On Proofs for SAT and QBF

Benjamin Kiesl

Vienna University of Technology, Institute of Information Systems

Topic of the Talk

- This talk is about our work during the last four months.

Topic of the Talk

- This talk is about our work during the last four months.
- Our work centered around proofs for SAT and QBF. It resulted in two papers and one abstract:

Topic of the Talk

- This talk is about our work during the last four months.
- Our work centered around proofs for SAT and QBF. It resulted in two papers and one abstract:

  Marijn J.H. Heule, Benjamin Kiesl, and Armin Biere:
  Short Proofs Without New Variables
  (Accepted at CADE)

Topic of the Talk

- This talk is about our work during the last four months.
- Our work centered around proofs for SAT and QBF. It resulted in two papers and one abstract:

  Marijn J.H. Heule, Benjamin Kiesl, and Armin Biere:
  Short Proofs Without New Variables
  (Accepted at CADE)

  Benjamin Kiesl, Marijn J.H. Heule, and Martina Seidl:
  A Little Blocked Literal Goes a Long Way
  (Submitted to SAT)

Topic of the Talk

- This talk is about our work during the last four months.
- Our work centered around proofs for SAT and QBF. It resulted in two papers and one abstract:

  Marijn J.H. Heule, Benjamin Kiesl, and Armin Biere:
  Short Proofs Without New Variables
  (Accepted at CADE)

  Benjamin Kiesl, Marijn J.H. Heule, and Martina Seidl:
  A Little Blocked Literal Goes a Long Way
  (Submitted to SAT)

  Marijn J.H. Heule and Benjamin Kiesl:
  The Potential of Interference-Based Proof Systems
  (Extended Abstract, Submitted to the ARCADE workshop)

Outline

- Overview on SAT and corresponding proofs.
  - What are proofs and why do we care about them?

Outline

- Overview on SAT and corresponding proofs.
  - What are proofs and why do we care about them?
- Short summary of our first paper:
  - In the paper, we introduce new proof systems for SAT solving.

Outline

- Overview on SAT and corresponding proofs.
  - What are proofs and why do we care about them?
- Short summary of our first paper:
  - In the paper, we introduce new proof systems for SAT solving.
- Short summary of our second paper:
  - We show how two important proof systems for QBF are related.

The Satisfiability Problem of Propositional Logic (SAT)

- Given a propositional formula in conjunctive normal form, is it satisfiable?

The Satisfiability Problem of Propositional Logic (SAT)

- Given a propositional formula in conjunctive normal form, is it satisfiable?
- A literal is a variable x or the negation $\bar{x}$ of a variable x.

The Satisfiability Problem of Propositional Logic (SAT)

- Given a propositional formula in conjunctive normal form, is it satisfiable?
- A literal is a variable x or the negation $\bar{x}$ of a variable x.
- A clause is a disjunction $l_1 \vee \cdots \vee l_n$ of literals.

The Satisfiability Problem of Propositional Logic (SAT)

- Given a propositional formula in conjunctive normal form, is it satisfiable?
- A literal is a variable x or the negation $\bar{x}$ of a variable x.
- A clause is a disjunction $l_1 \vee \cdots \vee l_n$ of literals.
- A formula (in CNF) is a conjunction $C_1 \wedge \cdots \wedge C_n$ of clauses.

The Satisfiability Problem of Propositional Logic (SAT)

- Given a propositional formula in conjunctive normal form, is it satisfiable?
- A literal is a variable x or the negation $\bar{x}$ of a variable x.
- A clause is a disjunction $l_1 \vee \cdots \vee l_n$ of literals.
- A formula (in CNF) is a conjunction $C_1 \wedge \cdots \wedge C_n$ of clauses.
- Example:

$$(a \vee \bar{b}) \wedge (c) \wedge (\bar{a} \vee \bar{c})$$

The Satisfiability Problem of Propositional Logic (SAT)

- A (truth) assignment is a mapping from variables to the truth values 0 (false) and 1 (true).

