Interpolants from SAT solving certificates

Adrián Rebola-Pardo
TU Wien

Automata, Logic and Games
Singapore
August 29th, 2016
Propositional interpolants

Let $A, B$ be propositional formulae such that $A \land B$ is unsatisfiable.
Propositional interpolants

Let \( A, B \) be propositional formulae such that \( A \land B \) is unsatisfiable.

**Interpolants** an \((A, B)\)-interpolant is a propositional formula \( P \) such that:

- \( A \models P \).
- \( P \land B \) is unsatisfiable.
- \( P \) contains only variables occurring in both \( A \) and \( B \).
Propositional interpolants

Let $A, B$ be propositional formulae such that $A \land B$ is unsatisfiable.

**Interpolants** an $(A, B)$-interpolant is a propositional formula $P$ such that:

- $A \models P$.
- $P \land B$ is unsatisfiable.
- $P$ contains only variables occurring in both $A$ and $B$.

Interpolants are essential tools in formal methods and software verification:

- (Un)bounded model checking [McMillan ’03]
- Boolean synthesis [Jiang et al. ’09]
- Fault localization [Ermis et al. ’12]
- Hardware verification [Keng Veneris ’09]
Interpolation in practice
Interpolant generation

The good old times...

unsatisfiable CNF instance → SAT solver → resolution proof → interpolation system → interpolant

[Zhang, Malik ’03] [Huang ’95]
Interpolant generation

The good old times are gone

unsatisfiable CNF instance → SAT solver → resolution proof → interpolation system → interpolant

[Heule, Hunt, Wetzler'14]

Properties of DRAT proofs

✔ Shorter and easier to generate or check than resolution proofs.

✔ Allow to express satisfiability-preserving techniques.

✘ We do not know how to generate interpolants from DRAT proofs.

[Zhang, Malik'03]
Interpolant generation

The good old times are gone

unsatisfiable CNF instance → SAT solver → DRAT proof → interpolation system → interpolant

Properties of DRAT proofs

✔ Shorter and easier to generate or check than resolution proofs.
✔ Allow to express satisfiability-preserving techniques.

[Heule, Hunt, Wetzler ’14]

[Huang ’95]
Interpolant generation

The good old times are gone

![Diagram showing the process of generating interpolants from unsatisfiable CNF instances using SAT solvers and DRAT proofs.]

[Heule, Hunt, Wetzler ’14]

Properties of DRAT proofs

- ✔ Shorter and easier to generate or check than resolution proofs.
- ✔ Allow to express satisfiability-preserving techniques.
- ❌ We do not know how to generate interpolants from DRAT proofs.
Interpolant generation from proofs

Three approaches

unsatisfiable CNF instance

CDCL SAT solver & friends

Purely CDCL SAT solver

Communicating SAT solvers

DRAT proof

resolution/DRUP proof

interpolant

???

interpolation system

interpolant
Interpolant generation from proofs

Three approaches

- Gaussian elimination
- Symmetry breaking
- Cardinality resolution
- Blocked clause addition
- Blocked clause elimination

unsatisfiable CNF instance

CDCL SAT solver & friends

Purely CDCL SAT solver

Communicating SAT solvers

DRAT proof

resolution/DRUP proof

interpolant system

interpolant

???
Interpolant generation from proofs

Three approaches

- Gaussian elimination
- Symmetry breaking
- Cardinality resolution
- Blocked clause addition
- Blocked clause elimination

unsatisfiable CNF instance

CDCL & friends

CDCL SAT solver

Purely CDCL SAT solver

Communicating SAT solvers

NO INTERPOLANT

DRAT proof

resolution/DRUP proof

interpolant system

interpolant

???
Interpolant generation from proofs

Three approaches

- Gaussian elimination
- Symmetry breaking
- Cardinality resolution
- Blocked clause addition
- Blocked clause elimination

unsatisfiable CNF instance

CDCL & friends

CDCL SAT solver

Purely CDCL SAT solver

Communicating SAT solvers

interpolant

interpolation system

interpolant

DRAT proof

resolution/DRUP proof

???

[Huang ’95]
[Pudlák ’97]
[McMillan ’05]
[D’Silva et al. ’10]
[Gurfinkel Vizel ’14]
[Weissenbacher Schlaipfer ’16]
Interpolant generation from proofs

Three approaches

- Gaussian elimination
- Symmetry breaking
- Cardinality resolution
- Blocked clause addition
- Blocked clause elimination

unsatisfiable CNF instance

CDCL & friends

Purely CDCL SAT solver

Communicating SAT solvers

interpolant

interpolant system

DRAT proof

resolution/DRUP proof

exponential gap

[Chockler et al. ’12]
[Bayles et al. ’13]
[Huang ’95]
[Pudlák ’97]
[McMillan ’05]
[D’Silva et al. ’10]
[Gurfinkel Vizel ’14]
[Weissenbacher Schlaipfer ’16]

NO INTERPOLANT
Interpolant generation from proofs

Three approaches

- Gaussian elimination
- Symmetry breaking
- Cardinality resolution
- Blocked clause addition
- Blocked clause elimination

unsatisfiable CNF instance

CDCL & friends

Gaussian elimination

resolution/DRUP proof

interpolant system

interpolant

[Chockler et al.'12]
[Bayless et al.'13]
[Huang '95]
[Pudlák '97]
[McMillan '05]
[D'Silva et al. '10]
[Gurfinkel Vizel '14]
[Weissenbacher Schlaipfer '16]

