TU WIEN TECHNISCHE UNIVERSITÄT WIEN

JOHANNES KEPLER UNIVERSITY LINZ

Open problems in interference and proofs for SAT solving

Adrián Rebola-Pardo

Vienna University of Technology
Johannes Kepler University

Dagstuhl Seminar ‘SAT and Interactions’
Dagstuhl, Germany
October 16th, 2024

Interference-based proofs

Reverse unit propagation (RUP) [Goldberg, Novikov '03]

a clause C is a RUP clause over a CNF formula F if $F \wedge \neg C$ leads to a conflict via unit propagation

Then, $F \vDash C$ holds

Interference-based proofs

Reverse unit propagation (RUP) [Goldberg, Novikov ’03]

a clause C is a RUP clause over a CNF formula F if $F \land \neg C$ leads to a conflict via unit propagation

Then, $F \vDash C$ holds

Substitution redundance (SR) [Buss, Thapen ’19] [Gocht, Nordström ’21] [Rebola ’23]

a clause C is an SR clause over a CNF formula F upon a substitution σ if $C \lor \sigma(F \land C)$ consists of RUP clauses over F

Then, if F is satisfiable, so is $F \land C$

Interference-based proofs

Reverse unit propagation (RUP) [Goldberg, Novikov '03]

a clause C is a RUP clause over a CNF formula F if $F \land \neg C$ leads to a conflict via unit propagation

Then, $F \models C$ holds

Substitution redundance (SR) [Buss, Thapen '19] [Gocht, Nordström '21] [Rebola '23]

a clause C is an SR clause over a CNF formula F upon a substitution σ if $C \lor \sigma(F \land C)$ consists of RUP clauses over F

Then, if F is satisfiable, so is $F \land C$

Interference [Heule, Kiesl '17]

SR depends on the whole formula derived so far

this makes SR introduction non-monotonic

Interference-based proofs

Reverse unit propagation (RUP) [Goldberg, Novikov ’03]

a clause C is a RUP clause over a CNF formula F if $F \wedge \neg C$ leads to a conflict via unit propagation

Then, $F \vDash C$ holds

Substitution redundance (SR) [Buss, Thapen ’19] [Gocht, Nordström ’21] [Rebola ’23]

a clause C is an SR clause over a CNF formula F upon a substitution σ if $C \vee \sigma(F \wedge C)$ consists of RUP clauses over F

Then, if F is satisfiable, so is $F \wedge C$

Interference [Heule, Kiesl ’17]

SR depends on the whole formula derived so far

this makes SR introduction non-monotonic

What is preserved in SR beyond satisfiability? [Tinelli ’24]? [Rebola, Suda ’18]!

Interference-based proofs

Reverse unit propagation (RUP) [Goldberg, Novikov '03]

a clause C is a RUP clause over a CNF formula F if $F \wedge \neg C$ leads to a conflict via unit propagation

Then, $F \vDash C$ holds

Substitution redundance (SR) [Buss, Thapen '19] [Gocht, Nordström '21] [Rebola '23]

a clause C is an SR clause over a CNF formula F upon a substitution σ if $C \vee \sigma(F \wedge C)$ consists of RUP clauses over F

Then, if F is satisfiable, so is $F \wedge C$

Interference [Heule, Kiesl '17]

SR depends on the whole formula derived so far

this makes SR introduction non-monotonic

What is preserved in SR beyond satisfiability? [Tinelli '24]? [Rebola, Suda '18]!

Let C be SR over F upon σ. If $m \vDash F$ then $m + (\sigma : \overline{C}) \vDash F \wedge C$

Interference-based proofs

Reverse unit propagation (RUP) [Goldberg, Novikov '03]

a clause C is a RUP clause over a CNF formula F if $F \land \neg C$ leads to a conflict via unit propagation

Then, $F \vDash C$ holds

Substitution redundance (SR) [Buss, Thapen '19] [Gocht, Nordström '21] [Rebola '23]

a clause C is an SR clause over a CNF formula F upon a substitution σ if $C \lor \sigma(F \land C)$ consists of RUP clauses over F

Then, if F is satisfiable, so is $F \land C$

Interference [Heule, Kiesl '17]

SR depends on the whole formula derived so far

this makes SR introduction non-monotonic

What is preserved in SR beyond satisfiability? [Tinelli '24]? [Rebola, Suda '18]!

Let C be SR over F upon σ. If $m \vDash F$ then $m + (\sigma : \overline{C}) \vDash F \land C$

Mutation logic [Rebola, Suda '18] [Rebola '23]

$$F \vDash \nabla(\sigma : \overline{C}).\ F \land C$$

Interference-based proofs

Reverse unit propagation (RUP) [Goldberg, Novikov ’03]

a clause C is a RUP clause over a CNF formula F if $F \wedge \neg C$ leads to a conflict via unit propagation

Then, $F \vDash C$ holds

Substitution redundance (SR) [Buss, Thapen ’19] [Gocht, Nordström ’21] [Rebola ’23]

a clause C is an SR clause over a CNF formula F upon a substitution σ if $C \vee \sigma(F \wedge C)$ consists of RUP clauses over F

Then, if F is satisfiable, so is $F \wedge C$

Interference [Heule, Kiesl ’17]

SR depends on the whole formula derived so far

this makes SR introduction non-monotonic

What is preserved in SR beyond satisfiability? [Tinelli ’24]? [Rebola, Suda ’18]!