The Satisfiability Problem of Propositional Logic (SAT)

- A (truth) assignment is a mapping from variables to the truth values 0 (false) and 1 (true).
- An assignment τ satisfies ...
  - ... a variable x if $\tau(x) = 1$.

The Satisfiability Problem of Propositional Logic (SAT)

- A (truth) assignment is a mapping from variables to the truth values 0 (false) and 1 (true).
- An assignment τ satisfies ...
  - ... a variable x if $\tau(x) = 1$.
  - ... a literal l if $l = x$ and $\tau(x) = 1$, or $l = \bar{x}$ and $\tau(x) = 0$.

The Satisfiability Problem of Propositional Logic (SAT)

- A (truth) assignment is a mapping from variables to the truth values 0 (false) and 1 (true).
- An assignment τ satisfies ...
  - ... a variable x if $\tau(x) = 1$.
  - ... a literal l if $l = x$ and $\tau(x) = 1$, or $l = \bar{x}$ and $\tau(x) = 0$.
  - ... a clause if it satisfies at least one literal in the clause.

The Satisfiability Problem of Propositional Logic (SAT)

- A (truth) assignment is a mapping from variables to the truth values 0 (false) and 1 (true).
- An assignment τ satisfies ...
  - ... a variable x if $\tau(x) = 1$.
  - ... a literal l if $l = x$ and $\tau(x) = 1$, or $l = \bar{x}$ and $\tau(x) = 0$.
  - ... a clause if it satisfies at least one literal in the clause.
  - ... a formula if it satisfies all its clauses.

The Satisfiability Problem of Propositional Logic (SAT)

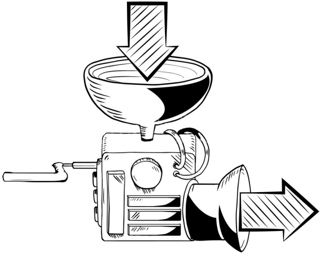
- A (truth) assignment is a mapping from variables to the truth values 0 (false) and 1 (true).
- An assignment τ satisfies ...
  - ... a variable x if $\tau(x) = 1$.
  - ... a literal l if $l = x$ and $\tau(x) = 1$, or $l = \bar{x}$ and $\tau(x) = 0$.
  - ... a clause if it satisfies at least one literal in the clause.
  - ... a formula if it satisfies all its clauses.

➥ SAT:

- Given a formula F, does there exist an assignment that satisfies F?

The Satisfiability Problem of Propositional Logic (SAT)

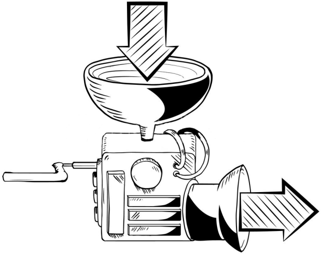
$$(x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$$



The Satisfiability Problem of Propositional Logic (SAT)

Input Formula

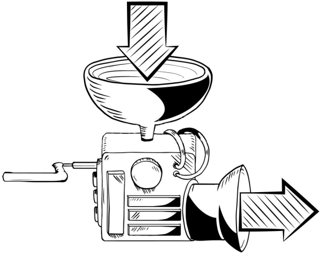
$$(x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$$



The Satisfiability Problem of Propositional Logic (SAT)

Formulas can be seen as sets of clauses

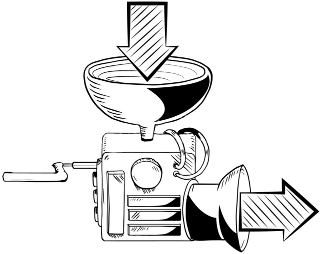
$$\left\{ x \vee y, \quad \bar{x} \vee \bar{y}, \quad z \vee \bar{z} \right\}$$



The Satisfiability Problem of Propositional Logic (SAT)