Communicating SAT solvers

interpolant

TOO INEFFICIENT

MODELENUMERATION ONLY LOCAL IN PROCESSING

NO INTERPOLANT

exponential gap

???
Interpolant generation from proofs

Three approaches

- Gaussian elimination
- Symmetry breaking
- Cardinality resolution
- Blocked clause addition
- Blocked clause elimination

unsatisfiable CNF instance

CDCL & friends

Purely CDCL SAT solver

Communicating SAT solvers

- NO INTERPOLANT
- TOO INEFFICIENT

exponential gap

DRAT proof

resolution/DRUP proof

interpolation system

interpolant

[Huang ’95] [Chockler et al. ’12] [Pudlák ’97] [Bayless et al. ’13] [McMillan ’05] [D’Silva et al. ’10] [Gurfinkel Vizel ’14] [Weissenbacher Schlaipfer ’16]
Interpolant generation from proofs

Three approaches

- Gaussian elimination
- Symmetry breaking
- Cardinality resolution
- Blocked clause addition
- Blocked clause elimination

unsatisfiable CNF instance

CDCL & friends

Gaussian elimination
Symmetry breaking
Cardinality resolution
Blocked clause addition
Blocked clause elimination

NO INTERPOLANT

resolution/DRUP proof

interpolant

interpolation system

interpolant

DRAT proof

interpolant

[Chockler et al. ’12]
[Bayless et al. ’13]
[McMillan ’05]
[D’Silva et al. ’10]
[Gurfinkel Vizel ’14]
[Weissenbacher Schlaipfer ’16]

[Chockler et al. ’12]
[Bayless et al. ’13]
[McMillan ’05]
[D’Silva et al. ’10]
[Gurfinkel Vizel ’14]
[Weissenbacher Schlaipfer ’16]
Interpolant generation from proofs

Three approaches

- Gaussian elimination
- Symmetry breaking
- Cardinality resolution
- Blocked clause addition
- Blocked clause elimination

unsatisfiable CNF instance

CDCL SAT solver & friends

Purely CDCL SAT solver

ONLY LOCAL INPROCESSING

interpolant

interpolant

interpolation system

resolution/DRUP proof

interpolant

[Chockler et al. ’12]
[Bayless et al. ’13]
[Huang ’95]
[Pudlák ’97]
[McMillan ’05]
[D’Silva et al. ’10]
[Gurfinkel Vizel ’14]
[Weissenbacher Schlaipfer ’16]
Interpolant generation from proofs

Three approaches

- Gaussian elimination
- Symmetry breaking
- Cardinality resolution

unsatisfiable CNF instance

CDCL & friends

Pure CDCL solver

Gaussian elimination

Blocked clause addition

Blocked clause elimination

interpolant

exponential gap

interpolant

resolution/DRUP proof

DRAT proof

interpolation system

rectification

[Chockler et al. ’12]
[Bayless et al. ’13]
[McMillan ’05]
[D’Silva et al. ’10]
[Gurfinkel Vizel ’14]
[Weissenbacher Schlaipfer ’16]

[Chockler et al. ’12]
[Bayless et al. ’13]
[McMillan ’05]
[D’Silva et al. ’10]
[Gurfinkel Vizel ’14]
[Weissenbacher Schlaipfer ’16]
Proof systems for SAT solvers
Reverse Unit Propagation (RUP)
Reverse Unit Propagation (RUP)

clauses in $F$

consequence of $F$

clauses in $F$

RUP in $F$
Reverse Unit Propagation (RUP)

DRUP proof system  RUP introduction + arbitrary clause deletion
**DRUP proofs**

**Reverse Unit Propagation (RUP)**

**DRUP proof system**  
RUP introduction + arbitrary clause deletion

- Essentially as powerful as resolution  
  [Beame et al. ’04]
- Interpolants can be easily generated  
  [Gurfinkel Vizel ’14]
A clause $C$ is a **resolution asymmetric tautology (RAT)** in a CNF formula $F$ upon a literal $l$ if every resolvent $C \otimes D$ upon $l$, where $D \in F$, is a RUP in $F$. 

**Theorem** If $C$ is a RAT in $F$, then $F$ is satisfiable if and only if $F \cup \{C\}$ is.
A clause $C$ is a **resolution asymmetric tautology (RAT)** in a CNF formula $F$ upon a literal $l$ if every resolvent $C \otimes D$ upon $l$, where $D \in F$, is a RUP in $F$. 

\[
F = \bigvee_{i=1}^{n} E_i
\]

\[
l \lor C
\]
A clause $C$ is a **resolution asymmetric tautology (RAT)** in a CNF formula $F$ upon a literal $l$ if every resolvent $C \otimes D$ upon $l$, where $D \in F$, is a RUP in $F$. 

**Theorem** If $C$ is a RAT in $F$, then $F$ is satisfiable if and only if $F \cup \{C\}$ is.

**RAT introduction** can be used as an inferencerule of a proof system.
A clause $C$ is a **resolution asymmetric tautology (RAT)** in a CNF formula $F$ upon a literal $l$ if every resolvent $C \otimes D$ upon $l$, where $D \in F$, is a RUP in $F$. 

![Diagram showing resolution asymmetric tautology](image-url)
A clause $C$ is a **resolution asymmetric tautology (RAT)** in a CNF formula $F$ upon a literal $l$ if every resolvent $C \otimes D$ upon $l$, where $D \in F$, is a RUP in $F$. 

Theorem: If $C$ is a RAT in $F$, then $F$ is satisfiable if and only if $F \cup \{C\}$ is. 

