Easy Generation and Efficient Validation of Proofs for SAT and QBF

Marijn J.H. Heule

THE UNIVERSITY OF TEXAS AT AUSTIN

Dress Code as Satisfiability Problem

Propositional logic:

- Boolean variables : **tie** and **shirt**
- negation : $\neg$ (not)
- disjunction $\vee$ disjunction (or)
- conjunction $\wedge$ conjunction (and)

Three conditions / clauses:

- clearly one should not wear a **tie** without a **shirt** $(\neg \textbf{tie} \vee \textbf{shirt})$
- not wearing a **tie** nor a **shirt** is impolite $(\textbf{tie} \vee \textbf{shirt})$
- wearing a **tie** and a **shirt** is overkill $\neg(\textbf{tie} \wedge \textbf{shirt}) \equiv (\neg \textbf{tie} \vee \neg \textbf{shirt})$

Is $(\neg \textbf{tie} \vee \textbf{shirt}) \wedge (\textbf{tie} \vee \textbf{shirt}) \wedge (\neg \textbf{tie} \vee \neg \textbf{shirt})$ satisfiable?

A Small Satisfiability (SAT) Problem

$$
\begin{aligned}
&(x_5 \vee x_8 \vee \bar{x}_2) \wedge (x_2 \vee \bar{x}_1 \vee \bar{x}_3) \wedge (\bar{x}_8 \vee \bar{x}_3 \vee \bar{x}_7) \wedge (\bar{x}_5 \vee x_3 \vee x_8) \wedge \\
&(\bar{x}_6 \vee \bar{x}_1 \vee \bar{x}_5) \wedge (x_8 \vee \bar{x}_9 \vee x_3) \wedge (x_2 \vee x_1 \vee x_3) \wedge (\bar{x}_1 \vee x_8 \vee x_4) \wedge \\
&(\bar{x}_9 \vee \bar{x}_6 \vee x_8) \wedge (x_8 \vee x_3 \vee \bar{x}_9) \wedge (x_9 \vee \bar{x}_3 \vee x_8) \wedge (x_6 \vee \bar{x}_9 \vee x_5) \wedge \\
&(x_2 \vee \bar{x}_3 \vee \bar{x}_8) \wedge (x_8 \vee \bar{x}_6 \vee \bar{x}_3) \wedge (x_8 \vee \bar{x}_3 \vee \bar{x}_1) \wedge (\bar{x}_8 \vee x_6 \vee \bar{x}_2) \wedge \\
&(x_7 \vee x_9 \vee \bar{x}_2) \wedge (x_8 \vee \bar{x}_9 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_9 \vee x_4) \wedge (x_8 \vee x_1 \vee \bar{x}_2) \wedge \\
&(x_3 \vee \bar{x}_4 \vee \bar{x}_6) \wedge (\bar{x}_1 \vee \bar{x}_7 \vee x_5) \wedge (\bar{x}_7 \vee x_1 \vee x_6) \wedge (\bar{x}_5 \vee x_4 \vee \bar{x}_6) \wedge \\
&(\bar{x}_4 \vee x_9 \vee \bar{x}_8) \wedge (x_2 \vee x_9 \vee x_1) \wedge (x_5 \vee \bar{x}_7 \vee x_1) \wedge (\bar{x}_7 \vee \bar{x}_9 \vee \bar{x}_6) \wedge \\
&(x_2 \vee x_5 \vee x_4) \wedge (x_8 \vee \bar{x}_4 \vee x_5) \wedge (x_5 \vee x_9 \vee x_3) \wedge (\bar{x}_5 \vee \bar{x}_7 \vee x_9) \wedge \\
&(x_2 \vee \bar{x}_8 \vee x_1) \wedge (\bar{x}_7 \vee x_1 \vee x_5) \wedge (x_1 \vee x_4 \vee x_3) \wedge (x_1 \vee \bar{x}_9 \vee \bar{x}_4) \wedge \\
&(x_3 \vee x_5 \vee x_6) \wedge (\bar{x}_6 \vee x_3 \vee \bar{x}_9) \wedge (\bar{x}_7 \vee x_5 \vee x_9) \wedge (x_7 \vee \bar{x}_5 \vee \bar{x}_2) \wedge \\
&(x_4 \vee x_7 \vee x_3) \wedge (x_4 \vee \bar{x}_9 \vee \bar{x}_7) \wedge (x_5 \vee \bar{x}_1 \vee x_7) \wedge (x_5 \vee \bar{x}_1 \vee x_7) \wedge \\
&(x_6 \vee x_7 \vee \bar{x}_3) \wedge (\bar{x}_8 \vee \bar{x}_6 \vee \bar{x}_7) \wedge (x_6 \vee x_2 \vee x_3) \wedge (\bar{x}_8 \vee x_2 \vee x_5)
\end{aligned}
$$

Does there exist an assignment satisfying all clauses?

Search for a satisfying assignment (or proof none exists)