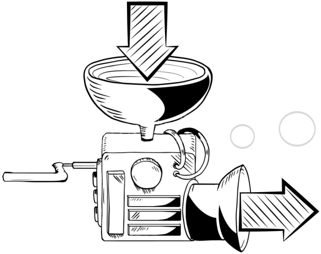
Clauses can be seen as sets of literals

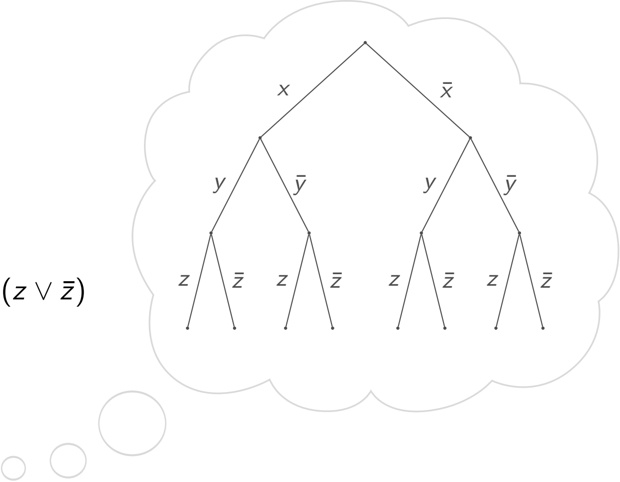
$$\big\{\{x, y\}, \quad \{\bar{x}, \bar{y}\}, \quad \{z, \bar{z}\}\big\}$$



The Satisfiability Problem of Propositional Logic (SAT)

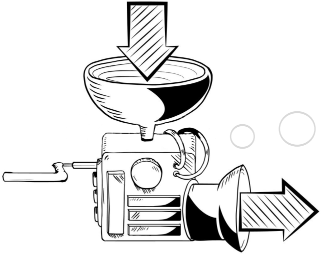
$(x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$

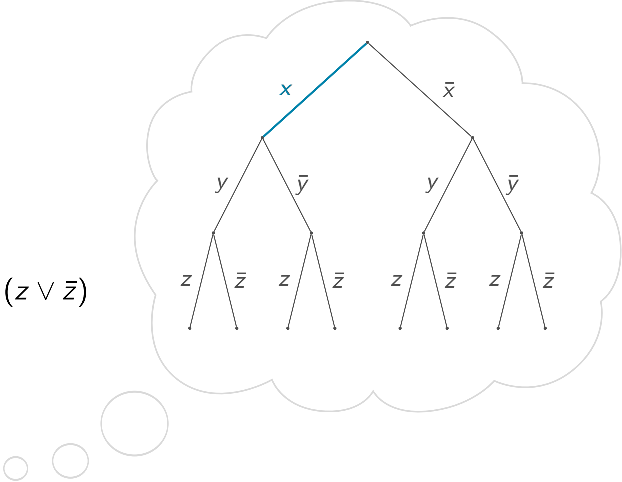




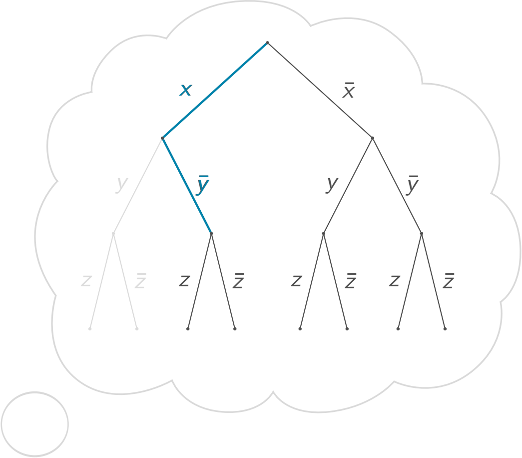
The Satisfiability Problem of Propositional Logic (SAT)

