Local proof transformations for flexible interpolation and proof reduction

N. Sharygina

Formal Verification and Security Group
University of Lugano

June 21, 2011
1 Background
Outline

1. Background

2. Motivation and Related Work
Outline

1 Background

2 Motivation and Related Work

3 Contribution
   - Proof Transformation for Interpolation and Reduction
Outline

1. Background

2. Motivation and Related Work

3. Contribution
   - Proof Transformation for Interpolation and Reduction

4. Summary and Future Work
Outline

1 Background

2 Motivation and Related Work

3 Contribution
   - Proof Transformation for Interpolation and Reduction

4 Summary and Future Work
Background
Formal Verification in Lugano, Switzerland

- Program Verification
• Program Verification
  • Model checking code (LoopFrog, Synergy, SatAbs (with Oxford), FunFrog), ANSI-C
  • Efficient decision procedures as computational engines of verification (OpenSMT)
Background

Formal Verification in Lugano, Switzerland

- Program Verification
  - Model checking code (LoopFrog, Synergy, SatAbs (with Oxford), FunFrog), ANSI-C
  - Efficient decision procedures as computational engines of verification (OpenSMT)

- Abstractions
Background

Formal Verification in Lugano, Switzerland

- Program Verification
  - Model checking code (LoopFrog, Synergy, SatAbs (with Oxford), FunFrog), ANSI-C
  - Efficient decision procedures as computational engines of verification (OpenSMT)

- Abstractions
  - Program Summarization [ATVA’08], [ASE’09]
    - Avoids fix-point computation by constructing symbolic abstract transformers instead
    - Performs sound over-approximation of (unbounded) loops
    - Precision is tuned by selection of abstract domains
    - Exploits efficiency of SAT/SMT solvers
• Program Termination [CAV’10, TACAS’11]
  • Integration of Loop Summarization with Termination Analysis
  • Compositional Transition Invariants avoid all paths computation of termination checks
  • Simple abstract domains are used for termination checks
Background
Formal Verification in Lugano, Switzerland

- Program Termination [CAV’10, TACAS’11]
  - Integration of Loop Summarization with Termination Analysis
  - Compositional Transition Invariants avoid all paths computation of termination checks
  - Simple abstract domains are used for termination checks

- Synergy of Abstractions [STTT’10]
  - Interleaves precise and over-approximated abstractions
  - Reduces CEGAR iterations
  - Removes multiple counterexamples within a single refinement step
  - Localizes precise abstraction/refinement to relevant parts of the program
Background

Formal Verification in Lugano, Switzerland

- Model checking mobile code [IFM’08], [JFAC’10]
  - Specification language for security policies
  - Formalization of mobile code distribution net
  - Location-specific abstractions and model checking of security policies
Background
Formal Verification in Lugano, Switzerland

- Model checking mobile code [IFM’08], [JFAC’10]
  - Specification language for security policies
  - Formalization of mobile code distribution net
  - Location-specific abstractions and model checking of security policies

- Boolean and Theory Reasoning (SMT)
  - Procedure for bit-vector extraction and concatenation [ICCAD’09]
    - Reduces formulae to the theory of equality to avoid, when possible, expensive reduction to SAT
Background
Formal Verification in Lugano, Switzerland

- Model checking mobile code [IFM’08], [JFAC’10]
  - Specification language for security policies
  - Formalization of mobile code distribution net
  - Location-specific abstractions and model checking of security policies

- Boolean and Theory Reasoning (SMT)
  - Procedure for bit-vector extraction and concatenation [ICCAD’09]
    - Reduces formulae to the theory of equality to avoid, when possible, expensive reduction to SAT
  - Generation of explanations in theory propagation [MEMOCODE’10]
    - Computes explanations on demand by reusing the consistency check algorithm for a generic theory $T$. 
Background

Formal Verification in Lugano, Switzerland

- Boolean and Theory Reasoning (SMT)
  - Generation of interpolants (for QF EUF, RDL)
  - Proof manipulation for interpolation [ICCAD’10]
  - Proof reduction [HVC’10]
• Boolean and Theory Reasoning (SMT)
  • Generation of interpolants (for QF EUF, RDL)
  • Proof manipulation for interpolation [ICCAD’10]
  • Proof reduction [HVC’10]
  • Solver, OpenSMT, combines MiniSAT2 SAT-Solver with state-of-the-art decision procedures for QF EUF, LRA, BV, RDL, IDL
    • Extensible: the SAT-to-theory interface facilitates design and plug-in of new decision procedures
    • Incremental: suitable for incremental verification
    • Open-source: available under GPL license
    • Efficient: currently the fastest open-source SMT Solver for QF UF, IDL, RDL, LRA according to SMT-Comp’10.
Background

Formal Verification in Lugano, Switzerland

- Boolean and Theory Reasoning (SMT)
  - Generation of interpolants (for QF EUF, RDL)
  - Proof manipulation for interpolation [S.F. Rollini, R. Bruttomesso, N. Sharygina, A. Tsitovich, ICCAD’10]
  - Resolution proof reduction [S.F. Rollini, R. Bruttomesso, N. Sharygina, HVC’10]
Outline

1. Background
2. Motivation and Related Work
3. Contribution
   - Proof Transformation for Interpolation and Reduction
4. Summary and Future Work
Resolution proofs find application in several ambits.
Proof Transformation and Reduction

Motivation

- Resolution proofs find application in several ambits
  - Interpolation-based model checking
  - Abstraction techniques
  - Unsatisfiable core extraction in SAT/SMT
  - Automatic theorem proving
Proof Transformation and Reduction

Motivation

• Resolution proofs find application in several ambits
  • Interpolation-based model checking
  • Abstraction techniques
  • Unsatisfiable core extraction in SAT/SMT
  • Automatic theorem proving

• Problems
Proof Transformation and Reduction

Motivation

• Resolution proofs find application in several ambits
  • Interpolation-based model checking
  • Abstraction techniques
  • Unsatisfiable core extraction in SAT/SMT
  • Automatic theorem proving

• Problems
  • Clean structure of proofs is required for interpolation generation
Proof Transformation and Reduction

Motivation

- Resolution proofs find application in several ambits
  - Interpolation-based model checking
  - Abstraction techniques
  - Unsatisfiable core extraction in SAT/SMT
  - Automatic theorem proving

- Problems
  - Clean structure of proofs is required for interpolation generation
  - Size affects efficiency
  - Size can be exponential w.r.t. input formula
Craig’s interpolant \( I \) for unsatisfiable conjunction of formulae \( A \land B \) [Craig57]
• Craig’s interpolant $I$ for unsatisfiable conjunction of formulae $A \land B$ [Craig57]
  
  • $A \Rightarrow I$, $I \land B$ unsatisfiable
• Craig’s interpolant $I$ for unsatisfiable conjunction of formulae $A \land B$ [Craig57]
  
  • $A \Rightarrow I$, $I \land B$ unsatisfiable
  • $I$ defined over common symbols of $A$ and $B$
Craig’s interpolant $I$ for unsatisfiable conjunction of formulae $A \land B$ [Craig57]

- $A \Rightarrow I, I \land B$ unsatisfiable
- $I$ defined over common symbols of $A$ and $B$
- $I$ as over-approximation $A$ conflicting with $B$
• Craig’s interpolant \( l \) for unsatisfiable conjunction of formulae \( A \land B \) [Craig57]

- \( A \Rightarrow l, l \land B \) unsatisfiable
- \( l \) defined over common symbols of \( A \) and \( B \)
- \( l \) as over-approximation \( A \) conflicting with \( B \)

• Example

\[
A \equiv (p \lor q) \land (p \lor q) \\
B \equiv (q \lor r) \land (q \lor r)
\]
Craig’s interpolant $I$ for unsatisfiable conjunction of formulae $A \land B$ [Craig57]

- $A \Rightarrow I$, $I \land B$ unsatisfiable
- $I$ defined over common symbols of $A$ and $B$
- $I$ as over-approximation $A$ conflicting with $B$

**Example**

- $A \triangleq (\bar{p} \lor \bar{q}) \land (p \lor \bar{q})$
- $B \triangleq (q \lor \bar{r}) \land (q \lor r)$
Craig’s interpolant $I$ for unsatisfiable conjunction of formulae $A \land B$ [Craig57]

- $A \Rightarrow I$, $I \land B$ unsatisfiable
- $I$ defined over common symbols of $A$ and $B$
- $I$ as over-approximation $A$ conflicting with $B$

Example

- $A \triangleq (\overline{p} \lor \overline{q}) \land (p \lor \overline{q})$
- $B \triangleq (q \lor \overline{r}) \land (q \lor r)$
- Interpolant $\overline{q}$
Craig’s interpolant $I$ for unsatisfiable conjunction of formulae $A \land B$ [Craig57]

- $A \Rightarrow I$, $I \land B$ unsatisfiable
- $I$ defined over common symbols of $A$ and $B$
- $I$ as over-approximation $A$ conflicting with $B$