RAT introduction can be used as an inferencerule of a proof system.
Resolution asymmetric tautologies

A clause $C$ is a resolution asymmetric tautology (RAT) in a CNF formula $F$ upon a literal $l$ if every resolvent $C \otimes D$ upon $l$, where $D \in F$, is a RUP in $F$. 
A clause $C$ is a resolution asymmetric tautology (RAT) in a CNF formula $F$ upon a literal $l$ if every resolvent $C \otimes D$ upon $l$, where $D \in F$, is a RUP in $F$. 

**Theorem** If $C$ is a RAT in $F$, then $F$ is satisfiable if and only if $F \cup \{C\}$ is.

RAT introduction can be used as an inferencerule of a proof system.
A clause $C$ is a **resolution asymmetric tautology (RAT)** in a CNF formula $F$ upon a literal $l$ if every resolvent $C \otimes D$ upon $l$, where $D \in F$, is a RUP in $F$. 

Theorem: If $C$ is a RAT in $F$, then $F$ is satisfiable if and only if $F \cup \{C\}$ is.

RAT introduction can be used as an inferencerule of a proofsystem.
A clause $C$ is a resolution asymmetric tautology (RAT) in a CNF formula $F$ upon
a literal $l$ if every resolvent $C \otimes D$ upon $l$, where $D \in F$, is a RUP in $F$. 
A clause $C$ is a resolution asymmetric tautology (RAT) in a CNF formula $F$ upon a literal $l$ if every resolvent $C \otimes D$ upon $l$, where $D \in F$, is a RUP in $F$. 

![Diagram showing resolution asymmetric tautology (RAT) in a CNF formula $F$ upon a literal $l$.](image)
A clause $C$ is a **resolution asymmetric tautology (RAT)** in a CNF formula $F$ upon a literal $l$ if every resolvent $C \otimes D$ upon $l$, where $D \in F$, is a RUP in $F$.

**Theorem**  If $C$ is a RAT in $F$, then $F$ is satisfiable if and only if $F \cup \{C\}$ is.

*RAT introduction can be used as an inference rule of a proof system.*
The DRAT proof system

DRAT proof system  RUP introduction + RAT introduction + arbitrary clause deletion

extended resolution

\[ p \leftrightarrow q \land r \]
where \( p \) is fresh

Extended resolution can be simulated by DRAT

\[ \neg p \lor q \]
\[ \neg p \lor r \]
\[ p \lor \neg q \lor \neg r \]

RAT upon \( \neg p \)

\[ \neg p \lor q \]
\[ \neg p \lor r \]
\[ p \lor \neg q \lor \neg r \]

tautology

\[ \Rightarrow RUP \]
\[ \neg q \lor r \lor \neg r \]

tautology

\[ \Rightarrow RUP \]
The DRAT proof system

**DRAT proof system**  RUP introduction + RAT introduction + arbitrary clause deletion

**Extended resolution**  resolution + definitions $p \leftrightarrow q \land r$ where $p$ is fresh

*Extended resolution can be simulated by DRAT*
The DRAT proof system

**DRAT proof system**  RUP introduction + RAT introduction + arbitrary clause deletion

**Extended resolution**  resolution + definitions $p \leftrightarrow q \land r$ where $p$ is fresh

*Extended resolution can be simulated by DRAT*

$p \leftrightarrow q \land r \equiv (\neg p \lor q) \land (\neg p \lor r) \land (p \lor \neg q \lor \neg r)$
The DRAT proof system

DRAT proof system  RUP introduction + RAT introduction + arbitrary clause deletion

Extended resolution  resolution + definitions $p \leftrightarrow q \land r$ where $p$ is fresh

Extended resolution can be simulated by DRAT

\[ \neg p \lor q \]
\[ \neg p \lor r \]
\[ p \lor \neg q \lor \neg r \]

DRAT proof  \((\neg p \lor q)^{\text{RAT}}\), \((\neg p \lor r)^{\text{RAT}}\), \((p \lor \neg q \lor \neg r)^{\text{RAT}}\)
The DRAT proof system

**DRAT proof system**  RUP introduction + RAT introduction + arbitrary clause deletion

**Extended resolution**  resolution + definitions $p \leftrightarrow q \land r$ where $p$ is fresh

*Extended resolution can be simulated by DRAT*

```
\[ \neg p \lor q \lor \neg r \]
```

**DRAT proof**  $(\neg p \lor q)^{\text{RAT}}$, $(\neg p \lor r)^{\text{RAT}}$, $(p \lor \neg q \lor \neg r)^{\text{RAT}}$
The DRAT proof system

**DRAT proof system**  
RUP introduction + RAT introduction + arbitrary clause deletion

**Extended resolution**  
resolution + definitions \( p \leftrightarrow q \land r \) where \( p \) is fresh

*Extended resolution can be simulated by DRAT*

---

**DRAT proof**  
\( (\neg p \lor q)^{RAT}, (\neg p \lor r)^{RAT}, (p \lor \neg q \lor \neg r)^{RAT} \)
The DRAT proof system

**DRAT proof system**  RUP introduction + RAT introduction + arbitrary clause deletion

**Extended resolution**   resolution + definitions $p \leftrightarrow q \land r$ where $p$ is fresh

*Extended resolution can be simulated by DRAT*

\[ \neg p \lor q \]

**DRAT proof**  $(\neg p \lor q)^{\text{RAT}}$, $(\neg p \lor r)^{\text{RAT}}$, $(p \lor \neg q \lor \neg r)^{\text{RAT}}$
The DRAT proof system

**DRAT proof system**  RUP introduction + RAT introduction + arbitrary clause deletion

**Extended resolution**  resolution + definitions $p \leftrightarrow q \land r$ where $p$ is fresh

*Extended resolution can be simulated by DRAT*

\[ \neg p \lor q \]
\[ \neg p \lor r \]
\[ p \lor \neg q \lor \neg r \]