$(x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$

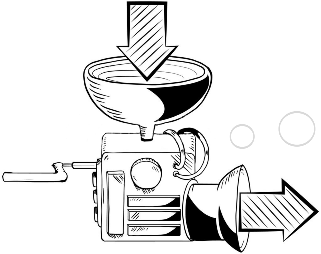




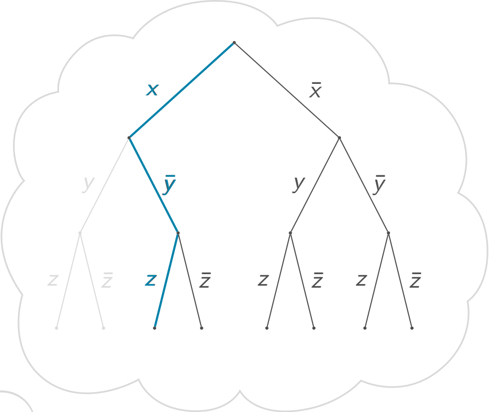
The Satisfiability Problem of Propositional Logic (SAT)



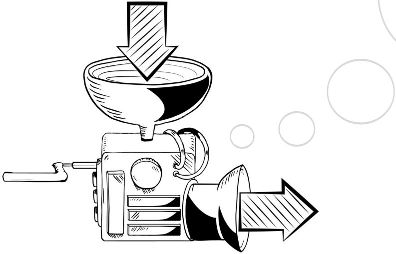
$(x \lor y) \land (\bar{x} \lor \bar{y}) \land (z \lor \bar{z})$



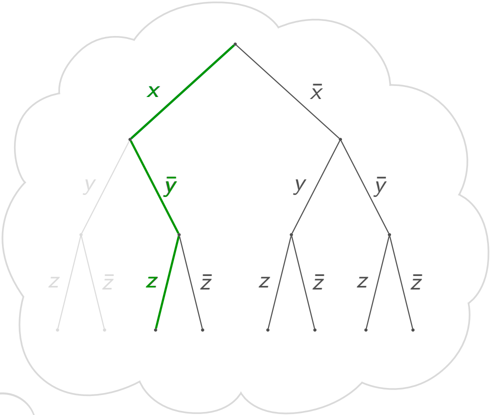
The Satisfiability Problem of Propositional Logic (SAT)



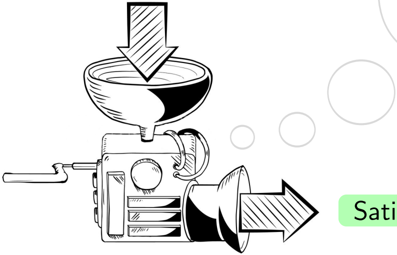
$(x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$



The Satisfiability Problem of Propositional Logic (SAT)



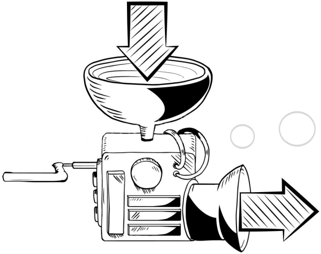
$(x \lor y) \land (\bar{x} \lor \bar{y}) \land (z \lor \bar{z})$