Example

- $A \triangleq (\overline{p} \lor \overline{q}) \land (p \lor \overline{q})$  
  $B \triangleq (q \lor \overline{r}) \land (q \lor r)$
- Interpolant $\overline{q}$
- $A \Rightarrow \overline{q}$  
  $\overline{q} \land B$ unsatisfiable
Interpolation

Background

• Craig’s interpolant $I$ for unsatisfiable conjunction of formulae $A \land B$ [Craig'57]
  
  • $I$ as over-approximation $A$ conflicting with $B$
Interpolation

Background

- Applications in symbolic model checking
• Applications in symbolic model checking

  • Bounded model checking: approximate cheaper reachability set computation [McMillan03]
Applications in symbolic model checking

- Bounded model checking: approximate cheaper reachability set computation [McMillan03]
- Predicate abstraction refinement based on spurious behaviors [Henzinger04]
Applications in symbolic model checking

- Bounded model checking: approximate cheaper reachability set computation [McMillan03]
- Predicate abstraction refinement based on spurious behaviors [Henzinger04]
- Property-based transition relation approximation [Jhala05]
Interpolation

Background

• Applications in symbolic model checking
  
  • Bounded model checking: approximate cheaper reachability set computation [McMillan03]
  
  • Predicate abstraction refinement based on spurious behaviors [Henzinger04]
  
  • Property-based transition relation approximation [Jhala05]
  
• Forementioned applications involve

Problem encoding into logic (SAT, SMT)

Problem solving by means of resolution based engines (SAT solvers, SMT solvers)
Interpolation

Background

- Applications in symbolic model checking
  - Bounded model checking: approximate cheaper reachability set computation [McMillan03]
  - Predicate abstraction refinement based on spurious behaviors [Henzinger04]
  - Property-based transition relation approximation [Jhala05]
- Forementioned applications involve
  - Problem encoding into logic (SAT, SMT)
Interpolation

Background

• Applications in symbolic model checking
  • Bounded model checking: approximate cheaper reachability set computation [McMillan03]
  • Predicate abstraction refinement based on spurious behaviors [Henzinger04]
  • Property-based transition relation approximation [Jhala05]

• Forementioned applications involve
  • Problem encoding into logic (SAT, SMT)
  • Problem solving by means of resolution based engines (SAT solvers, SMT solvers)
SAT and SMT

Background

- Satisfiability (SAT)

\[ A \equiv (p \lor q) \land (p \lor q) \]

\[ B \equiv (q \lor r) \land (q \lor r) \]

- Satisfiability Modulo Theories (SMT): more expressivity than boolean logic

- Timed automata, hybrid systems, . . .

- Arbitrary precision arithmetic, data structures . . .

\[ A \equiv (5x - y \leq 1) \land (y - 5x \leq -1) \]

\[ B \equiv (y - 5z \leq 3) \land (5z - y \leq -2) \]
SAT and SMT

Background

- Satisfiability (SAT)
  - Example
    \[ A \triangleq (\overline{p} \lor \overline{q}) \land (p \lor \overline{q}) \quad B \triangleq (q \lor \overline{r}) \land (q \lor r) \]
SAT and SMT

Background

- Satisfiability (SAT)
  - Example
    \[
    A \triangleq (\overline{p} \lor \overline{q}) \land (p \lor \overline{q}) \\
    B \triangleq (q \lor \overline{r}) \land (q \lor r)
    \]

- Satisfiability Modulo Theories (SMT): more expressivity than boolean logic
SAT and SMT

Background

- Satisfiability (SAT)
  - Example
    \[ A \triangleq (\bar{p} \lor \bar{q}) \land (p \lor \bar{q}) \quad B \triangleq (q \lor \bar{r}) \land (q \lor r) \]

- Satisfiability Modulo Theories (SMT): more expressivity than boolean logic
  - Timed automata, hybrid systems, ...
SAT and SMT

Background

- Satisfiability (SAT)
  - Example
    \[ A \triangleq (\overline{p} \lor \overline{q}) \land (p \lor \overline{q}) \quad B \triangleq (q \lor \overline{r}) \land (q \lor r) \]

- Satisfiability Modulo Theories (SMT): more expressivity than boolean logic
  - Timed automata, hybrid systems, ...
  - Arbitrary precision arithmetic, data structures ...

SAT and SMT

Background

- **Satisfiability (SAT)**
  - Example
    \[
    A \triangleq (\overline{p} \lor \overline{q}) \land (p \lor \overline{q}) \\
    B \triangleq (q \lor \overline{r}) \land (q \lor r)
    \]

- **Satisfiability Modulo Theories (SMT): more expressivity than boolean logic**
  - Timed automata, hybrid systems, . . .
  - Arbitrary precision arithmetic, data structures . . .
  - Example
    \[
    A \triangleq (5x - y \leq 1) \land (y - 5x \leq -1) \\
    B \triangleq (y - 5z \leq 3) \land (5z - y \leq -2)
    \]
• $A \land B$ unsatisfiable: certificate of unsatisfiability
• $A \land B$ unsatisfiable: certificate of unsatisfiability
  • Propositional proof of unsatisfiability
  • Generated by logging steps at solving time
• $A \land B$ unsatisfiable: certificate of unsatisfiability
  • Propositional proof of unsatisfiability
  • Generated by logging steps at solving time

• DPLL SAT solver [Davis60,62]
SAT and SMT
Proofs and Solving Engines

- $A \land B$ unsatisfiable: certificate of unsatisfiability
  - Propositional proof of unsatisfiability
  - Generated by logging steps at solving time

- DPLL SAT solver [Davis60,62]
  - Search space boolean assignments
  - Backtracking
SAT and SMT
Proofs and Solving Engines

- \( A \land B \) unsatisfiable: certificate of unsatisfiability
  - Propositional proof of unsatisfiability
  - Generated by logging steps at solving time

- DPLL SAT solver [Davis60,62]
  - Search space boolean assignments
  - Backtracking

- SMT solver
• $A \land B$ unsatisfiable: certificate of unsatisfiability
  • Propositional proof of unsatisfiability
  • Generated by logging steps at solving time

• DPLL SAT solver [Davis60,62]
  • Search space boolean assignments
  • Backtracking

• SMT solver
  • DPLL SAT solver
  • Theory solver
• Interpolant $I$ for unsatisfiable conjunction of formulae $A \land B$
• Interpolant \( I \) for unsatisfiable conjunction of formulae \( A \land B \)

• State-of-the-art approach [Pudlák97, McMillan04]
• Interpolant $I$ for unsatisfiable conjunction of formulae $A \land B$

• State-of-the-art approach [Pudlák97, McMillan04]
  • Derivation of unsatisfiability resolution proof of $A \land B$
Interpolation
Generation

- Interpolant $I$ for unsatisfiable conjunction of formulae $A \land B$

- State-of-the-art approach [Pudlák97, McMillan04]
  - Derivation of unsatisfiability resolution proof of $A \land B$
  - Computation of $I$ from proof structure in linear time
Resolution System

Background

- Literal $p \quad \bar{p}$
Resolution System

Background

- Literal \( p \) \( \overline{p} \)

- Clause \( p \lor \overline{q} \lor r \lor \ldots \rightarrow p\overline{q}r \ldots \) Empty clause \( \bot \)
Resolution System

Background

- Literal  \( p, \overline{p} \)

- Clause  \( p \lor \overline{q} \lor r \lor \ldots \rightarrow p\overline{q}r\ldots \)  Empty clause  \( \bot \)

- Input formula  \((p \lor q) \land (r \lor \overline{p})\ldots \rightarrow \{pq, r\overline{p}\}\)
Resolution System

Background

- **Literal** \( p \), \( \overline{p} \)

- **Clause** \( p \lor \overline{q} \lor r \lor \ldots \rightarrow p\overline{q}r\ldots \) **Empty clause** \( \bot \)

- **Input formula** \((p \lor q) \land (r \lor \overline{p})\ldots \rightarrow \{pq, r\overline{p}\}\)

- **Resolution rule**

  \[
  \frac{pC \quad \overline{p}D}{CD \quad \overline{p}D} \]

  **Antecedents**: \( pC \overline{p}D \) **Resolvent**: \( CD \) **Pivot**: \( p \)
Resolution System

Background

- Literal  \( p, \overline{p} \)

- Clause  \( p \lor \overline{q} \lor r \lor \ldots \rightarrow p\overline{q}r \ldots \)

- Empty clause  \( \bot \)

- Input formula  \((p \lor q) \land (r \lor \overline{p}) \ldots \rightarrow \{pq, r\overline{p}\}\)

- Resolution rule

\[
\begin{array}{c c c}
\text{pC} & \overline{pD} & \rightarrow \ p \\
\end{array}
\]

Antecedents:  \( pC, \overline{pD} \)

Resolvent:  \( CD \)

Pivot:  \( p \)

- Resolution proof of unsatisfiability of a set of clauses  \( S \)
Resolution System