$$
\begin{array}{l}
(x_5 \vee x_8 \vee \bar{x}_2) \wedge (x_2 \vee \bar{x}_1 \vee \bar{x}_3) \wedge (\bar{x}_8 \vee \bar{x}_3 \vee \bar{x}_7) \wedge (\bar{x}_5 \vee x_3 \vee x_8) \wedge \\
(\bar{x}_6 \vee \bar{x}_1 \vee \bar{x}_5) \wedge (x_8 \vee \bar{x}_9 \vee x_3) \wedge (x_2 \vee x_1 \vee x_3) \wedge (\bar{x}_1 \vee x_8 \vee x_4) \wedge \\
(\bar{x}_9 \vee \bar{x}_6 \vee x_8) \wedge (x_8 \vee x_3 \vee \bar{x}_9) \wedge (x_9 \vee \bar{x}_3 \vee x_8) \wedge (x_6 \vee \bar{x}_9 \vee x_5) \wedge \\
(x_2 \vee \bar{x}_3 \vee \bar{x}_8) \wedge (x_8 \vee \bar{x}_6 \vee \bar{x}_3) \wedge (x_8 \vee \bar{x}_3 \vee \bar{x}_1) \wedge (\bar{x}_8 \vee x_6 \vee \bar{x}_2) \wedge \\
(x_7 \vee x_9 \vee \bar{x}_2) \wedge (x_8 \vee \bar{x}_9 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_9 \vee x_4) \wedge (x_8 \vee x_1 \vee \bar{x}_2) \wedge \\
(x_3 \vee \bar{x}_4 \vee \bar{x}_6) \wedge (\bar{x}_1 \vee \bar{x}_7 \vee x_5) \wedge (\bar{x}_7 \vee x_1 \vee x_6) \wedge (\bar{x}_5 \vee x_4 \vee \bar{x}_6) \wedge \\
(\bar{x}_4 \vee x_9 \vee \bar{x}_8) \wedge (x_2 \vee x_9 \vee x_1) \wedge (x_5 \vee \bar{x}_7 \vee x_1) \wedge (\bar{x}_7 \vee \bar{x}_9 \vee \bar{x}_6) \wedge \\
(x_2 \vee x_5 \vee x_4) \wedge (x_8 \vee \bar{x}_4 \vee x_5) \wedge (x_5 \vee x_9 \vee x_3) \wedge (\bar{x}_5 \vee \bar{x}_7 \vee x_9) \wedge \\
(x_2 \vee \bar{x}_8 \vee x_1) \wedge (\bar{x}_7 \vee x_1 \vee x_5) \wedge (x_1 \vee x_4 \vee x_3) \wedge (x_1 \vee \bar{x}_9 \vee \bar{x}_4) \wedge \\
(x_3 \vee x_5 \vee x_6) \wedge (\bar{x}_6 \vee x_3 \vee \bar{x}_9) \wedge (\bar{x}_7 \vee x_5 \vee x_9) \wedge (x_7 \vee \bar{x}_5 \vee \bar{x}_2) \wedge \\
(x_4 \vee x_7 \vee x_3) \wedge (x_4 \vee \bar{x}_9 \vee \bar{x}_7) \wedge (x_5 \vee \bar{x}_1 \vee x_7) \wedge (x_5 \vee \bar{x}_1 \vee x_7) \wedge \\
(x_6 \vee x_7 \vee \bar{x}_3) \wedge (\bar{x}_8 \vee \bar{x}_6 \vee \bar{x}_7) \wedge (x_6 \vee x_2 \vee x_3) \wedge (\bar{x}_8 \vee x_2 \vee x_5)
\end{array}
$$

Play the SAT game:
`http://www.cril.univ-artois.fr/~roussel/satgame/satgame.php`

Motivation

Satisfiability solvers are used in amazing ways...

- Hardware verification: Centaur x86 verification
- Combinatorial problems:
  - van der Waerden numbers
    [Dransfield, Marek, and Truszczynski, 2004; Kouril and Paul, 2008]
  - Gardens of Eden in Conway's Game of Life
    [Hartman, Heule, Kwekkeboom, and Noels, 2013]
  - Erdős Discrepancy Problem [Konev and Lisitsa, 2014]

Motivation

Satisfiability solvers are used in amazing ways...

- Hardware verification: Centaur x86 verification
- Combinatorial problems:
  - van der Waerden numbers
    [Dransfield, Marek, and Truszczynski, 2004; Kouril and Paul, 2008]
  - Gardens of Eden in Conway's Game of Life
    [Hartman, Heule, Kwekkeboom, and Noels, 2013]
  - Erdős Discrepancy Problem [Konev and Lisitsa, 2014]

..., but satisfiability solvers have errors.

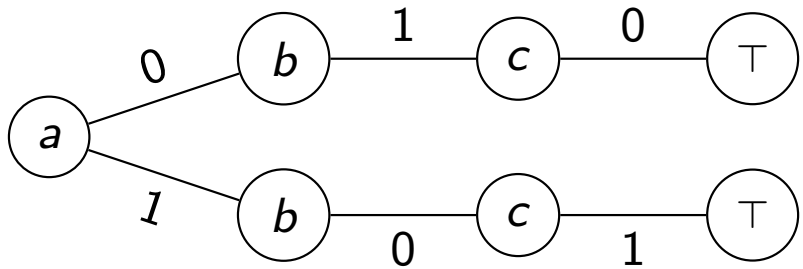
- Documented bugs in SAT, SMT, and QBF solvers
  [Brummayer and Biere, 2009; Brummayer et al., 2010]
- Competition winners have contradictory results
  (HWMCC winners from 2011 and 2012)
- Implementation errors often imply conceptual errors

Introduction to QBF

A quantified Boolean formula (QBF) is a propositional formula where variables are existentially ($\exists$) or universally ($\forall$) quantified.

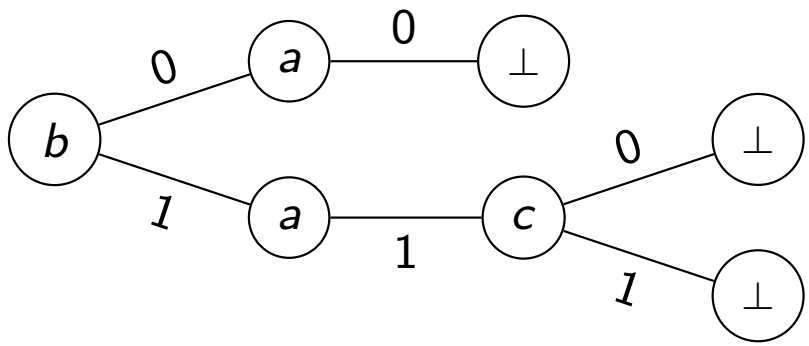
Consider the formula $\forall a \exists b, c.(a \vee b) \wedge (\bar{a} \vee c) \wedge (\bar{b} \vee \bar{c})$

A model is:



Consider the formula $\exists b \forall a \exists c.(a \vee b) \wedge (\bar{a} \vee c) \wedge (\bar{b} \vee \bar{c})$

A counter-model is:



Motivation for our QBF Proof System

Lots of "discrepancies" and unique results in QBF solvers:

- i.e., results that disagree with the majority of solvers.

To gain confidence in QBF results they need to be validated:

- existing methods cannot validate some QBF preprocessing.

QBF preprocessing is crucial for fast performance:

- most state-of-the-art solvers use the preprocessor bloqqer;
- current methods can produce exponentially large proofs or require exponential checking time in worst case;
- some techniques cannot be checked with these methods.