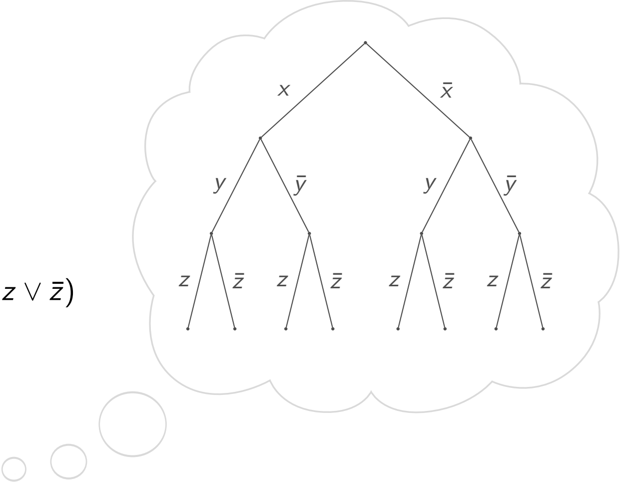


Satisfiable

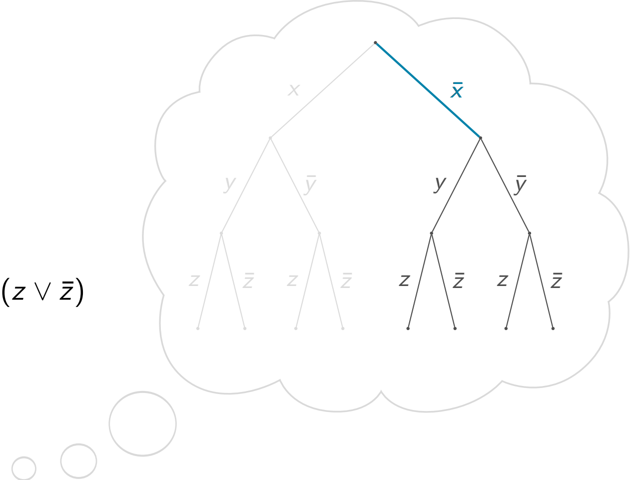
The Satisfiability Problem of Propositional Logic (SAT)

$\bar{x} \wedge y \wedge (x \vee \bar{y}) \wedge (z \vee \bar{z})$

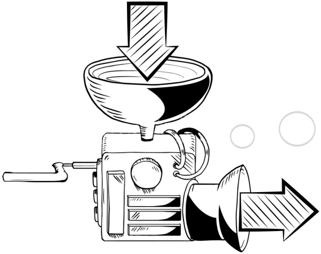




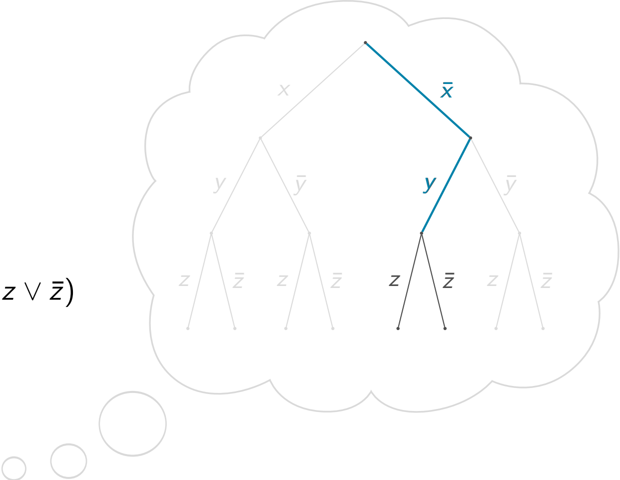
The Satisfiability Problem of Propositional Logic (SAT)



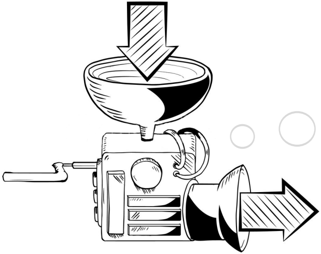
$\bar{x} \wedge y \wedge (x \vee \bar{y}) \wedge (z \vee \bar{z})$



