%</td>
<td>41.95%</td>
<td>24.19</td>
<td>0.46</td>
</tr>
<tr>
<td>2,3</td>
<td>180</td>
<td>40.09%</td>
<td>1.68%</td>
<td>44.50%</td>
<td>32.94</td>
<td>0.54</td>
</tr>
<tr>
<td>3,3</td>
<td>180</td>
<td>40.09%</td>
<td>1.68%</td>
<td>44.50%</td>
<td>32.94</td>
<td>0.54</td>
</tr>
</tbody>
</table>
References

S.F. Rollini, R. Bruttomesso and N. Sharygina
*An Efficient and Flexible Approach to Resolution Proof Reduction.*
HVC 2010.

R. Bruttomesso, S.F. Rollini, N. Sharygina and A. Tsitovich
*Flexible Interpolation with Local Proof Transformations.*
ICCAD 2010.

S.F. Rollini, R. Bruttomesso, N. Sharygina and A. Tsitovich
*Resolution Proof Transformation for Compression and Interpolation.*
http://arxiv.org/abs/1307.2028

S.F. Rollini
*PeRIPLO - Tool Description.*
http://verify.inf.unisi.ch/periplo.html
Outline

1. The PeRIPLO Framework
2. Proof Compression
3. Interpolation
Propositional Interpolation

- Resolution proof of unsatisfiability
Propositional Interpolation

- Resolution proof of unsatisfiability
- Single interpolants
Propositional Interpolation

- Resolution proof of unsatisfiability
- Single interpolants
- Collections of interpolants
Propositional Interpolation

- Resolution proof of unsatisfiability
- Single interpolants
- Collections of interpolants
- Interpolation properties in model checking
Propositional Interpolation

Interpolants Generation

- Interpolant $I$ for unsatisfiable $A \land B$
Propositional Interpolation

Interpolants Generation

- Interpolant $I$ for unsatisfiable $A \land B$
- Different procedures [P97, McM04, DPKW10]
Propositional Interpolation

Interpolants Generation

- Interpolant $I$ for unsatisfiable $A \land B$
- Different procedures [P97,McM04,DKPW10]
- Generation approach
Propositional Interpolation

Interpolants Generation

- Interpolant $I$ for unsatisfiable $A \land B$
- Different procedures [P97, McM04, DKPW10]
- Generation approach
  - Derivation of unsatisfiability resolution proof of $A \land B$
Propositional Interpolation

Interpolants Generation

- Interpolant $I$ for unsatisfiable $A \land B$

- Different procedures [P97, McM04, DPKW10]

- Generation approach
  - Derivation of unsatisfiability resolution proof of $A \land B$
  - Computation of $I$ from proof structure
Labeled Interpolation Systems

Propositional Interpolation

- Interpolation parametric in labeling function [DKPW10]
Labeled Interpolation Systems
Propositional Interpolation

- Interpolation parametric in labeling function [DKPW10]
- Interpolant determined by proof and labeling $L$
Labeled Interpolation Systems

Propositional Interpolation

- Interpolation parametric in labeling function [DKPW10]
- Interpolant determined by proof and labeling $L$
- Generalization of [P97,McM04] ($P, M, M'$)
• Interpolation parametric in labeling function [DKPW10]

• Interpolant determined by proof and labeling $L$

• Generalization of [P97, McM04] ($P, M, M'$)

• Strength comparison reduced to labeling comparison
• $L_1 \preceq L_2 \implies l_1 \rightarrow l_2$
Labeling Lattice

Labeled Interpolation Systems

- $L_1 \preceq L_2 \implies I_1 \rightarrow I_2$

- Labeling lattice

\[ \begin{array}{c}
\text{weaker} \\
\downarrow \\
M \\
\text{stronger} \\
\uparrow \\
M' \\
\end{array} \]
Focus on interpolant strength
Focus on interpolant strength

Strength affects overapproximation coarseness
Labeled Interpolation Systems

Interpolant Strength

- Focus on interpolant strength
- Strength affects overapproximation coarseness
- Strength can affect verification performance, convergence
Collections of Interpolants

- Path Interpolation [JM06]
Interpolation Properties in Model Checking

Collections of Interpolants

- Path Interpolation [JM06]
- Symmetric Interpolation / Simultaneous Abstraction [JM05]
Interpolation Properties in Model Checking