Clausal Proof Systems for SAT and QBF

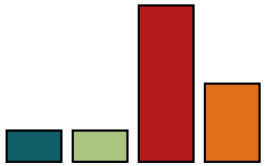
Ideal Properties of a Proof System for SAT Solvers

Easy to Emit

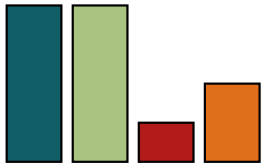
Compact

Checked Efficiently

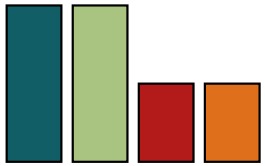
Expressive



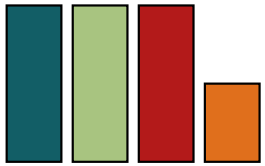
Resolution Proofs
Zhang and Malik, 2003
Van Gelder, 2008; Biere, 2008



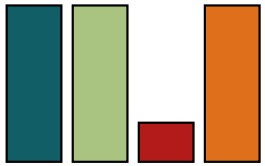
Clausal Proofs
Goldberg and Novikov, 2003
Van Gelder, 2008



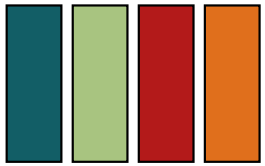
Clausal proofs + clause deletion
Heule, Hunt, Jr., and Wetzler [STVR 2014]



Optimized clausal proof checker
Heule, Hunt, Jr., and Wetzler [FMCAD '13]



Clausal RAT proofs
Heule, Hunt, Jr., and Wetzler [CADE 2013]



RAT proofs + clause deletion
Wetzler, Heule, and Hunt, Jr. [SAT 2014]

Clausal Proof System

Learn: add a clause
* Preserve satisfiability

Unsatisfiable
* Learn empty clause

init $\longrightarrow (\pi.)\psi$

Satisfiable
* Forget last clause

Forget: remove a clause
* Preserve unsatisfiablity

Abstract Proof System for SAT Inprocessing

joint work with Matti Järvisalo and Armin Biere

Inprocessing: Advantages

Interleave burst of preprocessing-style inference steps with conflict-driven clause-learning search

Combine various preprocessing techniques

- Variable elimination, subsumption, self-subsuming resolution, failed literals, equivalent literals, blocked clause elimination, hidden tautology elimination, unhiding, ...

Lingeling ats [Biere, 2013]
SAT Competition 2013 Applications SAT+UNSAT instances

300 instances, 1-h timeout per instance

Configuration	**#solved**	**SAT**	**UNSAT**	flags
default	182	90	92	
no inprocessing	158	89	69	–inprocessing=0
no pre/inprocessing	144	80	64	–plain=1

Abstract Inprocessing

Characterize inprocessing solving as a transition system

State $\varphi\,[\,\rho\,]\,\sigma$

- φ: current "irredundant" clauses
- ρ: current "redundant" clauses
- φ and $\varphi \wedge \rho$ are satisfiability-equivalent, $\varphi \models \rho$ is not required
- σ: sequence of literal-clause pairs $\langle l{:}C \rangle$ for model reconstruction

Legal next states $\varphi'\,[\,\rho'\,]\,\sigma'$ of $\varphi\,[\,\rho\,]\,\sigma$ expressed by rules:

$$\frac{\varphi\,[\,\rho\,]\,\sigma}{\varphi'\,[\,\rho'\,]\,\sigma'}$$

The Rules

Learn $\dfrac{\varphi\,[\,\rho\,]\,\sigma}{\varphi\,[\,\rho \wedge C\,]\,\sigma}\ \sharp$ **Forget** $\dfrac{\varphi\,[\,\rho \wedge C\,]\,\sigma}{\varphi\,[\,\rho\,]\,\sigma}$

Strengthen $\dfrac{\varphi\,[\,\rho \wedge C\,]\,\sigma}{\varphi \wedge C\,[\,\rho\,]\,\sigma}$ **Weaken** $\dfrac{\varphi \wedge C\,[\,\rho\,]\,\sigma}{\varphi\,[\,\rho \wedge C\,]\,\sigma \cup \langle I{:}C \rangle}\ \flat$

Learn new redundant clause C to ρ.

- Generic precondition $\sharp$: $\varphi \wedge \rho$ and $\varphi \wedge \rho \wedge C$ are satisfiability-equivalent.

Forget redundant clause C from ρ.

Strengthen φ by making redundant C irredundant

Weaken φ by making irredundant C redundant

- Generic precondition $\flat$: φ and $\varphi \wedge C$ are satisfiability-equivalent.

- A sound and complete proof system

Intuition why Learn has to take redundancy into account

$$\text{Learn} \qquad \frac{\varphi\,[\,\rho\,]\,\sigma}{\varphi\,[\,\rho \wedge C\,]\,\sigma}\ \sharp$$

- Q: Could the precondition $\sharp$ of Learn

  "$\varphi \wedge \rho$ and $\varphi \wedge \rho \wedge C$ are satisfiability-equivalent"

  be weakened to

  "φ and $\varphi \wedge C$ are satisfiability-equivalent"

  i.e., must the redundant clauses be taken into account for Learn?

- A: ρ is essential: ignoring ρ breaks main invariant φ sat-eq $\varphi \wedge \rho$
  - Consider $F = (a)$.

  1. Initial state $(a)\,[\emptyset]\,\langle\rangle$
  2. Obtain $\emptyset\,[(a)]\,\langle a{:}(a)\rangle$ through Weaken.
  3. In case ρ were ignored in $\sharp$:
     apply Learn and derive $\emptyset\,[(a) \wedge (\bar{a})]\,\langle a{:}(a)\rangle$.

  - Does not preserve satisfiability: $(a) \wedge (\bar{a})$ is unsatisfiable.