Background

- Literal \( p \quad \overline{p} \)

- Clause \( p \lor \overline{q} \lor r \lor \ldots \rightarrow p\overline{q}r \ldots \) Empty clause \( \bot \)

- Input formula \( (p \lor q) \land (r \lor \overline{p}) \ldots \rightarrow \{pq, r\overline{p}\} \)

- Resolution rule

\[
\begin{array}{c}
 pC \\
 \overline{p}D
\end{array}
\]

\[
\hline
 CD \\
 p
\end{array}
\]

Antecedents: \( pC \quad \overline{p}D \)  Resolvent: \( CD \)  Pivot: \( p \)

- Resolution proof of unsatisfiability of a set of clauses \( S \)
  - Tree
  - Leaves as clauses of \( S \)
  - Intermediate nodes as resolvents
  - Root as unique empty clause
Resolution Proofs

SAT

- $A \triangleq \overline{pq}, pq$
- $B \triangleq q\overline{r}, qr$
Resolution Proofs

SAT

- $A \triangleq \{\overline{pq}, pq\}$  
  $B \triangleq \{q\overline{r}, qr\}$

- Proof of unsatisfiability

\[
\begin{array}{ccccccccc}
\text{pq} & \overline{pq} & p & q & \overline{r} & qr & r & q & \bot \\
\overline{q} & p & \overline{q} & p & q & r & qr & q & \bot \\
\end{array}
\]
• Computation of interpolant $I$ for $A \land B$ from proof structure
Interpolant Generation
SAT [Pudlák97]

- Computation of interpolant $I$ for $A \land B$ from proof structure

- Partial interpolant for leaf
Interpolant Generation

SAT [Pudlák97]

- Computation of interpolant $I$ for $A \land B$ from proof structure
- Partial interpolant for leaf
- Partial interpolant for resolvent
  - Pivot
  - Partial interpolants for antecedents
Interpolant Generation

SAT [Pudlák97]

- Computation of interpolant $I$ for $A \land B$ from proof structure
- Partial interpolant for leaf
- Partial interpolant for resolvent
  - Pivot
    - Partial interpolants for antecedents
- Partial interpolant for $\bot$ is $/$
Interpolant Generation

SAT [Pudlák97]

- \( A \triangleq \{ \overline{pq}, pq \} \)
- \( B \triangleq \{ q\overline{r}, qr \} \)

- Proof of unsatisfiability

\[
\begin{array}{cccccc}
\overline{pq} & p\overline{q} & p & q\overline{r} & qr & r \\
\hline
\overline{q} & & p &$q\overline{r}$ & qr & r \\
\hline
\overline{q} & & & & q & q \\
\hline
\bot & & & & & q
\end{array}
\]
Interpolant Generation
SAT [Pudlák97]

- \( A \triangleq \{ \overline{pq}, \overline{pq} \} \)
- \( B \triangleq \{ q\overline{r}, qr \} \)

- Proof of unsatisfiability

\[
\begin{array}{cccc}
\overline{pq} & \{ \bot \} & p\overline{q} & \{ \bot \} \\
& & \overline{q} & \\
& & & \bot
\end{array}
\]

\[
\begin{array}{cccc}
q\overline{r} & & qr & \\
& & q & \\
& & & q
\end{array}
\]

Natasha Sharygina (USI) Flexible Proof Transformation June 21, 2011 20 / 72
• \( A \triangleq \{ \overline{pq}, \overline{pq} \} \quad B \triangleq \{ q\overline{r}, qr \} \)

• Proof of unsatisfiability

\[
\begin{array}{ccc}
\overline{pq} \quad \{ \bot \} & \quad p\overline{q} \quad \{ \bot \} & \quad p & \quad q \overline{r} \quad \{ \top \} & \quad qr \quad \{ \top \} & \quad r \\
\overline{q} & \quad q & \quad \overline{q} & \quad \overline{q} & \quad q
\end{array}
\]
• \( A \triangleq \{ \overline{pq}, pq \} \quad B \triangleq \{ q\overline{r}, qr \} \)

• Proof of unsatisfiability

\[
\begin{align*}
\overline{pq} & \quad \{ \bot \} & pq & \quad \{ \bot \} \\
\hline
\overline{q} & \quad \{ \bot \lor \bot \} & p
\end{align*}
\]

\[
\begin{align*}
q\overline{r} & \quad \{ \top \} & qr & \quad \{ \top \} \\
\hline
q & & q
\end{align*}
\]
Interpolant Generation
SAT [Pudlák97]

- $A \triangleq \{ \overline{p}q, pq \}$  
- $B \triangleq \{ q\overline{r}, qr \}$

- Proof of unsatisfiability

\[
\begin{align*}
\overline{p}q & \{ \bot \} & pq & \{ \bot \} & p & \\hline
\overline{q} & \{ \bot \} & q & \\hline
\bot
\end{align*}
\]

\[
\begin{align*}
q\overline{r} & \{ \top \} & qr & \{ \top \} & r & \\hline
q & \\hline
q
\end{align*}
\]
Interpolant Generation
SAT [Pudlák97]

- $A \triangleq \{ \overline{pq}, pq \} \quad B \triangleq \{ q\overline{r}, qr \}$

- Proof of unsatisfiability

\[
\begin{array}{c}
\overline{pq} \{ \bot \} \quad pq \{ \bot \} \\
p \\
\overline{q} \{ \bot \} \\
\downarrow \\
\downarrow \\
\downarrow \\
\bot
\end{array}
\begin{array}{c}
q\overline{r} \{ \top \} \\
qr \{ \top \} \\
r \\
q \{ \top \wedge \top \}
\end{array}
\]
Interpolant Generation

SAT [Pudlák97]

- \( A \triangleq \{ \overline{pq}, pq \} \quad B \triangleq \{ q\bar{r}, qr \} \)

- Proof of unsatisfiability

\[
\begin{array}{cccccc}
\overline{pq} & \{ \bot \} & pq & \{ \bot \} & \overline{q} & \{ \bot \} \\
\hline
& & p & & q & \{ \top \} \\
\overline{q} & \{ \bot \} & & & qr & \{ \top \} \\
\hline
& & & & q & \{ \top \} \\
& & & \bot & & q
\end{array}
\]
Interpolant Generation

SAT [Pudlák97]

- $A \triangleq \{ \overline{pq}, pq \}$
- $B \triangleq \{ q\overline{r}, qr \}$

Proof of unsatisfiability:

\[
\begin{array}{c}
\overline{pq} \{ \bot \} \\
\overline{pq} \{ \bot \} \\
\overline{q} \{ \bot \} \\
\bot \{ (\bot \lor \overline{q}) \land (\top \lor q) \}
\end{array}
\]

\[
\begin{array}{c}
pq \{ \bot \} \\
\top \{ \top \} \\
q \{ \top \} \\
q \{ \top \} \\
q \{ \top \}
\end{array}
\]

Natasha Sharygina (USI)
• $A \triangleq \{\overline{pq}, pq\}$  \quad $B \triangleq \{q\overline{r}, qr\}$

• Proof of unsatisfiability

$$
\overline{pq} \quad \{\bot\} \quad p\overline{q} \quad \{\bot\} \quad \overline{q} \quad \{\bot\} \quad r
\overline{q} \quad \{\bot\} \quad q \quad \{\top\} \quad \overline{q} \quad \{\top\} \quad q
\bot \quad \{\overline{q}\} \quad \bot \quad \{\overline{q}\} \quad q \quad \{\top\} \quad \overline{q} \quad \{\top\} \quad q
$$
• $A \triangleq \{(5x - y \leq 1), (y - 5x \leq -1)\} \quad B \triangleq \{(y - 5z \leq 3), (5z - y \leq -2)\}$
Resolution Proofs

SMT

- \[ A \triangleq \{ (5x - y \leq 1), (y - 5x \leq -1) \} \]
- \[ B \triangleq \{ (y - 5z \leq 3), (5z - y \leq -2) \} \]

- Theory lemmata
Resolution Proofs

SMT

- \( A \triangleq \{ (5x - y \leq 1), \ (y - 5x \leq -1) \} \)

- \( B \triangleq \{ (y - 5z \leq 3), \ (5z - y \leq -2) \} \)

- Theory lemmata

  - LIA:
    - \( t \triangleq \{ (x - z \leq 0) \} \)
    - \( u \triangleq \{ (x - z \geq 1) \} \)
Resolution Proofs

SMT

- $A \triangleq \{ (5x - y \leq 1), (y - 5x \leq -1) \} \quad B \triangleq \{ (y - 5z \leq 3), (5z - y \leq -2) \}$

- Theory lemmata
  - **LIA:**
    - $t$: $(x - z \leq 0)$
    - $u$: $(x - z \geq 1)$
  - **LRA:**
    - $\bar{p}$: $(5x - y \not\leq 1)$
    - $\bar{r}$: $(y - 5z \not\leq 3)$
    - $\bar{u}$: $(x - z \not\geq 1)$
Resolution Proofs