Let C be SR over F upon σ. If $m \vDash F$ then $m + (\sigma : \overline{C}) \vDash F \wedge C$

Mutation logic [Rebola, Suda ’18] [Rebola ’23]

$F \vDash \nabla(\sigma : \overline{C}).\ F \wedge C$

this is just reasoning without loss of generality!

Interpolants from interference-based proofs?

Mutation logic [Rebola, Suda ’18] [Rebola ’23]

$$F \models \nabla(\sigma : \overline{C}).\, F \wedge C$$

Interpolants from interference-based proofs?

Mutation logic [Rebola, Suda ’18] [Rebola ’23]

$$F \models \nabla(\sigma : \overline{C}).\ F \wedge C$$

We recover “natural” properties of resolution-like proof systems:

- Inferences are model-preserving
- Inferences depend only on specific clauses
- Exponentially more compact formulas *(maybe) (perhaps)*
- Satisfiability problem is NP-complete

Interpolants from interference-based proofs?

Mutation logic [Rebola, Suda '18] [Rebola '23]

$$F \models \nabla(\sigma : \overline{C}).\ F \wedge C$$

We recover "natural" properties of resolution-like proof systems:

- Inferences are model-preserving
- Inferences depend only on specific clauses
- Exponentially more compact formulas *(maybe) (perhaps)*
- Satisfiability problem is NP-complete

Craig interpolant [Craig '57]

$F \wedge G$ unsatisfiable $\Rightarrow$ $F \models P \models \neg G$ and $\mathrm{var}(P) \subseteq \mathrm{var}(F) \cap \mathrm{var}(G)$

Interpolants can be recursively computed in P-time from resolution proofs

Interpolants from interference-based proofs?

Mutation logic [Rebola, Suda '18] [Rebola '23]

$F \vDash \nabla(\sigma : \overline{C}).\ F \wedge C$

We recover "natural" properties of resolution-like proof systems:

- Inferences are model-preserving
- Inferences depend only on specific clauses
- Exponentially more compact formulas *(maybe) (perhaps)*
- Satisfiability problem is NP-complete

Craig interpolant [Craig '57]

$F \wedge G$ unsatisfiable $\Rightarrow$ $F \vDash P \vDash \neg G$ and $\mathrm{var}(P) \subseteq \mathrm{var}(F) \cap \mathrm{var}(G)$

Interpolants can be recursively computed in P-time from resolution proofs

Theorem [Reckhow '75] [Krajícek, Pudlák '98] [Kiesl, Rebola, Heule '18] [Heule, Biere, '18]

Interpolants cannot be constructed from SR proofs in polynomial time unless RSA is insecure

Open problem 1 does this result still hold for interpolants in mutation logic?

Interference-based reasoning

Despite claims to the contrary [Philipp, Rebola ’16] [Rebola, Suda ’18] [Rebola ’23] interference-based clause introduction was not very used... until recently!

Interference-based reasoning

Despite claims to the contrary [Philipp, Rebola ’16] [Rebola, Suda ’18] [Rebola ’23] **interference-based clause introduction was not very used... until recently!**

Bounded Variable Addition (BVA) [Manthey, Heule, Biere ’12]

- Turn $F \lor Q$ into $(F \lor x) \land (Q \lor \bar{x})$
  - *forgotten by 2015, enormous resurgence in 2023*

Interference-based reasoning

Despite claims to the contrary [Philipp, Rebola '16] [Rebola, Suda '18] [Rebola '23] interference-based clause introduction was not very used... until recently!

Bounded Variable Addition (BVA) [Manthey, Heule, Biere '12]

Turn $F \vee Q$ into $(F \vee x) \wedge (Q \vee \bar{x})$

forgotten by 2015, enormous resurgence in 2023

Satisfaction-driven clause learning (SDCL) [Heule, Kiesl, Seidl, Biere '17]

Encode "*C is an SR clause over F upon σ*" in a sub-solver

modest effect, enormous resurgence in 2023 *[Oliveras, Li, Wu, Chung, Ganesh '23]*

The encoding needs to account for the whole formula and every polarity of every variable

Interference-based reasoning

Despite claims to the contrary [Philipp, Rebola ’16] [Rebola, Suda ’18] [Rebola ’23] interference-based clause introduction was not very used... until recently!

Bounded Variable Addition (BVA) [Manthey, Heule, Biere ’12]

Turn $F \vee Q$ into $(F \vee x) \wedge (Q \vee \bar{x})$

forgotten by 2015, enormous resurgence in 2023

Satisfaction-driven clause learning (SDCL) [Heule, Kiesl, Seidl, Biere ’17]

Encode “*C is an SR clause over F upon σ*” in a sub-solver

modest effect, enormous resurgence in 2023 *[Oliveras, Li, Wu, Chung, Ganesh ’23]*

The encoding needs to account for the whole formula and every polarity of every variable

Idea: splitting-based inprocessing

Perform SDCL/BVA over a subformula $G \subseteq F$, then identify clauses that can be brought back into F.

Can be proven sound in mutation logic, but not even in WSR

Similar problems in incremental solving [Fazekas, Biere, Scholl ’19]

Interference-based reasoning

Despite claims to the contrary [Philipp, Rebola '16] [Rebola, Suda '18] [Rebola '23] interference-based clause introduction was not very used... until recently!