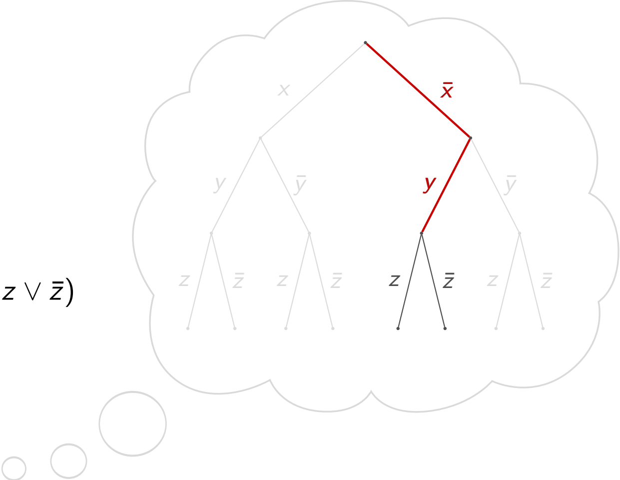
The Satisfiability Problem of Propositional Logic (SAT)



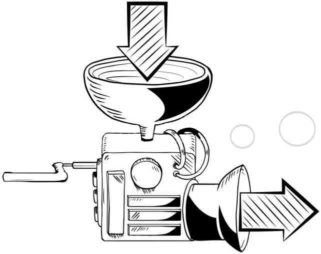
$$\bar{x} \wedge y \wedge (x \vee \bar{y}) \wedge (z \vee \bar{z})$$



The Satisfiability Problem of Propositional Logic (SAT)



$\bar{x} \wedge y \wedge (x \vee \bar{y}) \wedge (z \vee \bar{z})$



Unsatisfiable

Certifying Satisfiability and Unsatisfiability

- Certifying satisfiability of a formula is easy:

Certifying Satisfiability and Unsatisfiability

- Certifying satisfiability of a formula is easy:
  - Just consider a satisfying assignment:

$$(x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$$

Certifying Satisfiability and Unsatisfiability

- Certifying satisfiability of a formula is easy:
  - Just consider a satisfying assignment: $x\bar{y}z$

$$(x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$$

Certifying Satisfiability and Unsatisfiability

- Certifying satisfiability of a formula is easy:
  - Just consider a satisfying assignment: $x\bar{y}z$

$$(x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$$

  - We can easily check that the assignment is satisfying:
    Just check for every clause if it has a satisfied literal!

Certifying Satisfiability and Unsatisfiability

- Certifying satisfiability of a formula is easy:
  - Just consider a satisfying assignment: $x\bar{y}z$

$$(x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$$

  - We can easily check that the assignment is satisfying:
    Just check for every clause if it has a satisfied literal!
- Certifying unsatisfiability is not so easy:

Certifying Satisfiability and Unsatisfiability

- Certifying satisfiability of a formula is easy:
  - Just consider a satisfying assignment: $x\bar{y}z$

$$(x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$$

  - We can easily check that the assignment is satisfying:
    Just check for every clause if it has a satisfied literal!
- Certifying unsatisfiability is not so easy:
  - If a formula has n variables, there are 2^n possible assignments.

Certifying Satisfiability and Unsatisfiability

- Certifying satisfiability of a formula is easy:
  - Just consider a satisfying assignment: $x\bar{y}z$

$$(x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$$

  - We can easily check that the assignment is satisfying:
    Just check for every clause if it has a satisfied literal!
- Certifying unsatisfiability is not so easy:
  - If a formula has n variables, there are 2^n possible assignments.
  - ➥ Checking whether every assignment falsifies the formula is costly.

Certifying Satisfiability and Unsatisfiability

- Certifying satisfiability of a formula is easy:
  - Just consider a satisfying assignment: $x\bar{y}z$

$$(x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$$

  - We can easily check that the assignment is satisfying:
    Just check for every clause if it has a satisfied literal!
- Certifying unsatisfiability is not so easy:
  - If a formula has n variables, there are 2^n possible assignments.
  - ➥ Checking whether every assignment falsifies the formula is costly.
  - More compact certificates of unsatisfiability are desirable.