Collections of Interpolants

- Path Interpolation [JM06]
- Symmetric Interpolation / Simultaneous Abstraction [JM05]
- State-Transition Interpolation [AGC12]
Interpolation Properties in Model Checking

Collections of Interpolants

- Path Interpolation [JM06]
- Symmetric Interpolation / Simultaneous Abstraction [JM05]
- State-Transition Interpolation [AGC12]
- Tree Interpolation [MR13]
• Systematic exploitation of interpolant strength in model checking
• Systematic exploitation of interpolant strength in model checking

• Unsatisfiable formula \( \tau_1 \land \ldots \land \tau_m \)
Labeled Interpolation Systems and Interpolation Properties

- Systematic exploitation of interpolant strength in model checking
- Unsatisfiable formula $\tau_1 \land \ldots \land \tau_m$
- Generation of multiple interpolants $I_1, \ldots, I_n$
• Systematic exploitation of interpolant strength in model checking

• Unsatisfiable formula \( \tau_1 \land \ldots \land \tau_m \)

• Generation of multiple interpolants \( I_1, \ldots, I_n \)

• Generation of each \( I_i \) with different \( L_i \)
• Systematic exploitation of interpolant strength in model checking

• Unsatisfiable formula $\tau_1 \wedge \ldots \wedge \tau_m$

• Generation of multiple interpolants $I_1, \ldots, I_n$

• Generation of each $I_i$ with different $L_i$

• Interpolation property requirements
• Systematic exploitation of interpolant strength in model checking

• Unsatisfiable formula \( \tau_1 \land \ldots \land \tau_m \)

• Generation of multiple interpolants \( I_1, \ldots, I_n \)

• Generation of each \( I_i \) with different \( L_i \)

• Interpolation property requirements

• Identification of constraints on \( L_1, \ldots, L_n \)
Interpolation Property Requirements
Simultaneous Abstraction

- Requirement: \( I_1 \land \ldots \land I_n \) SAT
Interpolation Property Requirements
Simultaneous Abstraction

- Requirement: \[ I_1 \land \ldots \land I_n \land \text{UNSAT} \]
- Satisfied for: \[ L_1, \ldots, L_n \preceq \text{Pudlák [RSS12]} \]
Interpolation Property Requirements

Simultaneous Abstraction

- Requirement: \( I_1 \land \ldots \land I_n \) UNSAT

- Satisfied for: \( L_1, \ldots, L_n \preceq \) Pudlák [RSS12]

- Not satisfied in general for: \( L_i \succ \) Pudlák [GRS13]
Interpolation Property Requirements

Simultaneous Abstraction

- Requirement: \( I_1 \land \ldots \land I_n \) \( \text{UNSAT} \)
- Satisfied for: \( L_1, \ldots, L_n \preceq \) Pudlák [RSS12]
- Not satisfied in general for: \( L_i \succ \) Pudlák [GRS13]
Interpolation in PeRIPLO

- Labeled Interpolation Systems
Interpolation in PeRIPLO

- Labeled Interpolation Systems
- Single interpolants
Interpolation in PeRIPLO

- Labeled Interpolation Systems
- Single interpolants
- Collection of interpolants
Interpolation in PeRIPLO

- Labeled Interpolation Systems
- Single interpolants
- Collection of interpolants
  - (Generalized) Simultaneous Abstraction
  - Path Interpolation
  - State-transition Interpolation
  - Tree Interpolation
Interpolation in PeRIPLO

- Labeled Interpolation Systems

- Single interpolants

- Collection of interpolants
  - (Generalized) Simultaneous Abstraction
  - Path Interpolation
  - State-transition Interpolation
  - Tree Interpolation

- Independent verification of interpolants and requirements
S.F. Rollini, O. Sery and N. Sharygina
*Leveraging Interpolant Strength in Model Checking.*
CAV 2012.

A. Gurfinkel, S.F. Rollini, and N. Sharygina
*Interpolation Properties and SAT-based Model Checking.*
Summary

- PeRIPLO framework
  - Input, output, usage
- Proof compression
- Interpolation in model checking
- http://verify.inf.unisi.ch/periplo.html