**DRAT proof**  $(\neg p \lor q)^{\text{RAT}}$, $(\neg p \lor r)^{\text{RAT}}$, $(p \lor \neg q \lor \neg r)^{\text{RAT}}$
The DRAT proof system

**DRAT proof system**  RUP introduction + RAT introduction + arbitrary clause deletion

**Extended resolution**  resolution + definitions $p \leftrightarrow q \land r$ where $p$ is fresh

*Extended resolution can be simulated by DRAT*

\[
\neg p \lor q \\
\neg p \lor r \\
p \lor \neg q \lor \neg r
\]

no resolvents

RAT upon $\neg p$

**DRAT proof**  $(\neg p \lor q)^{\text{RAT}}$, $(\neg p \lor r)^{\text{RAT}}$, $(p \lor \neg q \lor \neg r)^{\text{RAT}}$
The DRAT proof system

**DRAT proof system**  RUP introduction + RAT introduction + arbitrary clause deletion

**Extended resolution**  resolution + definitions $p \leftrightarrow q \land r$ where $p$ is fresh

*Extended resolution can be simulated by DRAT*

\[
\begin{align*}
\neg p \lor q \\
\neg p \lor r \\
\neg p \lor \neg q \lor \neg r
\end{align*}
\]

**DRAT proof**  \((\neg p \lor q)^{\text{RAT}}, (\neg p \lor r)^{\text{RAT}}, (p \lor \neg q \lor \neg r)^{\text{RAT}}\)
The DRAT proof system

**DRAT proof system**  RUP introduction + RAT introduction + arbitrary clause deletion

**Extended resolution**  resolution + definitions $p \iff q \land r$ where $p$ is fresh

*Extended resolution can be simulated by DRAT*

**DRAT proof**  $(\neg p \lor q)^{\text{RAT}}, (\neg p \lor r)^{\text{RAT}}, (p \lor \neg q \lor \neg r)^{\text{RAT}}$
**The DRAT proof system**

**DRAT proof system**  RUP introduction + RAT introduction + arbitrary clause deletion

**Extended resolution**  resolution + definitions $p \leftrightarrow q \land r$ where $p$ is fresh

*Extended resolution can be simulated by DRAT*

**DRAT proof**  $(\neg p \lor q)^{\text{RAT}}$, $(\neg p \lor r)^{\text{RAT}}$, $(p \lor \neg q \lor \neg r)^{\text{RAT}}$
The DRAT proof system

DRAT proof system  RUP introduction + RAT introduction + arbitrary clause deletion

Extended resolution  resolution + definitions $p \leftrightarrow q \land r$ where $p$ is fresh

*Extended resolution can be simulated by DRAT*

\[ p \lor \neg q \lor \neg r \]

DRAT proof  $(\neg p \lor q)^{\text{RAT}}$, $(\neg p \lor r)^{\text{RAT}}$, $(p \lor \neg q \lor \neg r)^{\text{RAT}}$
The DRAT proof system

**DRAT proof system**  RUP introduction + RAT introduction + arbitrary clause deletion

**Extended resolution**  resolution + definitions $p \leftrightarrow q \land r$ where $p$ is fresh

*Extended resolution can be simulated by DRAT*

DRAT proof  $(\neg p \lor q)^{\text{RAT}}$, $(\neg p \lor r)^{\text{RAT}}$, $(p \lor \neg q \lor \neg r)^{\text{RAT}}$
The DRAT proof system

**DRAT proof system**  RUP introduction + RAT introduction + arbitrary clause deletion

**Extended resolution**  resolution + definitions $p \leftrightarrow q \land r$ where $p$ is fresh

*Extended resolution can be simulated by DRAT*

\[ p \lor \neg q \lor \neg r \]

**DRAT proof**  $(\neg p \lor q)^{\text{RAT}}$, $(\neg p \lor r)^{\text{RAT}}$, $(p \lor \neg q \lor \neg r)^{\text{RAT}}$
The DRAT proof system

**DRAT proof system**  RUP introduction + RAT introduction + arbitrary clause deletion

**Extended resolution**  resolution + definitions $p \leftrightarrow q \land r$ where $p$ is fresh

*Extended resolution can be simulated by DRAT*

\[
\neg p \lor q \land (\neg p \lor r)
\]

\[
(p \lor \neg q \lor \neg r)
\]

**DRAT proof**  $(\neg p \lor q)^{\text{RAT}}$, $(\neg p \lor r)^{\text{RAT}}$, $(p \lor \neg q \lor \neg r)^{\text{RAT}}$
The DRAT proof system

**DRAT proof system**  RUP introduction + RAT introduction + arbitrary clause deletion

**Extended resolution**  resolution + definitions $p \leftrightarrow q \land r$ where $p$ is fresh

*Extended resolution can be simulated by DRAT*

\[
\neg p \lor q \\
\neg p \lor r \\
p \lor \neg q \lor \neg r
\]

**DRAT proof**  $(\neg p \lor q)^{\text{RAT}}$, $(\neg p \lor r)^{\text{RAT}}$, $(p \lor \neg q \lor \neg r)^{\text{RAT}}$
The DRAT proof system

**DRAT proof system**  RUP introduction + RAT introduction + arbitrary clause deletion

**Extended resolution**  resolution + definitions $p \leftrightarrow q \land r$ where $p$ is fresh

*Extended resolution can be simulated by DRAT*

Properties of extended resolution

- No **lower bound** for length of extended resolution proofs is known.
- Used to express **inprocessing techniques** used in SAT solvers.
- Lacks the **efficient interpolation** property.
- No **interpolation method** is known.
Partial soundness of DRAT

Partial soundness \( F \vdash G \quad \not\Rightarrow \quad F \models G \)
Partial soundness \( F \vdash G \not\Rightarrow F \models G \)

- A CNF formula is unsatisfiable iff there is a **DRAT refutation**.
Partial soundness $ F \vdash G \nRightarrow F \models G$

- A CNF formula is unsatisfiable iff there is a DRAT refutation.
- Intermediate clauses are not necessarily consequences of the premise formula.