SMT

\[ A \triangleq \{ (5x - y \leq 1), (y - 5x \leq -1) \} \]

\[ B \triangleq \{ (y - 5z \leq 3), (5z - y \leq -2) \} \]

• Theory lemmata

- LIA:
  \[ (x - z \leq 0), (x - z \geq 1) \]

- LRA:
  \[ (5x - y \not\leq 1), (y - 5z \not\leq 3), (x - z \not\geq 1) \]

  \[ (y - 5x \not\leq -1), (5z - y \not\leq -2), (x - z \not\geq 0) \]
• $A \triangleq \{p, q\}$  
  $B \triangleq \{r, s\}$  
  $L \triangleq \{tu, \overline{pru}, \overline{qst}\}$
Resolution Proofs

SMT

- \( A \triangleq \{ p, q \} \quad B \triangleq \{ r, s \} \quad L \triangleq \{ tu, \overline{pru}, \overline{qst} \} \)

- Proof of unsatisfiability

\[
\begin{align*}
p & \vdash \overline{pru} \\
& \vdash \overline{ru} \quad \vdash r \\
& \vdash \overline{u} \quad \vdash tu \\
& \vdash t \\
& \vdash \overline{qs} \\
& \vdash \overline{s} \\
& \vdash \bot
\end{align*}
\]
• $A \triangleq \{p, q\}$, $B \triangleq \{r, s\}$, $L \triangleq \{tu, \overline{pru}, \overline{qst}\}$

• Proof of unsatisfiability
Interpolant Generation
SMT

- $A \triangleq \{ p, q \}$  \hspace{1cm} $B \triangleq \{ r, s \}$  \hspace{1cm} $L \triangleq \{ tu, \overline{pru}, \overline{qst} \}$

- Proof of unsatisfiability

\[
\begin{align*}
p & \{ \bot \} & \overline{pru} \\
\overline{ru} & \quad r \\
\overline{u} & \quad tu \\
\overline{qst} & \quad t \\
\overline{qs} & \quad q \{ \bot \} \\
\overline{s} & \quad s \\
\bot & 
\end{align*}
\]
Interpolant Generation

SMT

- \( A \triangleq \{ p, q \} \) \quad \( B \triangleq \{ r, s \} \) \quad \( L \triangleq \{ t, \overline{pru}, \overline{qst} \} \)

- Proof of unsatisfiability

\[
\begin{align*}
\overline{p} & \{ \bot \} \\
& \overline{pru} \\
& \overline{r}u \\
& \{ \top \} \\
& r \\
& \overline{ru} \quad \overline{r} \\
& \{ \top \} \\
& u \\
& \overline{tu} \quad \overline{qst} \\
& t \\
& \overline{qs} \quad q \\
& \{ \bot \} \\
& s \\
& \overline{s} \quad q \\
& \{ \top \} \\
& s \\
& \{ \bot \}
\end{align*}
\]
Interpolant Generation

SMT

- \( A \triangleq \{ p, q \} \)
- \( B \triangleq \{ r, s \} \)
- \( L \triangleq \{ tu, \overline{pru}, \overline{qst} \} \)

- Proof of unsatisfiability

\[
\begin{align*}
p & \{ \bot \} & \overline{pru} \\
\hline
\overline{ru} & r & \{ \top \} \\
\overline{u} & t & \overline{qst} \\
\hline
\overline{qs} & q & \{ \bot \} \\
\overline{s} & s & \{ \top \} \\
\bot
\end{align*}
\]
Interpolation

Challenge

- State-of-the-art approach [Pudlák97, McMillan04]
Interpolation

Challenge

- State-of-the-art approach [Pudlák97, McMillan04]
  - Derivation of unsatisfiability proof of $A \land B$
  - Computation of interpolant from proof structure in linear time

Natasha Sharygina (USI) Flexible Proof Transformation June 21, 2011
Interpolation

Challenge

- State-of-the-art approach [Pudlák97, McMillan04]
  - Derivation of unsatisfiability proof of $A \land B$
  - Computation of interpolant from proof structure in linear time

- Restriction
Interpolation

Challenge

- State-of-the-art approach [Pudlák97, McMillan04]
  - Derivation of unsatisfiability proof of $A \land B$
  - Computation of interpolant from proof structure in linear time

- Restriction
  - Need for proof not to contain AB-mixed predicates

<table>
<thead>
<tr>
<th>A-local</th>
<th>B-local</th>
<th>AB-common</th>
<th>AB-mixed</th>
</tr>
</thead>
</table>

Natasha Sharygina (USI) Flexible Proof Transformation June 21, 2011 24 / 72
Interpolation

Challenge

- State-of-the-art approach [Pudlák97, McMillan04]
  - Derivation of unsatisfiability proof of $A \land B$
  - Computation of interpolant from proof structure in linear time

- Restriction
  - Need for proof not to contain AB-mixed predicates

\[
A \triangleq \{ (5x - y \leq 1), \ldots \} \\
B \triangleq \{ (y - 5z \leq 3), \ldots \}
\]
Interpolation
Challenge

- State-of-the-art approach [Pudlák97, McMillan04]
  - Derivation of unsatisfiability proof of $A \land B$
  - Computation of interpolant from proof structure in linear time

- Restriction
  - Need for proof not to contain AB-mixed predicates

\[
\begin{align*}
A & \triangleq \{ (5x - y \leq 1), \ldots \} \\
B & \triangleq \{ (y - 5z \leq 3), \ldots \} \\
L & \triangleq \{ (x - z \leq 0), \ldots \}
\end{align*}
\]
• Need for proof not to contain AB-mixed predicates
Interpolation

Possible Solutions

- Need for proof not to contain AB-mixed predicates

- Tune solvers to avoid generating AB-mixed predicates
  [Cimatti08, Beyer08]
Interpolation
Possible Solutions

• Need for proof not to contain AB-mixed predicates

• Tune solvers to avoid generating AB-mixed predicates [Cimatti08, Beyer08]

• Transform proof to remove AB-mixed predicates
• Proof transformation approach
Proof transformation approach

Motivation: more flexibility by decoupling SMT solving and interpolant generation
Proof transformation approach

Motivation: more flexibility by decoupling SMT solving and interpolant generation

Motivation: standard SMT techniques can require addition of AB-mixed predicates
Proof Transformation

Motivation

• Proof transformation approach

• Motivation: more flexibility by decoupling SMT solving and interpolant generation

• Motivation: standard SMT techniques can require addition of AB-mixed predicates

  • Theory reduction via Lemma on Demand [DeMoura02, Barrett06]
    Reduction of AX to EUF
    Reduction of LIA to LRA
    Ackermann’s Expansion

  • Theory combination via DTC [Bozzano05]
Outline

1. Background

2. Motivation and Related Work

3. Contribution
   - Proof Transformation for Interpolation and Reduction

4. Summary and Future Work
Outline

1. Background

2. Motivation and Related Work

3. Contribution
   - Proof Transformation for Interpolation and Reduction

4. Summary and Future Work
• Proof rewriting framework based on local rules
• Proof rewriting framework based on local rules

• Isolation of AB-mixed predicates into subtrees
Contribution
Proof Transformation Framework

- Proof rewriting framework based on local rules
- Isolation of AB-mixed predicates into subtrees
- Removal of AB-mixed subtrees
Contribution
Proof Transformation Framework

- Proof rewriting framework based on local rules
- Isolation of AB-mixed predicates into subtrees
- Removal of AB-mixed subtrees
- No more AB-mixed predicates, proof still valid
Proof Transformation

Effect

(a) Initial proof: A-local, B-local, AB-common, AB-mixed
(b) Transformed proof: AB-mixed predicates isolated into subtrees
(c) Final proof: AB-mixed subtrees removed, new leaves are theory lemmata
Proof Transformation

Advantages

- No more AB-mixed predicates, new leaves are theory lemmata
Proof Transformation

Advantages

- No more AB-mixed predicates, new leaves are theory lemmata
- Easy combination of SMT and interpolation techniques
Advantages

- No more AB-mixed predicates, new leaves are theory lemmata
- Easy combination of SMT and interpolation techniques
  - Theory reduction, theory combination without restrictions
Proof Transformation

Advantages

- No more AB-mixed predicates, new leaves are theory lemmata
- Easy combination of SMT and interpolation techniques
  - Theory reduction, theory combination without restrictions
  - Interpolant generation for propositional resolution proofs of unsatisfiability [Pudlák97]
Proof Transformation

Advantages

- No more AB-mixed predicates, new leaves are theory lemmata
- Easy combination of SMT and interpolation techniques
  - Theory reduction, theory combination without restrictions
  - Interpolant generation for propositional resolution proofs of unsatisfiability [Pudlák97]
  - (Partial) interpolant generation for theory (combination) lemmata [Yorsh05]
Proof Transformation Framework