Bounded Variable Addition (BVA) [Manthey, Heule, Biere '12]

Turn $F \lor Q$ into $(F \lor x) \land (Q \lor \bar{x})$

forgotten by 2015, enormous resurgence in 2023

Satisfaction-driven clause learning (SDCL) [Heule, Kiesl, Seidl, Biere '17]

Encode "*C is an SR clause over F upon σ*" in a sub-solver

modest effect, enormous resurgence in 2023 *[Oliveras, Li, Wu, Chung, Ganesh '23]*

The encoding needs to account for the whole formula and every polarity of every variable

Idea: splitting-based inprocessing

Perform SDCL/BVA over a subformula $G \subseteq F$, then identify clauses that can be brought back into F.

Can be proven sound in mutation logic, but not even in WSR

Similar problems in incremental solving [Fazekas, Biere, Scholl '19]

Open problem 2 How does entailment behaves across interference?

are we being artificially limited by how we think of redundancy/interference?

A critical history of proof formats for CDCL SAT

[Goldberg, Novikov ’03]

RUP

○

[Malik, Zhang ’03]

resolution

○

$$\dfrac{C \lor x \qquad D \lor \overline{x}}{C \lor D}$$

resolution

$$\dfrac{\dfrac{E_0}{A_0} \qquad E_1}{A_1}$$

$$\ddots$$

$$\dfrac{A_{n-1} \qquad E_n}{A_n}$$

RUP

propagate $\overline{A_n}$, check for conflict

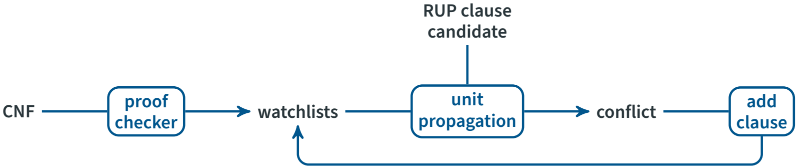
A critical history of proof formats for CDCL SAT

RUP

○

resolution

○



RUP clause candidate

CNF

proof checker

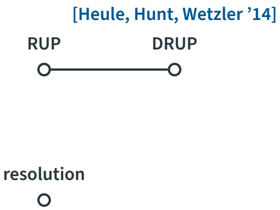
watchlists

unit propagation

conflict

add clause

A critical history of proof formats for CDCL SAT

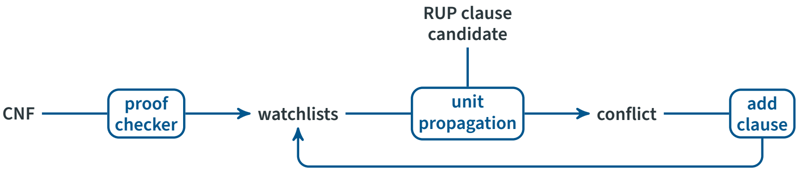


[Heule, Hunt, Wetzler ’14]

RUP

DRUP

resolution



RUP clause candidate

CNF

proof checker

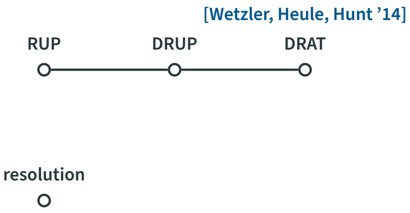
watchlists

unit propagation

conflict

add clause

A critical history of proof formats for CDCL SAT



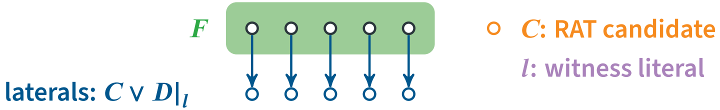
[Wetzler, Heule, Hunt ’14]

RUP

DRUP

DRAT

resolution



F

laterals: $C \lor D|_l$

C: RAT candidate

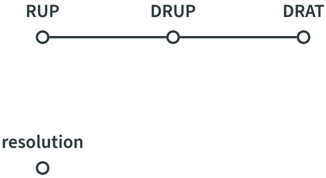
l: witness literal

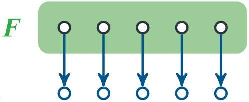
i) laterals must be RUP

ii) $l \models C$

[Järvisalo, Heule, Biere ’12]

A critical history of proof formats for CDCL SAT





○ C: PR candidate

Q: witness cube

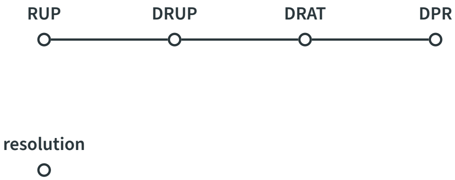
laterals: $C \lor D|_Q$

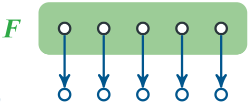
i) laterals must be RUP

ii) $Q \vDash C$

A critical history of proof formats for CDCL SAT

[Heule, Kiesl, Biere ’17]