If $p$ is fresh, then $F \not\models F \land (p \leftrightarrow q \land r)$
Partial soundness \( F \vdash G \not\Rightarrow F \models G \)

- A CNF formula is unsatisfiable iff there is a DRAT refutation.
- Intermediate clauses are not necessarily consequences of the premise formula.

If \( p \) is fresh, then \( F \not\models F \land (p \leftrightarrow q \land r) \)

- In fact, we can always derive any satisfiable CNF formula!

\[
F = p \quad (p)^{\text{DEL}}, (\neg p)^{\text{RAT}} \quad F' = \neg p
\]
Partial soundness \[ F \vdash G \not\Rightarrow F \models G \]

- A CNF formula is unsatisfiable iff there is a DRAT refutation.
- Intermediate clauses are not necessarily consequences of the premise formula.

If \( p \) is fresh, then \( F \not\models F \land (p \leftrightarrow q \land r) \)

- In fact, we can always derive any satisfiable CNF formula!

\[
F = p \quad (p)^{\text{DEL}}, \quad (\neg p)^{\text{RAT}} \quad F' = \neg p
\]

Interpolation and soundness

- Interpolation algorithms work because an induction invariant holds for partial interpolants.
- This invariant strongly requires soundness of the proof system.
Interpolation from DRAT proofs
Why does RAT work? Eventually, some successor of every RAT becomes a consequence.
RATs, consequences and bindings

Axioms from $F$

RAT in $F$ upon $l$

Not a RAT nor a consequence of $F$

Why does RAT work? Eventually, some successor of every RAT becomes a consequence.
Why does RAT work? Eventually, some successor of every RAT becomes a consequence.
Why does RAT work? Eventually, some successor of every RAT becomes a consequence.
**RATs, consequences and bindings**

**axioms from $F$**

- $l \in \text{RAT in } F \text{ upon } l$
- $l \in \text{not a RAT nor a consequence of } F$
- $l \not\in \text{consequence of } F$

**Why does RAT work?**

Eventually, some successor of every RAT becomes a consequence.

**But when?**

As soon as the pivot literal is eliminated by resolution.
Why does RAT work? Eventually, some successor of every RAT becomes a consequence.

But when? As soon as the pivot literal is eliminated by resolution.
RATs, consequences and bindings

Why does RAT work? Eventually, some successor of every RAT becomes a consequence.

But when? As soon as the pivot literal is eliminated by resolution.

Question Can we obtain a resolution proof of that consequence clause?
**Question** Can we obtain a resolution proof of that consequence clause?

Elimination by resolving the RAT with a clause from $F$
Question  Can we obtain a resolution proof of that consequence clause?

Elimination by resolving the RAT with a clause from $F$
A proof can be extracted when checking the RAT property.
Question  
Can we obtain a resolution proof of that consequence clause?

Elimination by resolving the RAT with a clause from $F$
A proof can be extracted when checking the RAT property.
Question  Can we obtain a resolution proof of that consequence clause?

Elimination by resolving the RAT with a consequence of $F$
**Question**  Can we obtain a resolution proof of that consequence clause?

**Elimination by resolving the RAT with a consequence of** \( F \)

Transform the RAT witnesses along the derivation of the consequence.
Refactoring DRAT proofs into resolution proofs

**Question**  Can we obtain a resolution proof of that consequence clause?

**Elimination by resolving the RAT with a consequence of** $F$
Transform the RAT witnesses along the derivation of the consequence.
Question Can we obtain a resolution proof of that consequence clause?

Elimination by resolving the RAT with a consequence of $F$
Transform the RAT witnesses along the derivation of the consequence.
**Question**  Can we obtain a resolution proof of that consequence clause?

**Elimination by resolving the RAT with a consequence of** $F$

Transform the RAT witnesses along the derivation of the consequence.
Question  Can we obtain a resolution proof of that consequence clause?

Elimination by resolving the RAT with a consequence of $F$
Transform the RAT witnesses along the derivation of the consequence.
**Question** Can we obtain a resolution proof of that consequence clause?

Elimination by resolving the RAT with a consequence of $F$
Transform the RAT witnesses along the derivation of the consequence.
Refactoring DRAT proofs into resolution proofs

Question Can we obtain a resolution proof of that consequence clause?

Elimination by resolving a consequence of the RAT with a consequence of $F$
Transform the RAT witnesses along the bound subproof.
**Question**  Can we obtain a resolution proof of that consequence clause?

**Elimination by resolving a consequence of the RAT with a consequence of** \( F \)

Transform the RAT witnesses along the bound subproof.
Question  Can we obtain a resolution proof of that consequence clause?

Elimination by resolving a consequence of the RAT with a consequence of $F$

Transform the RAT witnesses along the bound subproof.
Question Can we obtain a resolution proof of that consequence clause?

Elimination by resolving a consequence of the RAT with a consequence of $F$.
Transform the RAT witnesses along the bound subproof.
Question  Can we obtain a resolution proof of that consequence clause?