Features

- Local rewriting rules
Proof Transformation Framework

Features

- Local rewriting rules
- Rule context

\[
\begin{array}{c}
\text{pqC} \quad \bar{p}D \\
\hline
qCD \\
\bar{q}E \\
\hline
CDE
\end{array}
\]
Proof Transformation Framework

Features

- Local rewriting rules

- Rule context

- Exhaustiveness up to symmetry
Proof Transformation Framework
Local Rewriting Rules

\[ \begin{array}{c}
\frac{pqC\quad \bar{p}D}{q\quad \bar{q}E\quad p} \quad \Rightarrow \quad \frac{pqC\quad \bar{q}E}{q\quad pCE\quad \bar{p}D} \\
qCD\quad \bar{q}E\quad q
\end{array} \]
Proof Transformation Framework
Local Rewriting Rules

- \(pqC\), \(\overline{p}D\)  \(\overline{q}E\)  \(q\)  \(\Rightarrow\)  
  \(CDE\)

- Pivots swapping
• Pivots swapping

• AB-mixed predicates isolation into subtrees
Reduction LIA to LRA

Transformation

- \( A \equiv \{ p, q \} \quad B \equiv \{ r, s \} \quad L \equiv \{ tu, \overline{pru}, \overline{qst} \} \)

- Proof of unsatisfiability
Proof of unsatisfiability
• Proof of unsatisfiability
• Proof of unsatisfiability
• Proof of unsatisfiability
• Proof of unsatisfiability
Proof of unsatisfiability
- Proof of unsatisfiability
• Proof of unsatisfiability
Reduction LIA to LRA

Transformation

- Proof of unsatisfiability
Proof of unsatisfiability
• Proof of unsatisfiability
• Proof of unsatisfiability
Proof of unsatisfiability
• Proof of unsatisfiability
• Proof of unsatisfiability
• Potential drawbacks
Potential drawbacks

Overhead w.r.t. solving time
Proof Transformation Framework

Considerations

• Potential drawbacks

  • Overhead w.r.t. solving time

  • Increase of proof size
Transformation Framework

Features

- Local rewriting rules
Transformation Framework

Features

- Local rewriting rules
  - B reduction
  - A perturbation
Transformation Framework

Features

- Local rewriting rules
  - B reduction
  - A perturbation

- Rule context

\[
\begin{array}{c}
\text{pqC} \\
\hline
\text{p}D \\
\hline
\text{qCD} \\
\hline
\text{qE} \\
\hline
\text{CDE}
\end{array}
\]
Transformation Framework

Features

• Local rewriting rules
  • B reduction
  • A perturbation

• Rule context

\[
\frac{pqC \quad \bar{p}D}{qCD \quad \bar{q}E} \quad p \quad q
\]

• Exhaustiveness up to symmetry
• **B rules**

\[
\begin{array}{c|c}
\text{B1} & \frac{pqC \quad \neg pqD}{qCD \quad p} \quad \frac{p\neg E}{pE \quad q} \\
                & \frac{pCDE}{\Rightarrow} \\
\end{array}
\]

\[
\frac{pqC \quad p\neg E}{pCE \quad q}
\]
**B rules**

| B1 | $\frac{pqC}{qCD} \frac{\overline{pq}D}{p} \frac{\overline{pq}E}{q} \Rightarrow \frac{pqC}{pCE} \frac{\overline{pq}E}{q} \frac{pqC}{pCE} $ |

- Redundancy as reintroduction variable after elimination
### B rules

| $B1$ | $\begin{array}{c} pqC \\ \bar{p}qD \\ qC \bar{D} \\ \bar{C}DE \\ pCE \\ q \end{array}$ | $\Rightarrow$ | $\begin{array}{c} \bar{p}qC \\ p\bar{q}E \\ q \end{array}$ |

- Redundancy as reintroduction variable after elimination
- Subproof simplification
Transformation Framework

Local rewriting rules

- B rules

| B1 | \[ \frac{pqC \quad \bar{p}qD}{qCD \quad p\bar{q}E} p \Rightarrow \frac{pqC \quad p\bar{q}E}{pCDE} q ] |

- Redundancy as reintroduction variable after elimination

- Subproof simplification

- Subproof root strengthening
Transformation Framework

Local rewriting rules

- **A rules**

<table>
<thead>
<tr>
<th>A2</th>
<th>$\frac{pqC}{qCD} \frac{\overline{p}D}{\overline{q}E} \frac{p}{q} \Rightarrow \frac{pqC}{pCE} \frac{\overline{q}E}{\overline{p}D} \frac{q}{CDE} p$</th>
</tr>
</thead>
</table>
Transformation Framework

Local rewriting rules

- **A rules**

\[
\begin{array}{c|c}
\text{A2} & \begin{array}{c}
\frac{pqC}{qCD} & \frac{\overline{p}D}{p} \\
\frac{\overline{q}E}{q} & \Rightarrow \\
\frac{CDE}{q}\end{array} & \begin{array}{c}
\frac{pqC}{pCE} & \frac{\overline{q}E}{q} \\
\frac{\overline{p}D}{p} & \frac{CDE}{q}\end{array}
\end{array}
\]

- **Pivots swapping**
Transformation Framework

Local rewriting rules

- A rules

\[
\begin{array}{c|c}
A2 & \frac{pqC \quad \overline{p}D}{qCD} \quad p \\
 & \frac{\overline{q}E}{q} \quad \Rightarrow \\
 & \frac{CDE}{pCE} \quad \frac{qCDE}{\overline{p}D} \quad q \\
& \frac{pCE}{pCE} \quad q \\
& \frac{CDE}{CDE} \quad p
\end{array}
\]

- Pivots swapping

- Topology perturbation
### Transformation Framework

#### Local rewriting rules

- **A rules**

| A2 | \[
\begin{align*}
& pqC \quad \overline{p}D \\
& qCD \quad \overline{q}E \\
& CDE
\end{align*}
\]  
\[
\Rightarrow
\begin{align*}
& pqC \quad \overline{q}E \\
& pCE \quad \overline{p}D \\
& CDE
\end{align*}
\] |

- **Pivots swapping**

- **Topology perturbation**

- **Redundancies exposure**
### Local rewriting rules

<table>
<thead>
<tr>
<th>Rule</th>
<th>Premises</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A1</strong></td>
<td>(pqC \quad \neg qD \quad p) \quad qCD \quad \neg qE \quad q \quad \Rightarrow \quad pqC \quad \neg qE \quad \neg qE \quad \neg qD \quad q) \quad pCE \quad \neg pDE \quad p</td>
<td>CDE \quad pCE \quad \neg pDE \quad p</td>
</tr>
<tr>
<td><strong>A2</strong></td>
<td>(pqC \quad \neg pD \quad p) \quad qCD \quad \neg qE \quad q \quad \Rightarrow \quad pqC \quad \neg qE \quad \neg qE \quad pCE \quad p \quad \neg pD \quad p</td>
<td>CDE \quad \neg pE \quad pCE \quad \neg pD \quad p</td>
</tr>
<tr>
<td><strong>B1</strong></td>
<td>(pqC \quad \neg pD \quad p) \quad qCD \quad \neg p\neg qE \quad q \quad \Rightarrow \quad pqC \quad \neg p\neg qE \quad \neg p\neg qE \quad pCE \quad \neg pCE \quad q</td>
<td>pCDE \quad pCE \quad \neg pCE \quad q</td>
</tr>
<tr>
<td><strong>B2</strong></td>
<td>(pqC \quad \neg pD \quad p) \quad qDC \quad \neg p\neg qE \quad q \quad \Rightarrow \quad pqC \quad \neg p\neg qE \quad \neg p\neg qE \quad pCE \quad \neg pCE \quad \neg pD \quad p</td>
<td>pCDE \quad pCE \quad \neg pCE \quad \neg pD \quad p</td>
</tr>
<tr>
<td><strong>B2'</strong></td>
<td>(pqC \quad \neg pD \quad p) \quad qDC \quad \neg p\neg qE \quad q \quad \Rightarrow \quad pqC \quad \neg p\neg qE \quad \neg p\neg qE \quad pCE \quad \neg pCE \quad \neg pD \quad p</td>
<td>pCDE \quad pCE \quad \neg pCE \quad \neg pD \quad p</td>
</tr>
<tr>
<td><strong>B3</strong></td>
<td>(pqC \quad \neg pD \quad p) \quad qCD \quad \neg p\neg qE \quad q \quad \Rightarrow \quad \neg pD</td>
<td>pCDE \quad \neg pCE \quad \neg pCE \quad \neg pD</td>
</tr>
</tbody>
</table>
• opensmt
Evaluation
Framework and Benchmarks

- **opensmt**
  - C++ open-source SMT solver developed at USI
  - Fastest open-source solver in SMT-comp 2009, 2010 for various logics
• **opensmt**
  
  • C++ open-source SMT solver developed at USI
  
  • Fastest open-source solver in SMT-comp 2009, 2010 for various logics

• Benchmarks
• **opensmt**
  - C++ open-source SMT solver developed at USI
  - Fastest open-source solver in SMT-comp 2009, 2010 for various logics