laterals: $C \lor D|_Q$

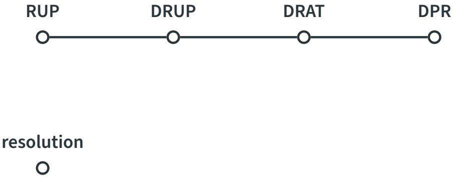
○ C: PR candidate

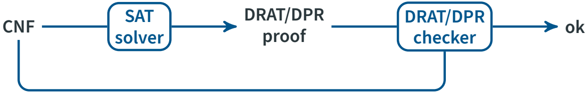
Q: witness cube

i) laterals must be RUP

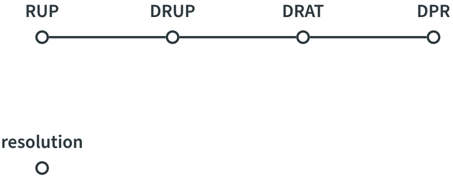
ii) $Q \vDash C$

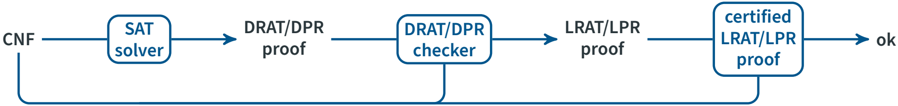
A critical history of proof formats for CDCL SAT


RUP
DRUP
DRAT
DPR
resolution

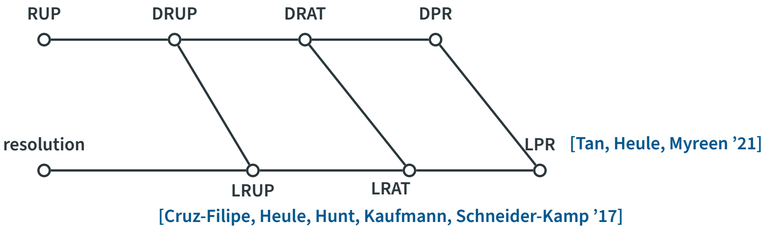

CNF
SAT solver
DRAT/DPR proof
DRAT/DPR checker
ok

A critical history of proof formats for CDCL SAT



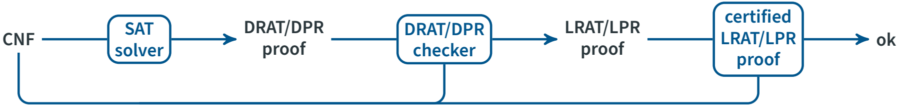


A critical history of proof formats for CDCL SAT

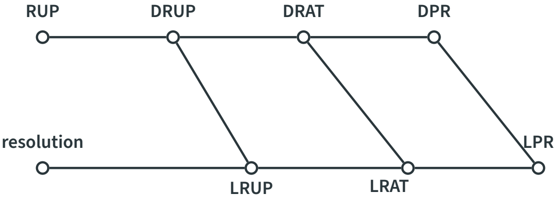


[Tan, Heule, Myreen '21]

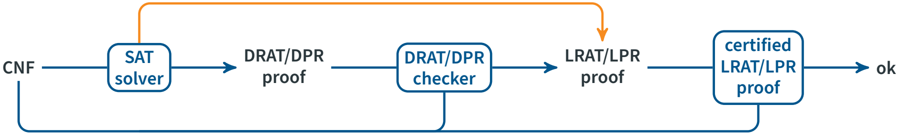
[Cruz-Filipe, Heule, Hunt, Kaufmann, Schneider-Kamp '17]



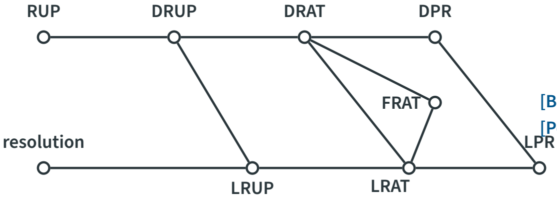
A critical history of proof formats for CDCL SAT



“There are several challenges regarding unsatisfiability proofs. How can one store resolution proofs using much less space on disk and using much less memory overhead?”
[Biere, Heule ’15]



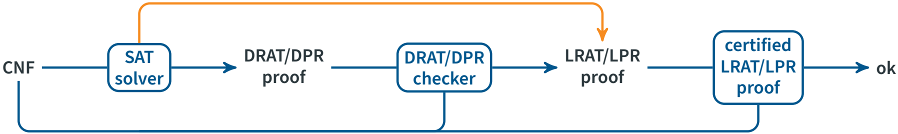
A critical history of proof formats for CDCL SAT



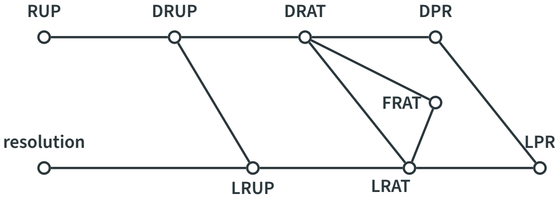
[Baek, Carneiro, Heule ’21]

[Pollitt, Fleury, Biere ’23]

“There are several challenges regarding unsatisfiability proofs. How can one store resolution proofs using much less space on disk and using much less memory overhead?”
[Biere, Heule ’15]

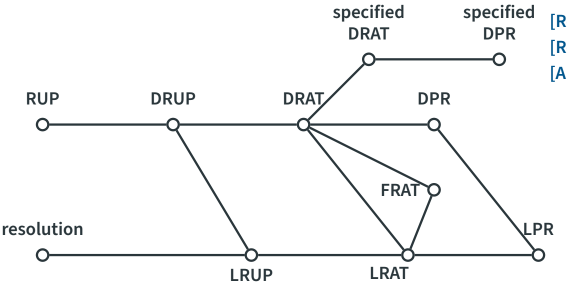


A critical history of proof formats for CDCL SAT



“The main reason to add deletion information to a clausal proof is to reduce the computation costs to validate that proof. However, deletion of unit clauses has the opposite effect. Notice that ignoring deletion steps of unit clauses can turn a valid DRAT proof into an invalid one — and the other way around.” [Heule ’16]