Elimination by resolving a consequence of the RAT with a consequence of $F$
Transform the RAT witnesses along the bound subproof.
Question Can we obtain a resolution proof of that consequence clause?

Elimination by resolving a consequence of the RAT with a consequence of $F$
Transform the RAT witnesses along the bound subproof.
Question Can we obtain a resolution proof of that consequence clause?

Elimination by resolving a consequence of the RAT with a consequence of $F$
Transform the RAT witnesses along the bound subproof.
Question: Can we obtain a resolution proof of that consequence clause?

Elimination by resolving a consequence of the RAT with a consequence of $F$.
Transform the RAT witnesses along the bound subproof.
Interpolant generation from DRAT proofs

Interpolation by rectification into a resolution proof

Axioms from $F$

RAT in $F$ upon $l$

Consequence of $F$

Issues

The interpolant may be exponential with respect to the DRAT proof. But DRAT proofs can be exponentially shorter than DRUP proofs!

Currently we only eliminate RATs and bound paths one by one. For a general enough case, the number of required sweeps is reduced.

Fully rectified DRAT proofs are huge and cannot be held in memory. We try to store only necessary information and exploit RUPs to compress it.
Interpolant generation from DRAT proofs

Interpolation by rectification into a resolution proof

Issues

The interpolant may be exponential with respect to the DRAT proof. But DRAT proofs can be exponentially shorter than DRUP proofs!

Currently we only eliminate RATs and bound paths one by one.

For a general enough case, the number of required sweeps is reduced.

Fully rectified DRAT proofs are huge and cannot be held in memory.

We try to store only necessary information and exploit RUPs to compress it.
Interpolant generation from DRAT proofs

Interpolation by rectification into a resolution proof

axioms from $F$
Interpolant generation from DRAT proofs

Interpolation by rectification into a resolution proof

Axioms from $F$

![Diagram showing the process of interpolant generation from DRAT proofs.](image-url)
Interpolant generation from DRAT proofs

Interpolation by rectification into a resolution proof

Axioms from $F$
Interpolant generation from DRAT proofs

Interpolation by rectification into a resolution proof

Axioms from $F$

Issues

The interpolant may be exponential with respect to the DRAT proof. But DRAT proofs can be exponentially shorter than DRUP proofs! Currently we only eliminate RATS and bound paths one by one. For a general enough case, the number of required sweeps is reduced. Fully rectified DRAT proofs are huge and cannot be held in memory. We try to store only necessary information and exploit RUPs to compress it.
Interpolant generation from DRAT proofs

Interpolation by rectification into a resolution proof

axioms from $F$
Interpolant generation from DRAT proofs

Interpolation by rectification into a resolution proof

Axioms from $F$

Interpolant
Interpolant generation from DRAT proofs

Interpolation by rectification into a resolution proof

Axioms from $F$

Issues

- The interpolant may be exponential with respect to the DRAT proof. 
  *But DRAT proofs can be exponentially shorter than DRUP proofs!*

- Currently we only eliminate RATs and bound paths one by one. 
  *For a general enough case, the number of required sweeps is reduced.*

- Fully rectified DRAT proofs are huge and cannot be held in memory. 
  *We try to store only necessary information and exploit RUPs to compress it.*
Conclusion
State-of-the-art SAT solvers do not (and most likely, will not) produce resolution proofs, because of inprocessing techniques.
Conclusions

- State-of-the-art SAT solvers do not (and most likely, will not) produce resolution proofs, because of inprocessing techniques.

- The de facto standard DRAT certificates can be rectified into resolution proofs, and then interpolants can be extracted.
State-of-the-art SAT solvers do not (and most likely, will not) produce resolution proofs, because of inprocessing techniques.

The *de facto* standard DRAT certificates can be rectified into resolution proofs, and then interpolants can be extracted.

Our efforts now are directed towards an efficient implementation of the algorithm by storing minimal information and using restrictive but general enough versions of DRAT.
Backup slides
A closer look into rectification

\[ C \quad \text{RAT upon} \quad F \quad \text{in} \quad l \]

\[ D, E \quad \text{clauses mutually resolvable upon} \quad k. \]
A closer look into rectification

$C$  RAT upon $F$ in $l$

$D, E$  clauses mutually resolvable upon $k$.

$D$ is resolvable with $C$ upon $l$
A closer look into rectification

\[ \begin{align*}
  C & \quad \text{RAT upon } F \text{ in } l \\
  D, E & \quad \text{clauses mutually resolvable upon } k.
\end{align*} \]

\[ D \otimes E \]

\[ E \] is not resolvable with \( C \) upon \( l \)
A closer look into rectification

\[
\begin{align*}
C & \quad \text{RAT upon } F \text{ in } l \\
D, E & \quad \text{clauses mutually resolvable upon } k.
\end{align*}
\]

\[
D \otimes E \text{ is resolvable with } C \text{ upon } l
\]
A closer look into rectification

\[ C \quad \text{RAT upon } F \text{ in } l \]
\[ D, E \quad \text{clauses mutually resolvable upon } k. \]
A closer look into rectification

$C$  RAT upon $F$ in $l$

$D, E$ clauses mutually resolvable upon $k$.

if $C \lor E$ is not a tautology
A closer look into rectification

$C$  RAT upon $F$ in $l$

$D, E$ clauses mutually resolvable upon $k$.