• Benchmarks
  - SMT: SMT-LIB library
  - Academic and industrial problems
## Evaluation

### Experimental results over QF_UFIDL

<table>
<thead>
<tr>
<th>Group</th>
<th>#</th>
<th>#AB</th>
<th>%&lt;sub&gt;time&lt;/sub&gt;</th>
<th>%&lt;sub&gt;nodes&lt;/sub&gt;</th>
<th>%&lt;sub&gt;edges&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDS</td>
<td>2</td>
<td>7</td>
<td>93%</td>
<td>2%</td>
<td>2%</td>
</tr>
<tr>
<td>EufLaAr</td>
<td>2</td>
<td>103</td>
<td>91%</td>
<td>30%</td>
<td>26%</td>
</tr>
<tr>
<td>pete</td>
<td>6</td>
<td>4</td>
<td>33%</td>
<td>8%</td>
<td>9%</td>
</tr>
<tr>
<td>pete2</td>
<td>56</td>
<td>17</td>
<td>59%</td>
<td>27%</td>
<td>32%</td>
</tr>
<tr>
<td>uclid</td>
<td>8</td>
<td>11</td>
<td>64%</td>
<td>37%</td>
<td>42%</td>
</tr>
<tr>
<td>Overall</td>
<td>74</td>
<td>17</td>
<td>59%</td>
<td>26%</td>
<td>30%</td>
</tr>
</tbody>
</table>

- # — number of benchmarks solved
- #<sub>AB</sub> — average number of AB-mixed predicates in proof
- %<sub>time</sub> — average time overhead
- %<sub>nodes</sub>, %<sub>edges</sub> — average difference in proof size
Comparison

- RecyclePivots (closest related work) [Strichman’08]

  - Pros
    - Global information
    - Fast and effective
  
  - Cons
    - Cannot expose redundancies

  - Rule-based approach

    - Pros
      - Flexibility in rules application
      - Flexibility in amount of transformation
      - Can expose redundancies

    - Cons
      - Local information
Comparison

- RecyclePivots (closest related work) [Strichman’08]
  - **Pros**
    - Global information
    - Fast and effective
  - **Cons**
    - Cannot expose redundancies
Comparison

- RecyclePivots (closest related work) [Strichman’08]
  - **Pros**
    - Global information
    - Fast and effective
  - **Cons**
    - Cannot expose redundancies

- Rule-based approach
Comparison

- RecyclePivots (closest related work) [Strichman’08]
  - **Pros**
    - Global information
    - Fast and effective
  - **Cons**
    - Cannot expose redundancies

- Rule-based approach
  - **Pros**
    - Flexibility in rules application
    - Flexibility in amount of transformation
    - Can expose redundancies
  - **Cons**
    - Local information
• Based on a sequence of proof traversals (e.g. topological order)
• Based on a sequence of proof traversals (e.g. topological order)

• Parameterized in number of traversals and time limit
- Based on a sequence of proof traversals (e.g. topological order)
- Parameterized in number of traversals and time limit
- Examination non-leaf clauses
Implementation
Reduction Algorithm

• Based on a sequence of proof traversals (e.g. topological order)

• Parameterized in number of traversals and time limit

• Examination non-leaf clauses
  
  • Pivot in both antecedents $\rightarrow$ update, match context, apply rule

\[
\frac{qC'D'}{CDE} \frac{\overline{qE'}}{q} \Rightarrow \frac{qC'D'}{C'D'E'} \frac{\overline{qE'}}{q} \Rightarrow \frac{pqC'}{qC'D'} \frac{\overline{pD'}}{C'D'E'} \frac{\overline{qE'}}{q}
\]
Implementation
Reduction Algorithm

- Based on a sequence of proof traversals (e.g. topological order)
- Parameterized in number of traversals and time limit
- Examination non-leaf clauses
  - Pivot in both antecedents → update, match context, apply rule
    \[
    \frac{qC'D'}{CDE} \quad \frac{\overline{qE'}}{q} \quad \Rightarrow \quad \frac{qC'D'}{C'D'E'} \quad \frac{\overline{qE'}}{q} \quad \Rightarrow \quad \frac{p\overline{qC'}}{qC'D'} \quad \frac{\overline{pD'}}{C'D'E'} \quad \frac{\overline{qE'}}{q}
    \]
  - Pivot not in both antecedents → remove resolution step
    \[
    \frac{C'D'}{CDE} \quad \frac{\overline{qE'}}{q} \quad \Rightarrow \quad C'D'
    \]
Implementation
Reduction Algorithm

- Based on a sequence of proof traversals (e.g. topological order)
- Parameterized in number of traversals and time limit
- Examination non-leaf clauses
  - Pivot in both antecedents $\rightarrow$ update, match context, apply rule
    $$\frac{qC'D'}{CDE} \frac{\overline{q}E'}{q} \Rightarrow \frac{qC'D'}{C'D'E'} \frac{\overline{q}E'}{q} \Rightarrow \frac{pqC'}{qC'D'} \frac{\overline{p}D'}{p} \frac{\overline{q}E'}{q}$$
  - Pivot not in both antecedents $\rightarrow$ remove resolution step
    $$\frac{C'D'}{CDE} \frac{\overline{q}E'}{q} \Rightarrow C'D'$$
- Easy combination with RecyclePivots
Evaluation
Framework and Benchmarks

- Implemented in C++ and integrated with OpenSMT
- Available at www.inf.usi.ch/phd/rollini/hvc.html
• Implemented in C++ and integrated with OpenSMT

• Available at www.inf.usi.ch/phd/rollini/hvc.html

• Benchmarks
• Implemented in C++ and integrated with OpenSMT

• Available at www.inf.usi.ch/phd/rollini/hvc.html

• Benchmarks
  • SMT: SMT-LIB library
  • SAT: SAT competition
  • Academic and industrial problems
### Combined Approach Evaluation

**Experimental results over SMT: QF-UF, QF_IDL, QF_LRA, QF_RDL**

<table>
<thead>
<tr>
<th></th>
<th>#</th>
<th>Avg_nodes</th>
<th>Avg_edges</th>
<th>Avg_core</th>
<th>T(s)</th>
<th>Max_nodes</th>
<th>Max_edges</th>
<th>Max_core</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP</td>
<td>1370</td>
<td>6.7%</td>
<td>7.5%</td>
<td>1.3%</td>
<td>1.7</td>
<td>65.1%</td>
<td>68.9%</td>
<td>39.1%</td>
</tr>
<tr>
<td>Ratio</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>1366</td>
<td>8.9%</td>
<td>10.7%</td>
<td>1.4%</td>
<td>3.4</td>
<td>66.3%</td>
<td>70.2%</td>
<td>45.7%</td>
</tr>
<tr>
<td>0.025</td>
<td>1366</td>
<td>9.8%</td>
<td>11.9%</td>
<td>1.5%</td>
<td>3.6</td>
<td>77.2%</td>
<td>79.9%</td>
<td>45.7%</td>
</tr>
<tr>
<td>0.05</td>
<td>1366</td>
<td>10.7%</td>
<td>13.0%</td>
<td>1.6%</td>
<td>4.1</td>
<td>78.5%</td>
<td>81.2%</td>
<td>45.7%</td>
</tr>
<tr>
<td>0.075</td>
<td>1366</td>
<td>11.4%</td>
<td>13.8%</td>
<td>1.7%</td>
<td>4.5</td>
<td>78.5%</td>
<td>81.2%</td>
<td>45.7%</td>
</tr>
<tr>
<td>0.1</td>
<td>1364</td>
<td>11.8%</td>
<td>14.4%</td>
<td>1.7%</td>
<td>5.0</td>
<td>78.8%</td>
<td>83.6%</td>
<td>45.7%</td>
</tr>
<tr>
<td>0.25</td>
<td>1359</td>
<td>13.6%</td>
<td>16.6%</td>
<td>1.9%</td>
<td>7.6</td>
<td>79.6%</td>
<td>84.4%</td>
<td>45.7%</td>
</tr>
<tr>
<td>0.5</td>
<td>1348</td>
<td>15.0%</td>
<td>18.4%</td>
<td>2.0%</td>
<td>11.5</td>
<td>79.1%</td>
<td>85.2%</td>
<td>45.7%</td>
</tr>
<tr>
<td>0.75</td>
<td>1341</td>
<td>16.0%</td>
<td>19.5%</td>
<td>2.1%</td>
<td>15.1</td>
<td>79.9%</td>
<td>86.1%</td>
<td>45.7%</td>
</tr>
<tr>
<td>1</td>
<td>1337</td>
<td>16.7%</td>
<td>20.4%</td>
<td>2.2%</td>
<td>18.8</td>
<td>79.9%</td>
<td>86.1%</td>
<td>45.7%</td>
</tr>
</tbody>
</table>