Towards Practice: Instantiating the Rules

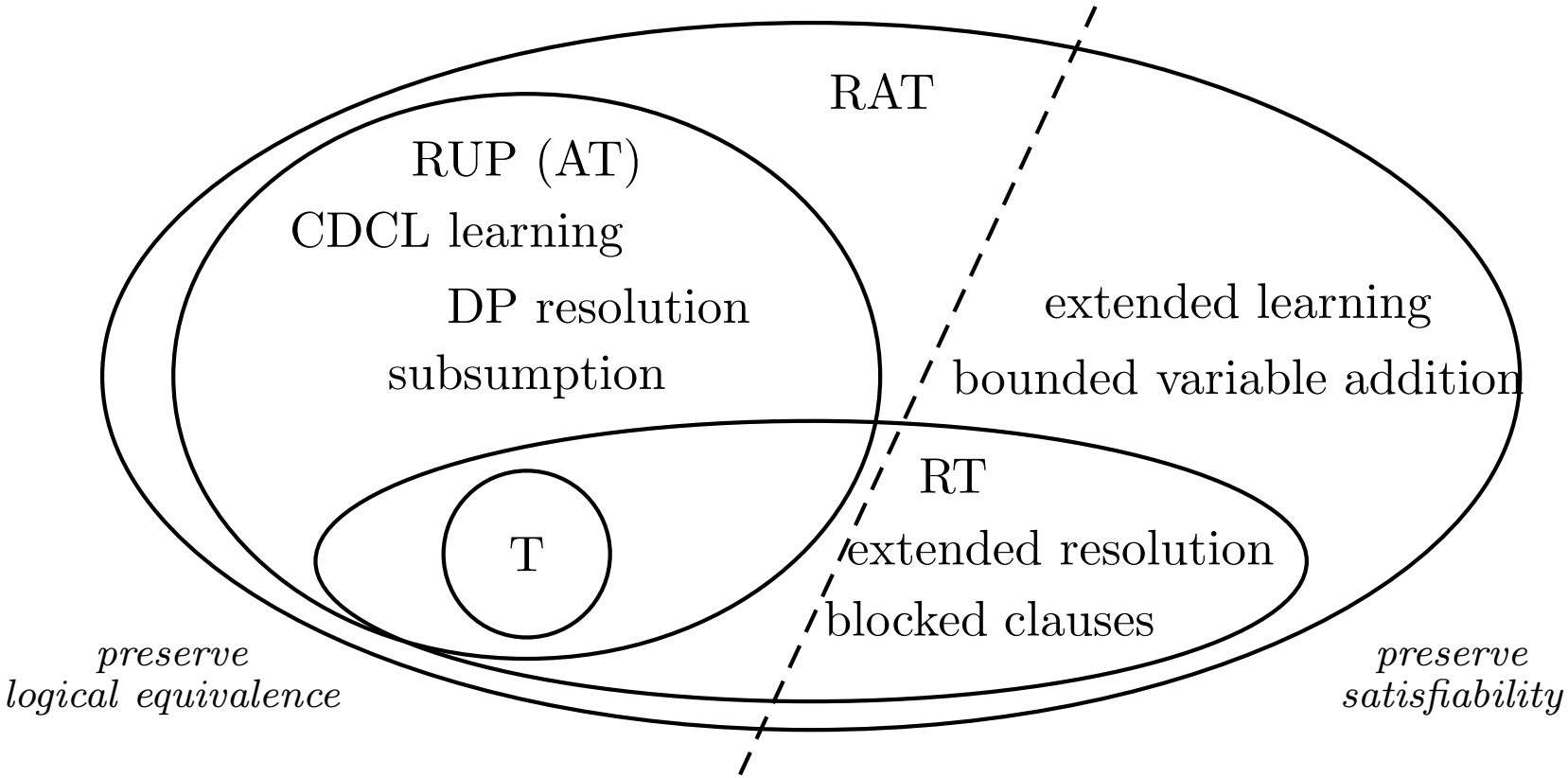
The generic preconditions $\sharp$ and $\flat$ for Learn and Weaken are impractical: checking satisfiablity-equivalence is NP-complete

In practice: procedures are based on polynomial-time computable redundancy properties

Moreover: a single polynomial-time computable clause redundancy property is enough for a generic system!

- RAT: *resolution asymmetric tautologies*

Relationship between Redundancy Properties



All known techniques can be expressed using RAT [IJCAR'12]

RAT: Resolution Asymmetric Tautologies

Clause C has AT (Asymmetric Tautology) w.r.t. $F \setminus C$ iff unit propagation derives a conflict in $(F \setminus C) \wedge \neg C$.

- E.g. $(a \vee b)$ has AT w.r.t. $(a \vee c) \wedge (\bar{c} \vee \bar{d}) \wedge (b \vee d)$
- Tautologies have AT

Clause C *has* RAT (Resolution Asymmetric Tautology) w.r.t. $F \setminus C$ iff

- there exists a literal $l \in C$ such that
  for each clause $C' \in F$ with $\bar{l} \in C'$
  clause $(C' \setminus \bar{l}) \cup C$ has AT w.r.t. $F \setminus C$.
- E.g. (a) has RAT w.r.t. $(a \vee b) \wedge (\bar{a} \vee c) \wedge (\bar{b} \vee c)$
- Clauses with AT w.r.t. F have RAT w.r.t. F

Capturing Inprocessing Solvers using RAT

Learn $\dfrac{\varphi\,[\,\rho\,]\,\sigma}{\varphi\,[\,\rho \wedge C\,]\,\sigma}\ \natural$

Forget $\dfrac{\varphi\,[\,\rho \wedge C\,]\,\sigma}{\varphi\,[\,\rho\,]\,\sigma}$

Strengthen $\dfrac{\varphi\,[\,\rho \wedge C\,]\,\sigma}{\varphi \wedge C\,[\,\rho\,]\,\sigma}$

Weaken $\dfrac{\varphi \wedge C\,[\,\rho\,]\,\sigma}{\varphi\,[\,\rho \wedge C\,]\,\sigma \cup \langle l{:}C \rangle}\ \flat$

Polynomial-time computable preconditions:

- $\natural$: C has RAT w.r.t. $\varphi \wedge \rho$.
- $\flat$: C has RAT (on l) w.r.t. φ.

- Simulates generally used inprocessing techniques
  - Pure literal elimination, clause elimination (including subsumption, blocked clause elimination, ...), clause addition, variable elimination, hyper-binary resolution, self-subsuming resolution, equivalent literal reasoning, hidden literal elimination, clause learning, extended resolution, ...
- Has a unifying linear-time model reconstruction algorithm *covering all these techniques*

Example of incorrect clause elimination

Idea: eliminate C if it is redundant w.r.t. $\varphi \wedge \rho$.

- This would allow using redundant learned clauses in ρ, which can later be forgotten, for weakening φ.

Bad Idea:

- Consider $\rho_0 = \emptyset$ and the minimally unsatisfiable formula $\varphi_0 = (a \vee \bar{b}) \wedge (\bar{a} \vee b) \wedge (\bar{a} \vee \bar{b}) \wedge (a \vee b \vee c) \wedge (a \vee b \vee \bar{c})$
- The clause $(a \vee b)$ has AT w.r.t. φ_0
- Applying Learn gives $\varphi_1 = \varphi_0$ and $\rho_1 = (a \vee b)$.
- $(a \vee b) \in \rho_1$ subsumes $(a \vee b \vee c) \in \varphi_1$
- Weaken would give $\varphi_2 = \varphi_1 \setminus (a \vee b \vee c)$
- However, φ_2 is satisfiable!