Certifying Satisfiability and Unsatisfiability

- Certifying satisfiability of a formula is easy:
  - Just consider a satisfying assignment: $x\bar{y}z$

$$(x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$$

  - We can easily check that the assignment is satisfying:
    Just check for every clause if it has a satisfied literal!
- Certifying unsatisfiability is not so easy:
  - If a formula has n variables, there are 2^n possible assignments.
  - ➥ Checking whether every assignment falsifies the formula is costly.
  - More compact certificates of unsatisfiability are desirable.
    - ➥ Proofs

What Is a Proof in SAT?

- In general, a proof is a string that certifies the unsatisfiability of a formula.

What Is a Proof in SAT?

- In general, a proof is a string that certifies the unsatisfiability of a formula.
  - Proofs are efficiently (usually polynomial-time) checkable.

What Is a Proof in SAT?

- In general, a proof is a string that certifies the unsatisfiability of a formula.
  - Proofs are efficiently (usually polynomial-time) checkable (but can be of exponential size with respect to a formula)

What Is a Proof in SAT?

- In general, a proof is a string that certifies the unsatisfiability of a formula.
  - Proofs are efficiently (usually polynomial-time) checkable (but can be of exponential size with respect to a formula)
- Example: Resolution proofs

What Is a Proof in SAT?

- In general, a proof is a string that certifies the unsatisfiability of a formula.
  - Proofs are efficiently (usually polynomial-time) checkable (but can be of exponential size with respect to a formula)
- Example: Resolution proofs
  - A resolution proof is a sequence $C_1, \ldots, C_n$ of clauses.

What Is a Proof in SAT?

- In general, a proof is a string that certifies the unsatisfiability of a formula.
  - Proofs are efficiently (usually polynomial-time) checkable (but can be of exponential size with respect to a formula)
- Example: Resolution proofs
  - A resolution proof is a sequence $C_1, \ldots, C_n$ of clauses.
  - Every clause is either contained in the formula or derived from two earlier clauses via the resolution rule:

$$\frac{C \vee l \qquad \bar{l} \vee D}{C \vee D}$$

What Is a Proof in SAT?

- In general, a proof is a string that certifies the unsatisfiability of a formula.
  - Proofs are efficiently (usually polynomial-time) checkable (but can be of exponential size with respect to a formula)
- Example: Resolution proofs
  - A resolution proof is a sequence $C_1, \ldots, C_n$ of clauses.
  - Every clause is either contained in the formula or derived from two earlier clauses via the resolution rule:

$$\frac{C \vee l \qquad \bar{l} \vee D}{C \vee D}$$

  - C_n is the empty clause (containing no literals).

What Is a Proof in SAT?

- In general, a proof is a string that certifies the unsatisfiability of a formula.
  - Proofs are efficiently (usually polynomial-time) checkable (but can be of exponential size with respect to a formula)
- Example: Resolution proofs
  - A resolution proof is a sequence $C_1, \ldots, C_n$ of clauses.
  - Every clause is either contained in the formula or derived from two earlier clauses via the resolution rule:

$$\frac{C \vee l \qquad \bar{l} \vee D}{C \vee D}$$

  - C_n is the empty clause (containing no literals).
  - There exists a resolution proof for every unsatisfiable formula.

Resolution Proofs

- Example: $F = (\bar{x} \vee \bar{y} \vee z) \wedge (\bar{z}) \wedge (x \vee \bar{y}) \wedge (\bar{u} \vee y) \wedge (u)$

Resolution Proofs

- Example: $F = (\bar{x} \vee \bar{y} \vee z) \wedge (\bar{z}) \wedge (x \vee \bar{y}) \wedge (\bar{u} \vee y) \wedge (u)$
- Resolution proof:
  $(\bar{x} \vee \bar{y} \vee z), (\bar{z}), (\bar{x} \vee \bar{y}), (x \vee \bar{y}), (\bar{y}), (\bar{u} \vee y), (\bar{u}), (u), \emptyset$