- **Ratio** — time threshold as fraction of solving time
- **#** — number of benchmarks solved
- **Avg\_nodes, Avg\_edges, Avg\_core** — average reduction in proof size
- **T(s)** — average transformation time in seconds
- **Max\_nodes, Max\_edges, Max\_core** — max reduction in proof size
### Combined Approach Evaluation

Experimental results over SMT: QF_UF, QF_IDL, QF_LRA, QF_RDL

<table>
<thead>
<tr>
<th></th>
<th>#</th>
<th>Avg_nodes</th>
<th>Avg_edges</th>
<th>Avg_core</th>
<th>T(s)</th>
<th>Max_nodes</th>
<th>Max_edges</th>
<th>Max_core</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP</td>
<td>1370</td>
<td>6.7%</td>
<td>7.5%</td>
<td>1.3%</td>
<td>1.7</td>
<td>65.1%</td>
<td>68.9%</td>
<td>39.1%</td>
</tr>
<tr>
<td>Ratio</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>1366</td>
<td>8.9%</td>
<td>10.7%</td>
<td>1.4%</td>
<td>3.4</td>
<td>66.3%</td>
<td>70.2%</td>
<td>45.7%</td>
</tr>
<tr>
<td>0.025</td>
<td>1366</td>
<td>9.8%</td>
<td>11.9%</td>
<td>1.5%</td>
<td>3.6</td>
<td>77.2%</td>
<td>79.9%</td>
<td>45.7%</td>
</tr>
<tr>
<td>0.05</td>
<td>1366</td>
<td>10.7%</td>
<td>13.0%</td>
<td>1.6%</td>
<td>4.1</td>
<td>78.5%</td>
<td>81.2%</td>
<td>45.7%</td>
</tr>
<tr>
<td>0.075</td>
<td>1366</td>
<td>11.4%</td>
<td>13.8%</td>
<td>1.7%</td>
<td>4.5</td>
<td>78.5%</td>
<td>81.2%</td>
<td>45.7%</td>
</tr>
<tr>
<td>0.1</td>
<td>1364</td>
<td>11.8%</td>
<td>14.4%</td>
<td>1.7%</td>
<td>5.0</td>
<td>78.8%</td>
<td>83.6%</td>
<td>45.7%</td>
</tr>
<tr>
<td>0.25</td>
<td>1359</td>
<td>13.6%</td>
<td>16.6%</td>
<td>1.9%</td>
<td>7.6</td>
<td>79.6%</td>
<td>84.4%</td>
<td>45.7%</td>
</tr>
<tr>
<td>0.5</td>
<td>1348</td>
<td>15.0%</td>
<td>18.4%</td>
<td>2.0%</td>
<td>11.5</td>
<td>79.1%</td>
<td>85.2%</td>
<td>45.7%</td>
</tr>
<tr>
<td>0.75</td>
<td>1341</td>
<td>16.0%</td>
<td>19.5%</td>
<td>2.1%</td>
<td>15.1</td>
<td>79.9%</td>
<td>86.1%</td>
<td>45.7%</td>
</tr>
<tr>
<td>1</td>
<td>1337</td>
<td>16.7%</td>
<td>20.4%</td>
<td>2.2%</td>
<td>18.8</td>
<td>79.9%</td>
<td>86.1%</td>
<td>45.7%</td>
</tr>
</tbody>
</table>

- **Ratio** — time threshold as fraction of solving time
- **#** — number of benchmarks solved
- **Avg\_nodes, Avg\_edges, Avg\_core** — average reduction in proof size
- **T(s)** — average transformation time in seconds
- **Max\_nodes, Max\_edges, Max\_core** — max reduction in proof size
### Combined Approach Evaluation

**Experimental results over SMT: QF_UF, QF_IDL, QF_LRA, QF_RDL**

<table>
<thead>
<tr>
<th>#</th>
<th>Avg_nodes</th>
<th>Avg_edges</th>
<th>Avg_core</th>
<th>T(s)</th>
<th>Max_nodes</th>
<th>Max_edges</th>
<th>Max_core</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP</td>
<td>1370</td>
<td>6.7%</td>
<td>7.5%</td>
<td>1.3%</td>
<td>1.7</td>
<td>65.1%</td>
<td>68.9%</td>
</tr>
<tr>
<td>Ratio</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>1366</td>
<td>8.9%</td>
<td>10.7%</td>
<td>1.4%</td>
<td>3.4</td>
<td>66.3%</td>
<td>70.2%</td>
</tr>
<tr>
<td>0.025</td>
<td>1366</td>
<td>9.8%</td>
<td>11.9%</td>
<td>1.5%</td>
<td>3.6</td>
<td>77.2%</td>
<td>79.9%</td>
</tr>
<tr>
<td>0.05</td>
<td>1366</td>
<td>10.7%</td>
<td>13.0%</td>
<td>1.6%</td>
<td>4.1</td>
<td>78.5%</td>
<td>81.2%</td>
</tr>
<tr>
<td>0.075</td>
<td>1366</td>
<td>11.4%</td>
<td>13.8%</td>
<td>1.7%</td>
<td>4.5</td>
<td>78.5%</td>
<td>81.2%</td>
</tr>
<tr>
<td>0.1</td>
<td>1364</td>
<td>11.8%</td>
<td>14.4%</td>
<td>1.7%</td>
<td>5.0</td>
<td>78.8%</td>
<td>83.6%</td>
</tr>
<tr>
<td>0.25</td>
<td>1359</td>
<td>13.6%</td>
<td>16.6%</td>
<td>1.9%</td>
<td>7.6</td>
<td>79.6%</td>
<td>84.4%</td>
</tr>
<tr>
<td>0.5</td>
<td>1348</td>
<td>15.0%</td>
<td>18.4%</td>
<td>2.0%</td>
<td>11.5</td>
<td>79.1%</td>
<td>85.2%</td>
</tr>
<tr>
<td>0.75</td>
<td>1341</td>
<td>16.0%</td>
<td>19.5%</td>
<td>2.1%</td>
<td>15.1</td>
<td>79.9%</td>
<td>86.1%</td>
</tr>
<tr>
<td>1</td>
<td>1337</td>
<td>16.7%</td>
<td>20.4%</td>
<td>2.2%</td>
<td>18.8</td>
<td>79.9%</td>
<td>86.1%</td>
</tr>
</tbody>
</table>

- **Ratio** — time threshold as fraction of solving time
- **#** — number of benchmarks solved
- **Avg\_nodes, Avg\_edges, Avg\_core** — average reduction in proof size
- **T(s)** — average transformation time in seconds
- **Max\_nodes, Max\_edges, Max\_core** — max reduction in proof size
## Combined Approach Evaluation

Experimental results over SAT

<table>
<thead>
<tr>
<th>#</th>
<th>Avg_nodes</th>
<th>Avg_edges</th>
<th>Avg_core</th>
<th>T(s)</th>
<th>Max_nodes</th>
<th>Max_edges</th>
<th>Max_core</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP</td>
<td>25</td>
<td>5.9%</td>
<td>6.5%</td>
<td>1.7%</td>
<td>10.8</td>
<td>33.1%</td>
<td>33.4%</td>
</tr>
<tr>
<td>Ratio</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>25</td>
<td>6.8%</td>
<td>7.9%</td>
<td>1.7%</td>
<td>32.3</td>
<td>34.0%</td>
<td>34.4%</td>
</tr>
<tr>
<td>0.025</td>
<td>25</td>
<td>6.8%</td>
<td>7.9%</td>
<td>1.7%</td>
<td>32.3</td>
<td>34.0%</td>
<td>34.4%</td>
</tr>
<tr>
<td>0.05</td>
<td>25</td>
<td>7.0%</td>
<td>8.2%</td>
<td>1.8%</td>
<td>40.0</td>
<td>34.0%</td>
<td>34.4%</td>
</tr>
<tr>
<td>0.075</td>
<td>25</td>
<td>7.2%</td>
<td>8.4%</td>
<td>1.8%</td>
<td>49.3</td>
<td>34.7%</td>
<td>35.1%</td>
</tr>
<tr>
<td>0.1</td>
<td>25</td>
<td>7.3%</td>
<td>8.4%</td>
<td>1.8%</td>
<td>60.2</td>
<td>34.7%</td>
<td>35.1%</td>
</tr>
<tr>
<td>0.25</td>
<td>25</td>
<td>7.6%</td>
<td>8.8%</td>
<td>1.9%</td>
<td>125.3</td>
<td>39.8%</td>
<td>40.6%</td>
</tr>
<tr>
<td>0.5</td>
<td>25</td>
<td>7.8%</td>
<td>9.1%</td>
<td>1.9%</td>
<td>243.5</td>
<td>41.0%</td>
<td>41.9%</td>
</tr>
<tr>
<td>0.75</td>
<td>25</td>
<td>7.9%</td>
<td>9.3%</td>
<td>1.9%</td>
<td>360.0</td>
<td>41.6%</td>
<td>42.6%</td>
</tr>
<tr>
<td>1</td>
<td>23</td>
<td>8.4%</td>
<td>9.9%</td>
<td>2.1%</td>
<td>175.6</td>
<td>33.1%</td>
<td>33.4%</td>
</tr>
</tbody>
</table>