Fixed Idea:

The clauses in ρ cannot be used to eliminate clauses in φ

- First move the desired clauses from ρ to φ (Strengthen)

Examples: Simulating Resolution and More

Resolution and Clause Learning

- For any φ, $(C \vee D)$ is an AT w.r.t. $\varphi \wedge (C \vee x) \wedge (D \vee \bar{x})$
- Thus $(C \vee D)$ can be learned by applying Learn.

$\Rightarrow$ Covers resolution-based techniques such as hyper-binary resolution

Extended resolution

- Extension rule: Introduce fresh definitions of the form $x \equiv a \wedge b$ i.e. the CNF formula $(x \vee \bar{a} \vee \bar{b}) \wedge (\bar{x} \vee a) \wedge (\bar{x} \vee b)$
- Simulation:
  1. $(x \vee \bar{a} \vee \bar{b})$ has RAT on x w.r.t. $\varphi \wedge \rho$ (Learn);
  2. $(\bar{x} \vee a)$ and $(\bar{x} \vee b)$ have RAT on $\bar{x}$ w.r.t. $\varphi \wedge (x \vee \bar{a} \vee \bar{b}) \wedge \rho$ (Learn)

Bounded Variable Elimination

- Perhaps the most important SAT preprocessing technique
- Generate all resolvents w.r.t. variable x, then forget all antecedents
- Simulation:
  1. Learn and Strengthen resolvents; 2. Weaken and Forget antecedents

Model Reconstruction

Weaken may introduce new models

$$\text{Weaken} \quad \frac{\varphi \wedge C\,[\,\rho\,]\,\sigma}{\varphi\,[\,\rho \wedge C\,]\,\sigma \cup \langle l{:}C \rangle} \; \flat$$

Given a model τ for the current φ:

```
1   while σ is not empty do
2     remove the last literal-clause pair ⟨l:C⟩ from σ
3     if C is not satisfied by τ then τ := (τ \ {l = 0}) ∪ {l = 1}
4   return τ
```

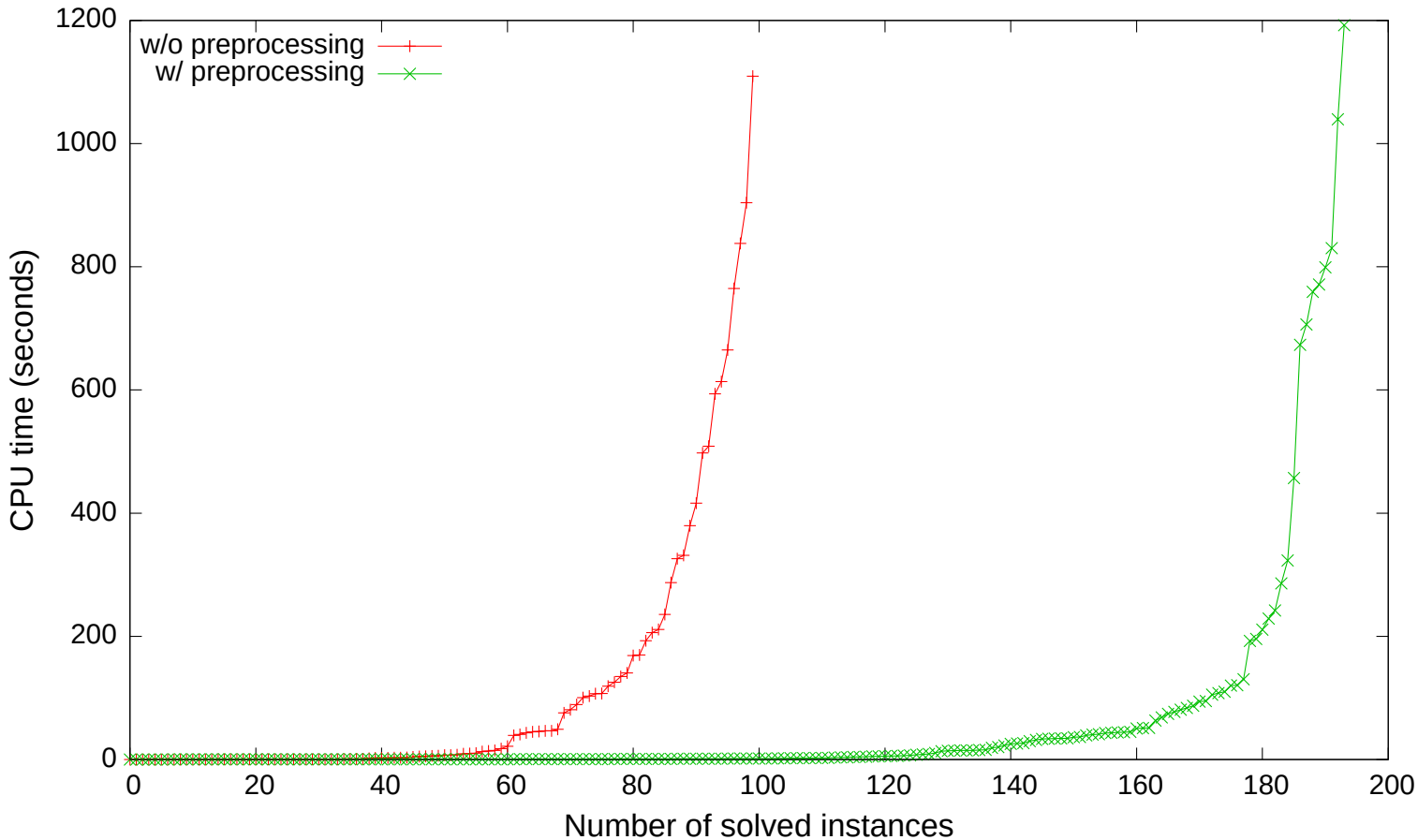
Clausal Proofs for QBF Preprocessing

joint work with Martina Seidl and Armin Biere

QBF Preprocessing

Preprocessing is crucial to solve most QBF instances efficiently.

Results of DepQBF w/ and w/o bloqqer on QBF Eval 2012



QBF Preprocessing

Preprocessing is crucial to solve most QBF instances efficiently.

There exists lots of techniques. The most important ones are:

- tautology elimination, subsumption, universal reduction, existential pure literal elimination, strengthening, blocked clause elimination, unit literal elimination, universal pure literal elimination, covered literal addition, variable elimination, and universal expansion.