$D \otimes E$ is resolvable with $C$ upon $l$

$E$ is not resolvable with $C$ upon $l$

$D \otimes E$ is resolvable with $C$ upon $l$

$P$ if $C \lor E$ is not a tautology

otherwise
Interpolants from resolution proofs

Example

\[ A = (\overline{a} \, b \, c) \land (\overline{b} \, d) \land (a \, b) \quad B = (\overline{c} \, e \, f) \land (\overline{e} \, f) \land (\overline{d} \, f) \land (\overline{f}) \]
Interpolants from resolution proofs

Example
\[ A = (\overline{a \ b \ c}) \land (\overline{b \ d}) \land (a \ b) \quad B = (\overline{c \ e \ f}) \land (\overline{e \ f}) \land (\overline{d \ f}) \land (\overline{f}) \]

Interpolation rules [Huang '95]
Interpolants from resolution proofs

Example

\[ A = (\overline{a} \land b \land c) \land (\overline{b} \land d) \land (a \land b) \quad B = (\overline{c} \land e \land f) \land (\overline{e} \land f) \land (d \land f) \land (f) \]

Interpolation rules [Huang '95]
Interpolants from resolution proofs

Example

\[ A = (\overline{a \ b \ c} \land \overline{b \ d} \land (a \ b)) \quad B = (\overline{c \ e \ f} \land (\overline{e} \ f) \land (d \ f) \land \overline{f}) \]

Interpolation rules [Huang ’95]
Interpolants from resolution proofs

Example

\[ A = (\overline{a} \land b \land c) \land (\overline{b} \land d) \land (a \land b) \quad B = (\overline{c} \land e \land f) \land (\overline{e} \land f) \land (\overline{d} \land f) \land (\overline{f}) \]

Interpolation rules [Huang ’95]
Interpolants from resolution proofs

Example

\[ A = (\overline{a} \, b \, c) \land (\overline{b} \, d) \land (a \, b) \quad B = (\overline{c} \, e \, f) \land (\overline{e} \, f) \land (\overline{d} \, f) \land (\overline{f}) \]

Interpolation rules [Huang ’95]
Interpolants from resolution proofs

Example

\[ A = (\overline{a \ b \ c}) \land (\overline{b \ d}) \land (a \ b) \quad B = (\overline{c \ e \ f}) \land (\overline{e \ f}) \land (\overline{d \ f}) \land (\overline{f}) \]

Interpolation rules [Huang ’95]
Interpolants from resolution proofs

Example

\[ A = (\overline{a} \, b \, c) \land (\overline{b} \, d) \land (a \, b) \quad B = (\overline{c} \, e \, f) \land (\overline{e} \, f) \land (d \, f) \land (\overline{f}) \]

Interpolation rules [Huang '95]
Interpolants from resolution proofs

Example

\[ A = (\overline{a} \; \overline{b} \; c) \land (\overline{b} \; d) \land (a \; b) \]

\[ B = (\overline{c} \; e \; f) \land (\overline{e} \; f) \land (\overline{d} \; f) \land (f) \]

Interpolation rules [Huang ’95]
Interpolants from resolution proofs

Example

\[ A = (\overline{a} \land b \land c) \land (\overline{b} \land d) \land (a \land b) \]

\[ B = (\overline{c} \land e \land f) \land (\overline{e} \land f) \land (d \land f) \land (f) \]

Interpolation rules [Huang ’95]
Proofless interpolation

Interpolation through communicating SAT solvers [Chockler Ivrii Matsliah ’12]

\[ A \land P \land Q \]
\[ B \land P \land Q \]

\[ m \models A \land P \land Q \]
\[ m \models B \land P \land Q \]

\[ \neg P \land Q \text{ UNSAT} \]
\[ \neg Q \land P \text{ UNSAT} \]

Disadvantages

Simplification techniques can only be applied locally.

Obtained interpolants are in DNF or CNF.

Clause minimization is required to obtain reasonably-sized interpolants.
Proofless interpolation

Interpolation through communicating SAT solvers [Chockler Ivrii Matsliah '12]

\[ P = \top \]
\[ Q = \top \]

\[ A \quad \text{and} \quad B \]

\[ P \quad \text{and} \quad Q \]

Disadvantages:
- Simplification techniques can only be applied locally.
- Obtained interpolants are in DNF or CNF.
- Clause minimization is required to obtain reasonably-sized interpolants.
Interpolation through communicating SAT solvers [Chockler Ivrii Matsliah ’12]

\[ P = \top \quad Q = \top \]

\[ A \land P \land Q \quad B \land P \land Q \]

Disadvantages

Simplification techniques can only be applied locally.

Obtained interpolants are in DNF or CNF.

Clause minimization is required to obtain reasonably-sized interpolants.
Proofless interpolation

Interpolation through communicating SAT solvers [Chockler Ivrii Matsliah ’12]

\[ P = \top \quad Q = \top \]

\[ A \land P \land Q \]

incremental SAT solver

\[ \neg P \]

\[ P \]

\[ m \models A \land P \land Q \]

\[ m \models B \land P \land Q \]

\[ P = P \land \neg m \mid A \cap B \]

\[ Q = Q \land \neg m \mid A \cap B \]

\[ \neg P \neg Q \]

Disadvantages

- Simplification techniques can only be applied locally.
- Obtained interpolants are in DNF or CNF.
- Clause minimization is required to obtain reasonably-sized interpolants.
Proofless interpolation

Interpolation through communicating SAT solvers [Chockler Ivrii Matsliah ’12]

\[ P = \top \]

\[ Q = \top \]

\[ A \land P \land Q \]

\[ B \land P \land Q \]

\[ m \models A \land P \land Q \]

\[ m \models B \land P \land Q \]

SAT

incremental SAT solver

UNSAT

\[ \neg P \]

\[ Q \]
Proofless interpolation

Interpolation through communicating SAT solvers [Chockler Ivrii Matsliah ’12]

$P = P \land \neg m_{|A \land B}$

$P = T$
$Q = T$

$Q = Q \land \neg m_{|A \land B}$

$A \land P \land Q$

$B \land P \land Q$

$\neg P$

$Q$

SAT

UNSAT

incremental SAT solver

incremental SAT solver

Simplification techniques can only be applied locally.