A critical history of proof formats for CDCL SAT



[Rebola-Pardo, Biere ’18]
[Rebola-Pardo, Cruz-Felipe ’18]
[Altmanninger, Rebola-Pardo ’20]

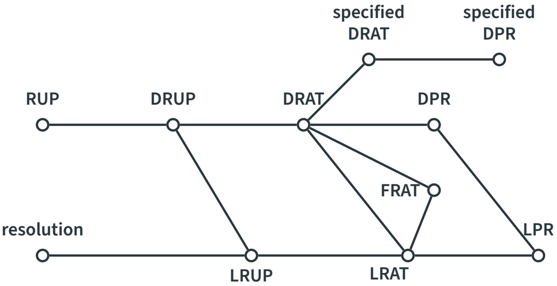
“The main reason to add deletion information to a clausal proof is to reduce the computation costs to validate that proof. However, deletion of unit clauses has the opposite effect. Notice that ignoring deletion steps of unit clauses can turn a valid DRAT proof into an invalid one — and the other way around.” [Heule ’16]

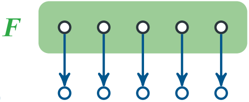
Problem 1 checking if a clause is unit requires unit propagation

Problem 2 DRAT is non-monotonic, so skipping a deletion gives surprising results

Problem 3 the computational cost is the same (in our implementation, slightly lower)

A critical history of proof formats for CDCL SAT





laterals: $C \vee D|_Q$

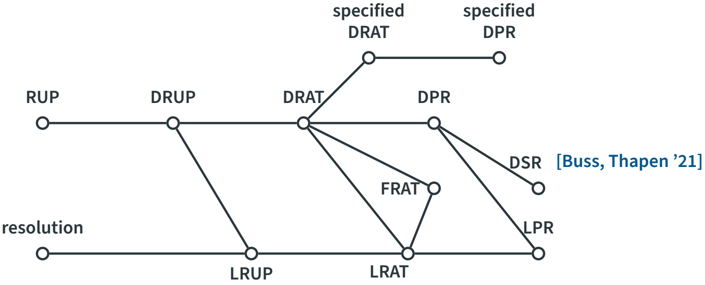
○ C: PR candidate

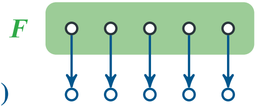
Q: witness cube

i) laterals must be RUP

ii) $Q \models C$

A critical history of proof formats for CDCL SAT





laterals: $C \vee \sigma(D)$

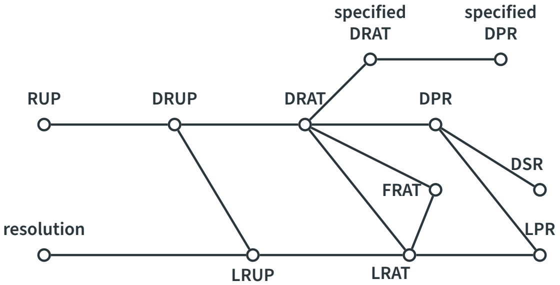
○ C: SR candidate

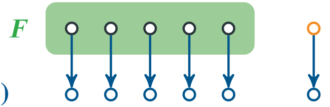
σ: witness substitution

i) laterals must be RUP

ii) $\sigma(C)$

A critical history of proof formats for CDCL SAT





laterals: $C \vee \sigma(D)$

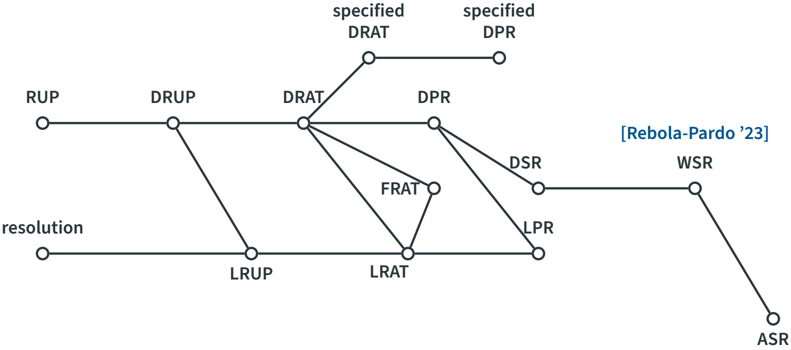
C: SR candidate

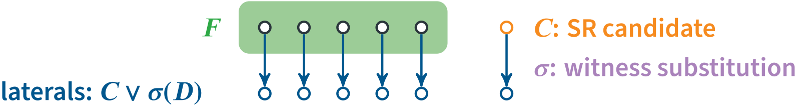
σ: witness substitution

i) laterals must be RUP

[Gocht, Nordström ’21]
[Rebola-Pardo ’23]