Existing methods and proof formats have shortcomings:

- some techniques require exponentially-sized proofs; and
- for some other techniques, it is not even known whether one can construct such a proof.

Challenges for Quantified Boolean Formulas (QBF)

Preprocessing is crucial to solve most QBF instances efficiently.

Proofs are useful for applications and to validate solver output.

Main challenges regarding QBF and preprocessing [Janota'13]:

1. produce proofs that can be validated in polynomial time;
2. develop methods to validate all QBF preprocessing; and
3. narrow the performance gap between solving with and without proof generation.

In our IJCAR'14 paper [1], we meet all three challenges!

[1] Marijn J. H. Heule, Matina Seidl and Armin Biere:
A Unified Proof System for QBF Preprocessing.
IJCAR 2014, LNCS 8562, pp 91-106 (2014)

QRAT: Quantified Resolution Asymmetric Tautologies

Clause C has AT (Asymmetric Tautology) w.r.t. $\psi \setminus \{C\}$ iff unit propagation derives a conflict in $(\psi \setminus \{C\}) \wedge \neg C$.

- E.g. $(a \vee b)$ has AT w.r.t. $(a \vee c) \wedge (\bar{c} \vee \bar{d}) \wedge (b \vee d)$
- Tautologies have AT

Clause C *has* QRAT (Quantified Resolution Asymmetric Tautology) w.r.t. $\psi \setminus \{C\}$ under π iff

- there exists a literal $l \in C$ such that
  for each clause $D \in \psi$ with $\bar{l} \in D$
  clause $\{k \mid k \in D, k <_\pi \bar{l}\} \cup C$ has AT w.r.t. $\psi \setminus C$.
- E.g. (a) has QRAT w.r.t.
  $\forall b, c \exists a.(a \vee b) \wedge (\bar{a} \vee c) \wedge (\bar{b} \vee c)$
- Clauses with AT w.r.t. ψ have QRAT w.r.t. ψ

Rules of the QRAT Proof System

	Rule	Preconditions	Postconditions
(N1)	$\dfrac{\pi.\psi}{\pi.\psi \setminus \{C\}}$	C is an asymmetric tautology	
(N2)	$\dfrac{\pi.\psi}{\pi'.\psi \cup \{C\}}$	C is an asymmetric tautology	$\pi' = \pi \exists X$ with $X = \{x \mid x \in \mathsf{vars}(C), x \notin \mathsf{vars}(\pi)\}$
(E1)	$\dfrac{\pi.\psi}{\pi.\psi \setminus \{C\}}$	$C \in \psi$, $\mathsf{Q}(\pi, l) = \exists$ C has QRAT on l w.r.t. ψ	
(E2)	$\dfrac{\pi.\psi}{\pi'.\psi \cup \{C\}}$	$C \notin \psi$, $\mathsf{Q}(\pi, l) = \exists$ C has QRAT on l w.r.t. ψ	$\pi' = \pi \exists X$ with $X = \{x \mid x \in \mathsf{vars}(C), x \notin \mathsf{vars}(\pi)\}$
(U1)	$\dfrac{\pi.\psi \cup \{C\}}{\pi.\psi \cup \{C \setminus \{l\}\}}$	$l \in C$, $\mathsf{Q}(\pi, l) = \forall$, $\bar{l} \notin C$, C has QRAT on l w.r.t. ψ	
(U2)	$\dfrac{\pi.\psi \cup \{C\}}{\pi.\psi \cup \{C \setminus \{l\}\}}$	$l \in C$, $\mathsf{Q}(\pi, l) = \forall$, $\bar{l} \notin C$, C has EUR on l w.r.t. ψ	

Informal QRAT Example

Consider the false QBF formula $\pi.\psi$:

$$\forall a \exists b \forall c \exists d.(a \vee c \vee d) \wedge (\bar{a} \vee b \vee \bar{d}) \wedge (\bar{b} \vee \bar{d}) \wedge (a \vee \bar{b} \vee c) \wedge (b \vee \bar{c})$$

Clause C has QRAT on l w.r.t. $\pi.\psi$ if:

- assign all literals in C to false;
- apply unit propagation;
- check whether all D with $\bar{l} \in D$ are satisfied on a literal $k <_\pi l$.

$\forall a$	$\exists b$	$\forall c$	$\exists d$
a		c	d
$\bar{a}$	b		$\bar{d}$
	$\bar{b}$		$\bar{d}$
a	$\bar{b}$	c	
	b	$\bar{c}$	

Informal QRAT Example

Consider the false QBF formula $\pi.\psi$:

$$\forall a \exists b \forall c \exists d.(a \vee c \vee d) \wedge (\bar{a} \vee b \vee \bar{d}) \wedge (\bar{b} \vee \bar{d}) \wedge (a \vee \bar{b} \vee c) \wedge (b \vee \bar{c})$$

Clause C has QRAT on l w.r.t. $\pi.\psi$ if:

- assign all literals in C to false;
- apply unit propagation;
- check whether all D with $\bar{l} \in D$ are satisfied on a literal $k <_\pi l$.

$\forall a$	$\exists b$	$\forall c$	$\exists d$
a		c	d
$\bar{a}$	b		$\bar{d}$
	$\bar{b}$		$\bar{d}$
a	$\bar{b}$	c	
	b	$\bar{c}$	

Informal QRAT Example

Consider the false QBF formula $\pi.\psi$:

$$\forall a \exists b \forall c \exists d.(a \vee c \vee d) \wedge (\bar{a} \vee b \vee \bar{d}) \wedge (\bar{b} \vee \bar{d}) \wedge (a \vee \bar{b} \vee c) \wedge (b \vee \bar{c})$$

Clause C has QRAT on l w.r.t. $\pi.\psi$ if:

- assign all literals in C to false;
- apply unit propagation;
- check whether all D with $\bar{l} \in D$ are satisfied on a literal $k <_\pi l$.

$\forall a$	$\exists b$	$\forall c$	$\exists d$
a		c	d
$\bar{a}$	b		$\bar{d}$
	$\bar{b}$		$\bar{d}$
a	$\bar{b}$	c	
	b	$\bar{c}$	

Clause $(a \vee c \vee d)$ has QRAT on d and can thus be removed.