- **Ratio** — time threshold as fraction of solving time
- **#** — number of benchmarks solved
- **Avg\_nodes, Avg\_edges, Avg\_core** — average reduction in proof size
- **T(s)** — average transformation time in seconds
- **Max\_nodes, Max\_edges, Max\_core** — max reduction in proof size
## Combined Approach Evaluation

Experimental results over SAT

<table>
<thead>
<tr>
<th></th>
<th>#</th>
<th>Avg_nodes</th>
<th>Avg_edges</th>
<th>Avg_core</th>
<th>T(s)</th>
<th>Max_nodes</th>
<th>Max_edges</th>
<th>Max_core</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP</td>
<td>25</td>
<td>5.9%</td>
<td>6.5%</td>
<td>1.7%</td>
<td>10.8</td>
<td>33.1%</td>
<td>33.4%</td>
<td>30.3%</td>
</tr>
<tr>
<td>Ratio</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>25</td>
<td>6.8%</td>
<td>7.9%</td>
<td>1.7%</td>
<td>32.3</td>
<td>34.0%</td>
<td>34.4%</td>
<td>30.5%</td>
</tr>
<tr>
<td>0.025</td>
<td>25</td>
<td>6.8%</td>
<td>7.9%</td>
<td>1.7%</td>
<td>32.3</td>
<td>34.0%</td>
<td>34.4%</td>
<td>30.5%</td>
</tr>
<tr>
<td>0.05</td>
<td>25</td>
<td>7.0%</td>
<td>8.2%</td>
<td>1.8%</td>
<td>40.0</td>
<td>34.0%</td>
<td>34.4%</td>
<td>30.5%</td>
</tr>
<tr>
<td>0.075</td>
<td>25</td>
<td>7.2%</td>
<td>8.4%</td>
<td>1.8%</td>
<td>49.3</td>
<td>34.7%</td>
<td>35.1%</td>
<td>30.5%</td>
</tr>
<tr>
<td>0.1</td>
<td>25</td>
<td>7.3%</td>
<td>8.4%</td>
<td>1.8%</td>
<td>60.2</td>
<td>34.7%</td>
<td>35.1%</td>
<td>30.5%</td>
</tr>
<tr>
<td>0.25</td>
<td>25</td>
<td>7.6%</td>
<td>8.8%</td>
<td>1.9%</td>
<td>125.3</td>
<td>39.8%</td>
<td>40.6%</td>
<td>31.7%</td>
</tr>
<tr>
<td>0.5</td>
<td>25</td>
<td>7.8%</td>
<td>9.1%</td>
<td>1.9%</td>
<td>243.5</td>
<td>41.0%</td>
<td>41.9%</td>
<td>32.1%</td>
</tr>
<tr>
<td>0.75</td>
<td>25</td>
<td>7.9%</td>
<td>9.3%</td>
<td>1.9%</td>
<td>360.0</td>
<td>41.6%</td>
<td>42.6%</td>
<td>32.1%</td>
</tr>
<tr>
<td>1</td>
<td>23</td>
<td>8.4%</td>
<td>9.9%</td>
<td>2.1%</td>
<td>175.6</td>
<td>33.1%</td>
<td>33.4%</td>
<td>30.6%</td>
</tr>
</tbody>
</table>

- **Ratio** — time threshold as fraction of solving time
- **#** — number of benchmarks solved
- **Avg\_nodes, Avg\_edges, Avg\_core** — average reduction in proof size
- **T(s)** — average transformation time in seconds
- **Max\_nodes, Max\_edges, Max\_core** — max reduction in proof size
### Combined Approach Evaluation

**Experimental results over SAT**

<table>
<thead>
<tr>
<th>Ratio</th>
<th>#</th>
<th>$\text{Avg}_{\text{nodes}}$</th>
<th>$\text{Avg}_{\text{edges}}$</th>
<th>$\text{Avg}_{\text{core}}$</th>
<th>$T(s)$</th>
<th>$\text{Max}_{\text{nodes}}$</th>
<th>$\text{Max}_{\text{edges}}$</th>
<th>$\text{Max}_{\text{core}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP</td>
<td>25</td>
<td>5.9%</td>
<td>6.5%</td>
<td>1.7%</td>
<td>10.8</td>
<td>33.1%</td>
<td>33.4%</td>
<td>30.3%</td>
</tr>
<tr>
<td>Ratio</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>25</td>
<td>6.8%</td>
<td>7.9%</td>
<td>1.7%</td>
<td>32.3</td>
<td>34.0%</td>
<td>34.4%</td>
<td>30.5%</td>
</tr>
<tr>
<td>0.025</td>
<td>25</td>
<td>6.8%</td>
<td>7.9%</td>
<td>1.7%</td>
<td>32.3</td>
<td>34.0%</td>
<td>34.4%</td>
<td>30.5%</td>
</tr>
<tr>
<td>0.05</td>
<td>25</td>
<td>7.0%</td>
<td>8.2%</td>
<td>1.8%</td>
<td>40.0</td>
<td>34.0%</td>
<td>34.4%</td>
<td>30.5%</td>
</tr>
<tr>
<td>0.075</td>
<td>25</td>
<td>7.2%</td>
<td>8.4%</td>
<td>1.8%</td>
<td>49.3</td>
<td>34.7%</td>
<td>35.1%</td>
<td>30.5%</td>
</tr>
<tr>
<td>0.1</td>
<td>25</td>
<td>7.3%</td>
<td>8.4%</td>
<td>1.8%</td>
<td>60.2</td>
<td>34.7%</td>
<td>35.1%</td>
<td>30.5%</td>
</tr>
<tr>
<td>0.25</td>
<td>25</td>
<td>7.6%</td>
<td>8.8%</td>
<td>1.9%</td>
<td>125.3</td>
<td>39.8%</td>
<td>40.6%</td>
<td>31.7%</td>
</tr>
<tr>
<td>0.5</td>
<td>25</td>
<td>7.8%</td>
<td>9.1%</td>
<td>1.9%</td>
<td>243.5</td>
<td>41.0%</td>
<td>41.9%</td>
<td>32.1%</td>
</tr>
<tr>
<td>0.75</td>
<td>25</td>
<td>7.9%</td>
<td>9.3%</td>
<td>1.9%</td>
<td>360.0</td>
<td>41.6%</td>
<td>42.6%</td>
<td>32.1%</td>
</tr>
<tr>
<td>1</td>
<td>23</td>
<td>8.4%</td>
<td>9.9%</td>
<td>2.1%</td>
<td>175.6</td>
<td>33.1%</td>
<td>33.4%</td>
<td>30.6%</td>
</tr>
</tbody>
</table>

- **Ratio** — time threshold as fraction of solving time
- **#** — number of benchmarks solved
- **$\text{Avg}_{\text{nodes}}, \text{Avg}_{\text{edges}}, \text{Avg}_{\text{core}}$** — average reduction in proof size
- **$T(s)$** — average transformation time in seconds
- **$\text{Max}_{\text{nodes}}, \text{Max}_{\text{edges}}, \text{Max}_{\text{core}}$** — max reduction in proof size
Outline

1 Background
2 Motivation and Related Work
3 Contribution
   ✷ Proof Transformation for Interpolation and Reduction
4 Summary and Future Work
Summary

• Proof transformation
  1 Interpolation, SMT, AB-mixed predicates
Proof transformation

1. Interpolation, SMT, AB-mixed predicates
2. Proof transformation framework for AB-mixed predicates removal
Summary

- Proof transformation
  1. Interpolation, SMT, AB-mixed predicates
  2. Proof transformation framework for AB-mixed predicates removal
  3. Easy combination:
     - Standard SMTs
     - State-of-the art interpolant generation procedures
Summary

- Proof transformation
  1. Interpolation, SMT, AB-mixed predicates
  2. Proof transformation framework for AB-mixed predicates removal
  3. Easy combination:
     - Standard SMTs
     - State-of-the art interpolant generation procedures
  - Rule-based proof reduction
Summary

• Proof transformation
  1. Interpolation, SMT, AB-mixed predicates
  2. Proof transformation framework for AB-mixed predicates removal
  3. Easy combination:
     • Standard SMTs
     • State-of-the art interpolant generation procedures
• Rule-based proof reduction
• Pivots redundancies detection and removal
Future Work

- Exploitation of DPLL proof structure
Future Work

- Exploitation of DPLL proof structure
- Evaluation on concrete applications (e.g. interpolation)
Future Work

- Exploitation of DPLL proof structure
- Evaluation on concrete applications (e.g. interpolation)
- Rule-based control of interpolants’ strength
• Proof reduction

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

• Proof manipulation for interpolation

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

http://www.verify.inf.usi.ch/