A critical history of proof formats for CDCL SAT

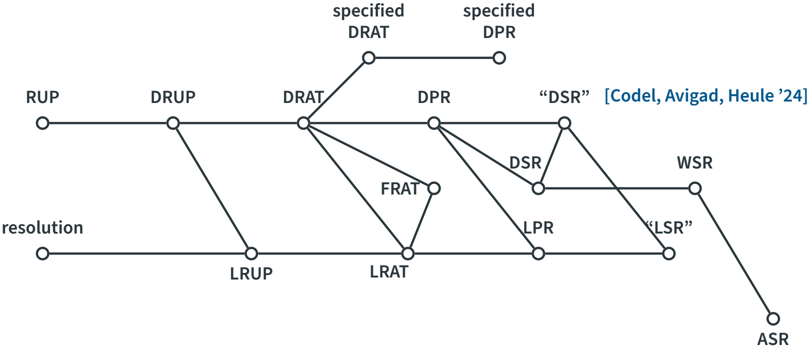




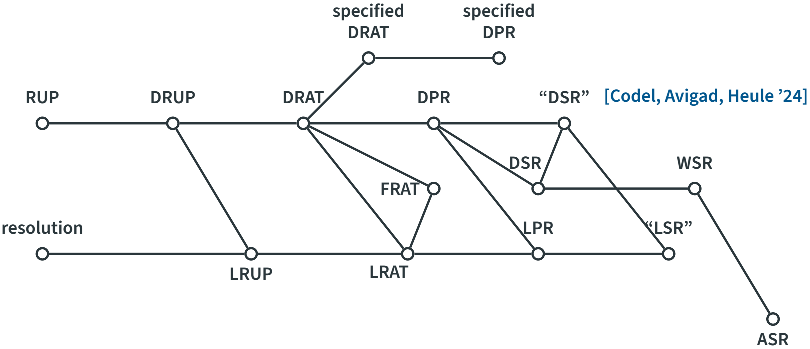
i) laterals must be RUP

[Gocht, Nordström '21]
[Rebola-Pardo '23]

A critical history of proof formats for CDCL SAT



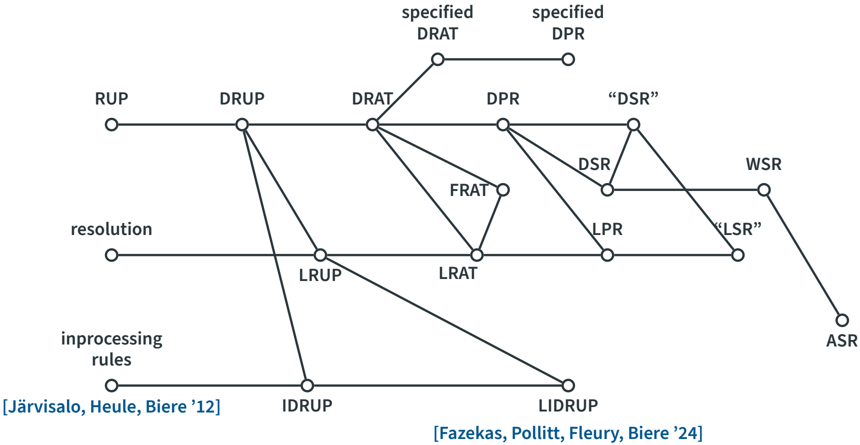
A critical history of proof formats for CDCL SAT



“Currently, our checkers assume that the witness σ satisfies the candidate clause C. However, the DSR and LSR formats can also express proofs where σ causes $C|_\sigma$ to be a tautology: the proof can simply map the pivot p to itself or to any other literal in the substitution portion. We plan to support this general case in the future.”

“At the moment, `dsr-trim` can only perform forwards checking, which means that it checks DSR proofs from start to finish and adds hints as it goes. (…) In practice, backwards checking can significantly reduce the size of proofs. Adding backwards checking to `dsr-trim` is ongoing work.” [Codel, Avigad, Heule ’24]

A critical history of proof formats for CDCL SAT



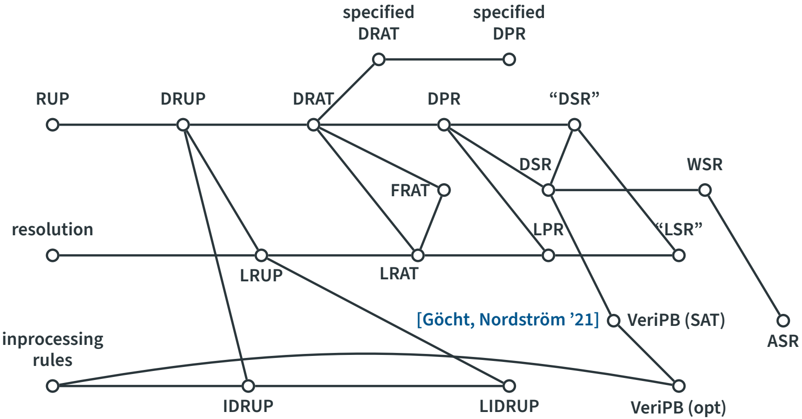
$$\frac{\varphi \wedge C\ [\rho]\ \sigma}{\varphi\ [\rho]\ \sigma \cdot (\omega : C)}\ \boxed{\flat} \qquad \frac{\varphi \wedge C\ [\rho]\ \sigma}{\varphi\ [\rho]\ \sigma}\ \boxed{\varnothing}$$