Informal QRAT Example

Consider the false QBF formula $\pi.\psi$:

$$\forall a \exists b \forall c \exists d.(a \vee c \vee d) \wedge (\bar{a} \vee b \vee \bar{d}) \wedge (\bar{b} \vee \bar{d}) \wedge (a \vee \bar{b} \vee c) \wedge (b \vee \bar{c})$$

Clause C has QRAT on l w.r.t. $\pi.\psi$ if:

- assign all literals in C to false;
- apply unit propagation;
- check whether all D with $\bar{l} \in D$ are satisfied on a literal $k <_\pi l$.

$\forall a$	$\exists b$	$\forall c$	$\exists d$
$\bar{a}$	b		$\bar{d}$
	$\bar{b}$		$\bar{d}$
a	$\bar{b}$	c	
	b	$\bar{c}$	

Clause $(a \vee c \vee d)$ has QRAT on d and can thus be removed.

Informal QRAT Example

Consider the false QBF formula $\pi.\psi$:

$$\forall a \exists b \forall c \exists d.(a \vee c \vee d) \wedge (\bar{a} \vee b \vee \bar{d}) \wedge (\bar{b} \vee \bar{d}) \wedge (a \vee \bar{b} \vee c) \wedge (b \vee \bar{c})$$

Clause C has QRAT on l w.r.t. $\pi.\psi$ if:

- assign all literals in C to false;
- apply unit propagation;
- check whether all D with $\bar{l} \in D$ are satisfied on a literal $k <_\pi l$.

$\forall a$	$\exists b$	$\forall c$	$\exists d$
$\bar{a}$	b		$\bar{d}$
	$\bar{b}$		$\bar{d}$
a	$\bar{b}$	c	
	b	$\bar{c}$	

Clause $(a \vee c \vee d)$ has QRAT on d and can thus be removed.

Clause $(a \vee \bar{b} \vee c)$ has QRAT on c and can be strengthened.

Main Theoretical Result

We defined one Forget, one Learn, and two Strengthen rules:

- The rules are based on a redundancy property called QRAT
- The property QRAT can be computed in polynomial time

We showed that all QBF preprocessing techniques can be translated into a sequence of these Learn and Forget rules

- Our proof system can be used to validate all techniques
- The validation costs is similar to solving costs

Example

$\forall x_1..x_n \exists y_1..y_n.(x_1 \vee \bar{y}_1) \wedge (\bar{x}_1 \vee y_1)..(x_n \vee \bar{y}_n) \wedge (\bar{x}_n \vee y_n)$

- Our Forget rule can eliminate all clauses (linear time)
- A model for the formula is exponential in n

QBF: Universal Expansion Example

Universal expansion eliminates an innermost universal variable x by duplicating the formula inner to x.

$$\frac{\pi \forall x \exists Y.\psi, C_1 \vee \bar{x}, \ldots, C_i \vee \bar{x}, D_1 \vee x, \ldots, D_j \vee x, E_1, \ldots, E_k}{\pi \exists Y Y'.\psi, C_1, \ldots, C_i, D'_1, \ldots, D'_j, E_1, \ldots, E_k, E'_1, \ldots, E'_k}$$

QBF: Universal Expansion Example

Universal expansion eliminates an innermost universal variable x by duplicating the formula inner to x.

$$\frac{\pi \forall x \exists Y.\psi, C_1 \vee \bar{x}, \ldots, C_i \vee \bar{x}, D_1 \vee x, \ldots, D_j \vee x, E_1, \ldots, E_k}{\pi \exists Y Y'.\psi, C_1, \ldots, C_i, D'_1, \ldots, D'_j, E_1, \ldots, E_k, E'_1, \ldots, E'_k}$$

The true formula $\forall a \exists b, c.(\bar{a} \vee c) \wedge (a \vee b) \wedge (\bar{b} \vee \bar{c})$ can be expanded to:

$$\exists b, c, b', c'.(c) \wedge (b') \wedge (\bar{b} \vee \bar{c}) \wedge (\bar{b'} \vee \bar{c'})$$

QBF: Universal Expansion Example

Universal expansion eliminates an innermost universal variable x by duplicating the formula inner to x.

$$\frac{\pi \forall x \exists Y.\psi, C_1 \vee \bar{x}, \ldots, C_i \vee \bar{x}, D_1 \vee x, \ldots, D_j \vee x, E_1, \ldots, E_k}{\pi \exists Y Y'.\psi, C_1, \ldots, C_i, D'_1, \ldots, D'_j, E_1, \ldots, E_k, E'_1, \ldots, E'_k}$$

The true formula $\forall a \exists b, c.(\bar{a} \vee c) \wedge (a \vee b) \wedge (\bar{b} \vee \bar{c})$ can be expanded to:

$$\exists b, c, b', c'.(c) \wedge (b') \wedge (\bar{b} \vee \bar{c}) \wedge (\bar{b}' \vee \bar{c}')$$

The false formula $\exists b \forall a \exists c.(\bar{a} \vee c) \wedge (a \vee b) \wedge (\bar{b} \vee \bar{c})$ can be expanded to:

$$\exists b, c, c'.(c) \wedge (b) \wedge (\bar{b} \vee \bar{c}) \wedge (\bar{b} \vee \bar{c}')$$

QBF: Universal Expansion Example with QRAT

$$\frac{\pi \forall x \exists Y.\psi,\ C_1 \vee \bar{x},\ \ldots,\ C_i \vee \bar{x},\ D_1 \vee x,\ \ldots,\ D_j \vee x,\ E_1,\ \ldots,\ E_k}{\pi \exists Y Y'.\psi,\ C_1,\ \ldots,\ C_i,\ D'_1,\ \ldots,\ D'_j,\ E_1,\ \ldots,\ E_k,\ E'_1,\ \ldots,\ E'_k}$$

$$\frac{\forall a \exists b, c.(\bar{a} \vee c) \wedge (a \vee b) \wedge (\bar{b} \vee \bar{c})}{\exists b, c, b', c'.(c) \wedge (b') \wedge (\bar{b} \vee \bar{c}) \wedge (\bar{b'} \vee \bar{c'})}$$

QBF: Universal Expansion Example with QRAT

$$\frac{\pi \forall x \exists Y.\psi, C_1 \vee \bar{x}, \ldots, C_i \vee \bar{x}, D_1 \vee x, \ldots, D_j \vee x, E_1, \ldots, E_k}{\pi \exists Y Y'.\psi, C_1, \ldots, C_i, D'_1, \ldots, D'_j, E_1, \ldots, E_k, E'_1, \ldots, E'_k}$$

$$\frac{\forall a \exists b, c.(\bar{a} \vee c) \wedge (a \vee b) \wedge (\bar{b} \vee \bar{c})}{\exists b, c, b', c'.(c) \wedge (b') \wedge (\bar{b} \vee \bar{c}) \wedge (\bar{b}' \vee \bar{c}')}$$