Obtained interpolants are in DNF or CNF.

Clause minimization is required to obtain reasonably sized interpolants.
Proofless interpolation

Interpolation through communicating SAT solvers [Chockler Ivrii Matsliah ’12]

\[ P = P \land \neg m|_{A \land B} \]
\[ Q = Q \land \neg m|_{A \land B} \]

\[ m \models A \land P \land Q \]
\[ m \models B \land P \land Q \]

\[ A \land P \land Q \]
\[ B \land P \land Q \]

Disadvantages
- Simplification techniques can only be applied locally.
- Obtained interpolants are in DNF or CNF.
- Clause minimization is required to obtain reasonably-sized interpolants.
Proof systems for SAT solvers

A timeline of proof logging and interpolation for SAT solvers

Properties of proof systems

<table>
<thead>
<tr>
<th></th>
<th>Resolution</th>
<th>RUP</th>
<th>DRUP</th>
<th>DRAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manageable proof size</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Easily expresses CDCL</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Efficient verification</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expressive enough for inprocessing</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interpolant generation</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Proof systems for SAT solvers

A timeline of proof logging and interpolation for SAT solvers

Properties of proof systems

<table>
<thead>
<tr>
<th></th>
<th>Resolution</th>
<th>RUP</th>
<th>DRUP</th>
<th>DRAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manageable proof size</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Easily expresses CDCL</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Efficient verification</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expressive enough for inprocessing</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Interpolant generation</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
</tr>
</tbody>
</table>
Proof systems for SAT solvers

A timeline of proof logging and interpolation for SAT solvers

Interpolants from resolution proofs [Huang]

Proof-emitting SAT solvers
[Zhang, Malik]: resolution
[Goldberg, Novikov]: RUP proofs

DRUP proofs
[Heule, Hunt, Wetzler]

DRAT proofs
[Heule, Hunt, Wetzler]

Interpolants from DRUP proofs [Gurfinkel, Vizel]

Properties of proof systems

<table>
<thead>
<tr>
<th>Property</th>
<th>Resolution</th>
<th>RUP</th>
<th>DRUP</th>
<th>DRAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manageable proof size</td>
<td>✗</td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Easily expresses CDCL</td>
<td>✗</td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Efficient verification</td>
<td>✔</td>
<td></td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Expressive enough for inprocessing</td>
<td>✗</td>
<td></td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Interpolant generation</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Proof systems for SAT solvers

A timeline of proof logging and interpolation for SAT solvers

Interpolants from resolution proofs [Huang]

Proof-emaniting SAT solvers
[Zhang, Malik]: resolution
[Goldberg, Novikov]: RUP proofs

DRUP proofs [Heule, Hunt, Wetzler]

Interpolants from DRUP proofs [Gurfinkel, Vizel]

DRAT proofs [Heule, Hunt, Wetzler]

Properties of proof systems

<table>
<thead>
<tr>
<th>Property</th>
<th>Resolution</th>
<th>RUP</th>
<th>DRUP</th>
<th>DRAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manageable proof size</td>
<td>×</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Easily expresses CDCL</td>
<td>×</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Efficient verification</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Expressive enough for inprocessing</td>
<td>×</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Interpolant generation</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>
Proof systems for SAT solvers

A timeline of proof logging and interpolation for SAT solvers

Interpolants from resolution proofs [Huang]

Proof-emitting SAT solvers
[Zhang, Malik]: resolution
[Goldberg, Novikov]: RUP proofs

DRUP proofs [Heule, Hunt, Wetzler]

Interpolants from DRUP proofs [Gurfinkel, Vizel]

DRAT proofs [Heule, Hunt, Wetzler]

Properties of proof systems

<table>
<thead>
<tr>
<th></th>
<th>Resolution</th>
<th>RUP</th>
<th>DRUP</th>
<th>DRAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manageable proof size</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Easily expresses CDCL</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Efficient verification</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Expressive enough for inprocessing</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Interpolant generation</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>
Proof systems for SAT solvers

A timeline of proof logging and interpolation for SAT solvers

Properties of proof systems

<table>
<thead>
<tr>
<th></th>
<th>Resolution</th>
<th>RUP</th>
<th>DRUP</th>
<th>DRAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manageable proof size</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Easily expresses CDCL</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Efficient verification</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Expressive enough for inprocessing</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Interpolant generation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Proof systems for SAT solvers

A timeline of proof logging and interpolation for SAT solvers

Interpolants from resolution proofs [Huang]

Proof-emitting SAT solvers
- [Zhang, Malik]: resolution
- [Goldberg, Novikov]: RUP proofs

DRUP proofs
[Heule, Hunt, Wetzler]

Interpolants from DRUP proofs
[Gurfinkel, Vizel]

DRAT proofs
[Heule, Hunt, Wetzler]

Properties of proof systems

<table>
<thead>
<tr>
<th></th>
<th>Resolution</th>
<th>RUP</th>
<th>DRUP</th>
<th>DRAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manageable proof size</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Easily expresses CDCL</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Efficient verification</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Expressive enough for inprocessing</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Interpolant generation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>???</td>
</tr>
</tbody>
</table>