$\textsc{Weaken}^{+}$ $\qquad$ $\textsc{Drop}$

where $\boxed{\flat}$ is $\varphi \wedge C \equiv^{\omega}_{sat} \varphi$ and $\boxed{\varnothing}$ is $\varphi \models C$

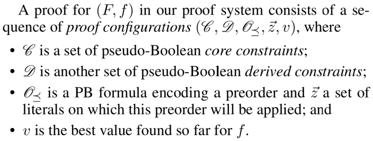
Fig. 2. New weakening and dropping rules

A critical history of proof formats for CDCL SAT



[Göcht, Nordström '21]

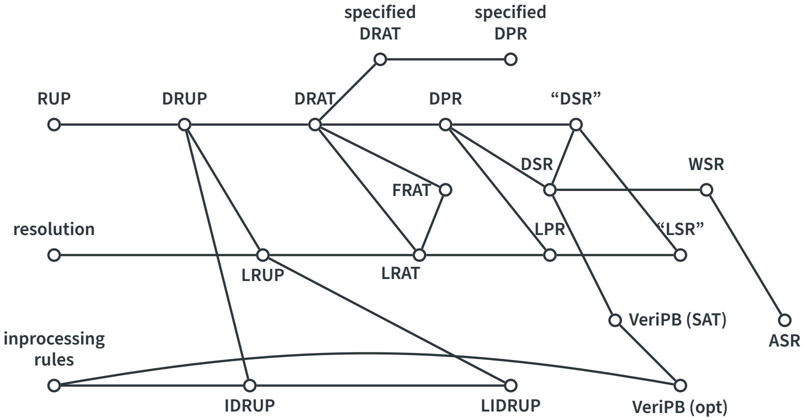
[Bogaerts, Gocht, McCreesh, Nordström '22]



A proof for (F, f) in our proof system consists of a sequence of *proof configurations* $(\mathscr{C}, \mathscr{D}, \mathscr{O}_{\preceq}, \vec{z}, v)$, where

- $\mathscr{C}$ is a set of pseudo-Boolean *core constraints*;
- $\mathscr{D}$ is another set of pseudo-Boolean *derived constraints*;
- $\mathscr{O}_{\preceq}$ is a PB formula encoding a preorder and $\vec{z}$ a set of literals on which this preorder will be applied; and
- v is the best value found so far for f.

A critical history of proof formats for CDCL SAT



… plus binary vs text variants!

Proof formats: what do we need, what must go?

Open Problem 3 consolidating proof formats

Proof formats: what do we need, what must go?

Open Problem 3 consolidating proof formats

Are binary formats needed? XFS + raw byte encoding? [idea by Max Heisinger]

Proof formats: what do we need, what must go?

Open Problem 3 consolidating proof formats

Are binary formats needed? XFS + raw byte encoding? [idea by Max Heisinger]

Are text formats needed? a filter tool could suffice [Matthias Fleury disagrees]

Proof formats: what do we need, what must go?

Open Problem 3 consolidating proof formats

Are binary formats needed? XFS + raw byte encoding? [idea by Max Heisinger]

Are text formats needed? a filter tool could suffice [Matthias Fleury disagrees]

Are clausal proofs still needed? (SAT only)

Proof formats: what do we need, what must go?

Open Problem 3 consolidating proof formats

Are binary formats needed? XFS + raw byte encoding? [idea by Max Heisinger]

Are text formats needed? a filter tool could suffice [Matthias Fleury disagrees]

Are clausal proofs still needed? (SAT only)

Implicit clause ids? trade-off with solvers! [Matthias Fleury]

Proof formats: what do we need, what must go?

Open Problem 3 consolidating proof formats

Are binary formats needed? XFS + raw byte encoding? [idea by Max Heisinger]

Are text formats needed? a filter tool could suffice [Matthias Fleury disagrees]

Are clausal proofs still needed? (SAT only)

Implicit clause ids? trade-off with solvers! [Matthias Fleury]

Zero-terminated vs length-prefixed lists? is this just about readability?

Proof formats: what do we need, what must go?

Open Problem 3 consolidating proof formats

Are binary formats needed? XFS + raw byte encoding? [idea by Max Heisinger]

Are text formats needed? a filter tool could suffice [Matthias Fleury disagrees]

Are clausal proofs still needed? (SAT only)

Implicit clause ids? trade-off with solvers! [Matthias Fleury]

Zero-terminated vs length-prefixed lists? is this just about readability?

Explicit SR vs RUP introductions? [Cynthia Peyrer, Ilija Vorontsov]

Proof formats: what do we need, what must go?

Open Problem 3 consolidating proof formats

Are binary formats needed? XFS + raw byte encoding? [idea by Max Heisinger]

Are text formats needed? a filter tool could suffice [Matthias Fleury disagrees]

Are clausal proofs still needed? (SAT only)

Implicit clause ids? trade-off with solvers! [Matthias Fleury]

Zero-terminated vs length-prefixed lists? is this just about readability?

Explicit SR vs RUP introductions? [Cynthia Peyrer, Ilija Vorontsov]

Beyond CNF reasoning? should the format be extensible?

Proof formats: what do we need, what must go?

Open Problem 3 consolidating proof formats

Are binary formats needed? XFS + raw byte encoding? [idea by Max Heisinger]

Are text formats needed? a filter tool could suffice [Matthias Fleury disagrees]

Are clausal proofs still needed? (SAT only)

Implicit clause ids? trade-off with solvers! [Matthias Fleury]

Zero-terminated vs length-prefixed lists? is this just about readability?