Phase 1: Learn

1. $(a \vee b \vee \bar{b}')$
2. $(a \vee \bar{b} \vee b')$
3. $(a \vee c \vee \bar{c}')$
4. $(a \vee \bar{c} \vee c')$
5. $(\bar{a} \vee \bar{b} \vee \bar{c})$
6. $(a \vee b')$
7. $(a \vee \bar{b}' \vee \bar{c}')$

Phase 2: Forget

1. $(a \vee b)$
2. $(\bar{b} \vee \bar{c})$
3. $(a \vee b \vee \bar{b}')$
4. $(a \vee \bar{b} \vee b')$
5. $(a \vee c \vee \bar{c}')$
6. $(a \vee \bar{c} \vee c')$

Phase 3: Strengthen

1. $(\bar{a} \vee c)$
2. $(a \vee b')$
3. $(\bar{a} \vee \bar{b} \vee \bar{c})$
4. $(a \vee \bar{b}' \vee \bar{c}')$

Future Directions and Conclusions

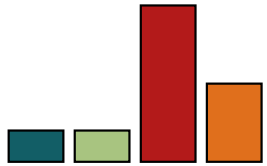
All Work Done Regarding SAT Proof Checking? NO

Easy to Emit

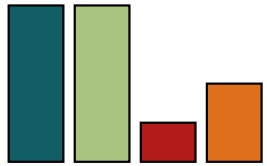
Compact

Checked Efficiently

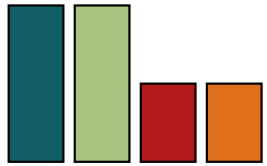
Expressive



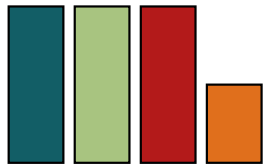
Resolution Proofs
Zhang and Malik, 2003
Van Gelder, 2008; Biere, 2008



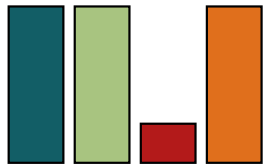
Clausal Proofs
Goldberg and Novikov, 2003
Van Gelder, 2008



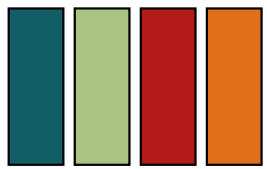
Clausal proofs + clause deletion
Heule, Hunt, Jr., and Wetzler [STVR 2014]



Optimized clausal proof checker
Heule, Hunt, Jr., and Wetzler [FMCAD '13]



Clausal RAT proofs
Heule, Hunt, Jr., and Wetzler [CADE 2013]



RAT proofs + clause deletion
Wetzler, Heule, and Hunt, Jr. [SAT 2014]

All Work Done Regarding SAT Proof Checking? NO

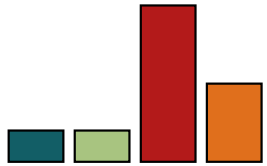
Easy to Emit

Compact

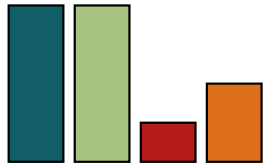
Checked Efficiently

Expressive

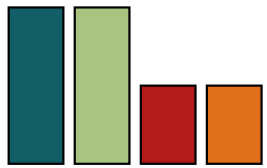
Verified



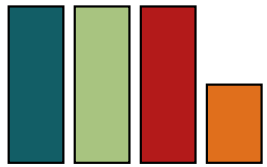
Resolution Proofs
Zhang and Malik, 2003
Van Gelder, 2008; Biere, 2008



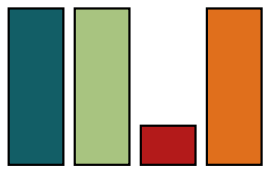
Clausal Proofs
Goldberg and Novikov, 2003
Van Gelder, 2008



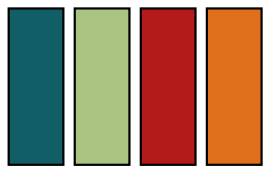
Clausal proofs + clause deletion
Heule, Hunt, Jr., and Wetzler [STVR 2014]



Optimized clausal proof checker
Heule, Hunt, Jr., and Wetzler [FMCAD '13]



Clausal RAT proofs
Heule, Hunt, Jr., and Wetzler [CADE 2013]



RAT proofs + clause deletion
Wetzler, Heule, and Hunt, Jr. [SAT 2014]

Future Directions

Novel techniques arise from the proof systems

- SAT: Elimination and addition of RAT clauses
- SAT: Partial variable elimination
- QBF: Elimination of universal RAT literals
- Many other options

Efficient expression of all techniques

- Main focus: all QBF solving techniques (i.e., not only preprocessing)
- Gaussian Elimination
- Symmetry breaking
- Cardinality / pseudo-Boolean reasoning

Conclusions

Our Abstract Proof System for SAT Inprocessing

- Captures generally used inprocessing and CDCL techniques
- Check individual techniques for correctness via the inprocessing rules
- Yields a generic and simple model reconstruction algorithm
- A basis for developing novel inprocessing techniques

Conclusions

Our Abstract Proof System for SAT Inprocessing

- Captures generally used inprocessing and CDCL techniques
- Check individual techniques for correctness via the inprocessing rules
- Yields a generic and simple model reconstruction algorithm
- A basis for developing novel inprocessing techniques

Our Proof System for QBF Preprocessing

- Polynomially-verifiable certificates for true and false QBFs;
- Overhead of emitting QRAT proofs is very low; and
- All preprocessing techniques used in state-of-the-art QBF tools are covered by QRAT, including universal expansion.
- A basis for developing novel QBF preprocessing techniques

Conclusions

Our Abstract Proof System for SAT Inprocessing

- Captures generally used inprocessing and CDCL techniques
- Check individual techniques for correctness via the inprocessing rules
- Yields a generic and simple model reconstruction algorithm
- A basis for developing novel inprocessing techniques

Our Proof System for QBF Preprocessing

- Polynomially-verifiable certificates for true and false QBFs;
- Overhead of emitting QRAT proofs is very low; and
- All preprocessing techniques used in state-of-the-art QBF tools are covered by QRAT, including universal expansion.
- A basis for developing novel QBF preprocessing techniques

Thanks!