Explicit SR vs RUP introductions? [Cynthia Peyrer, Ilija Vorontsov]

Beyond CNF reasoning? should the format be extensible?

Unified framework for satisfiable instances? for incremental SAT and optimization

Proof formats: what do we need, what must go?

Open Problem 3 consolidating proof formats

Are binary formats needed? XFS + raw byte encoding? [idea by Max Heisinger]

Are text formats needed? a filter tool could suffice [Matthias Fleury disagrees]

Are clausal proofs still needed? (SAT only)

Implicit clause ids? trade-off with solvers! [Matthias Fleury]

Zero-terminated vs length-prefixed lists? is this just about readability?

Explicit SR vs RUP introductions? [Cynthia Peyrer, Ilija Vorontsov]

Beyond CNF reasoning? should the format be extensible?

Unified framework for satisfiable instances? for incremental SAT and optimization

Partial proof checking? unit deletions are tricky here!

Proof formats: what do we need, what must go?

Open Problem 3 consolidating proof formats

Are binary formats needed? XFS + raw byte encoding? [idea by Max Heisinger]

Are text formats needed? a filter tool could suffice [Matthias Fleury disagrees]

Are clausal proofs still needed? (SAT only)

Implicit clause ids? trade-off with solvers! [Matthias Fleury]

Zero-terminated vs length-prefixed lists? is this just about readability?

Explicit SR vs RUP introductions? [Cynthia Peyrer, Ilija Vorontsov]

Beyond CNF reasoning? should the format be extensible?

Unified framework for satisfiable instances? for incremental SAT and optimization

Partial proof checking? unit deletions are tricky here!

Political theory? should provers adapt to checkers or checkers to provers?

Proof formats: what do we need, what must go?

Open Problem 3 consolidating proof formats

Are binary formats needed? XFS + raw byte encoding? [idea by Max Heisinger]

Are text formats needed? a filter tool could suffice [Matthias Fleury disagrees]

Are clausal proofs still needed? (SAT only)

Implicit clause ids? trade-off with solvers! [Matthias Fleury]

Zero-terminated vs length-prefixed lists? is this just about readability?

Explicit SR vs RUP introductions? [Cynthia Peyrer, Ilija Vorontsov]

Beyond CNF reasoning? should the format be extensible?

Unified framework for satisfiable instances? for incremental SAT and optimization

Partial proof checking? unit deletions are tricky here!

Political theory? should provers adapt to checkers or checkers to provers?

What do we certify? what counts as a “correct” proof? [yes, again]

The many flavors of deletion

Linear deletion (DRAT, DPR, DSR) [Heule, Hunt, Wetzler ’14]

clauses can be introduced or deleted, but only sequentially

The many flavors of deletion

Linear deletion (DRAT, DPR, DSR) [Heule, Hunt, Wetzler ’14]

clauses can be introduced or deleted, but only sequentially

Covert deletion (WSR) [Rebola ’23]

when introducing a clause, other clauses can be deleted concurrently

enabling redundancy lemmas, efficient backwards checking, smaller unsat cores

The many flavors of deletion

Linear deletion (DRAT, DPR, DSR) [Heule, Hunt, Wetzler ’14]

clauses can be introduced or deleted, but only sequentially

Covert deletion (WSR) [Rebola ’23]

when introducing a clause, other clauses can be deleted concurrently

enabling redundancy lemmas, efficient backwards checking, smaller unsat cores

Core-based deletion (IDRUP, VeriPB) [Fazekas, Pollitt, Fleury, Biere ’24] [Bogaerts, Gocht, McCreesh, Nordström ’22]

deletions have requirements based on two privileged sets of clauses

needed for satisfiability proofs, e.g. incremental SAT and optimization

The many flavors of deletion

Linear deletion (DRAT, DPR, DSR) [Heule, Hunt, Wetzler ’14]

clauses can be introduced or deleted, but only sequentially

Covert deletion (WSR) [Rebola ’23]

when introducing a clause, other clauses can be deleted concurrently

enabling redundancy lemmas, efficient backwards checking, smaller unsat cores

Core-based deletion (IDRUP, VeriPB) [Fazekas, Pollitt, Fleury, Biere ’24] [Bogaerts, Gocht, McCreesh, Nordström ’22]

deletions have requirements based on two privileged sets of clauses

needed for satisfiability proofs, e.g. incremental SAT and optimization

Scope-based deletion

create scopes on the fly, move clauses between them

needed for splitting-based inprocessing

The many flavors of deletion

Linear deletion (DRAT, DPR, DSR) [Heule, Hunt, Wetzler ’14]

clauses can be introduced or deleted, but only sequentially

Covert deletion (WSR) [Rebola ’23]

when introducing a clause, other clauses can be deleted concurrently

enabling redundancy lemmas, efficient backwards checking, smaller unsat cores

Core-based deletion (IDRUP, VeriPB) [Fazekas, Pollitt, Fleury, Biere ’24] [Bogaerts, Gocht, McCreesh, Nordström ’22]

deletions have requirements based on two privileged sets of clauses

needed for satisfiability proofs, e.g. incremental SAT and optimization

Scope-based deletion

create scopes on the fly, move clauses between them

needed for splitting-based inprocessing

Open Problem 4 can we find a sensible framework unifying these ideas?