Resolution Proofs

- Example: $F = (\bar{x} \vee \bar{y} \vee z) \wedge (\bar{z}) \wedge (x \vee \bar{y}) \wedge (\bar{u} \vee y) \wedge (u)$
- Resolution proof:
  $(\bar{x} \vee \bar{y} \vee z), (\bar{z}), (\bar{x} \vee \bar{y}), (x \vee \bar{y}), (\bar{y}), (\bar{u} \vee y), (\bar{u}), (u), \emptyset$

$$\dfrac{\dfrac{\bar{u} \vee y \qquad \dfrac{\dfrac{\bar{x} \vee \bar{y} \vee z \qquad \bar{z}}{\bar{x} \vee \bar{y}} \qquad x \vee \bar{y}}{\bar{y}}}{\bar{u}} \qquad u}{\emptyset}$$

Resolution Proofs

- Example: $F = (\bar{x} \vee \bar{y} \vee z) \wedge (\bar{z}) \wedge (x \vee \bar{y}) \wedge (\bar{u} \vee y) \wedge (u)$
- Resolution proof:
  $(\bar{x} \vee \bar{y} \vee z), (\bar{z}), (\bar{x} \vee \bar{y}), (x \vee \bar{y}), (\bar{y}), (\bar{u} \vee y), (\bar{u}), (u), \emptyset$

$$\dfrac{\dfrac{\bar{u} \vee y \qquad \dfrac{\dfrac{\bar{x} \vee \bar{y} \vee z \qquad \bar{z}}{\bar{x} \vee \bar{y}} \qquad x \vee \bar{y}}{\bar{y}}}{\bar{u}} \qquad u}{\emptyset}$$

- Drawbacks of resolution:

Resolution Proofs

- Example: $F = (\bar{x} \lor \bar{y} \lor z) \land (\bar{z}) \land (x \lor \bar{y}) \land (\bar{u} \lor y) \land (u)$
- Resolution proof:
  $(\bar{x} \lor \bar{y} \lor z), (\bar{z}), (\bar{x} \lor \bar{y}), (x \lor \bar{y}), (\bar{y}), (\bar{u} \lor y), (\bar{u}), (u), \emptyset$

$$\dfrac{\dfrac{\bar{u} \lor y \qquad \dfrac{\dfrac{\bar{x} \lor \bar{y} \lor z \qquad \bar{z}}{\bar{x} \lor \bar{y}} \qquad x \lor \bar{y}}{\bar{y}}}{\bar{u}} \qquad u}{\emptyset}$$

- Drawbacks of resolution:
  - For many seemingly simple formulas, there are only resolution proofs of exponential size.

Resolution Proofs

- Example: $F = (\bar{x} \lor \bar{y} \lor z) \land (\bar{z}) \land (x \lor \bar{y}) \land (\bar{u} \lor y) \land (u)$
- Resolution proof:
  $(\bar{x} \lor \bar{y} \lor z), (\bar{z}), (\bar{x} \lor \bar{y}), (x \lor \bar{y}), (\bar{y}), (\bar{u} \lor y), (\bar{u}), (u), \emptyset$

$$\dfrac{\dfrac{\bar{u} \lor y \qquad \dfrac{\dfrac{\bar{x} \lor \bar{y} \lor z \qquad \bar{z}}{\bar{x} \lor \bar{y}} \qquad x \lor \bar{y}}{\bar{y}}}{\bar{u}} \qquad u}{\emptyset}$$

- Drawbacks of resolution:
  - For many seemingly simple formulas, there are only resolution proofs of exponential size.
  - State-of-the-art solving techniques are not succinctly expressible.

Properties of a Desirable Proof System for SAT

Properties of a Desirable Proof System for SAT

1. Succinctness: Proofs of unsatisfiability should be short strings that certify the unsatisfiability of formulas.

Properties of a Desirable Proof System for SAT

1. Succinctness: Proofs of unsatisfiability should be short strings that certify the unsatisfiability of formulas.
2. Efficient Checkability: It should be easy to verify that a proof is correct, i.e., that it certifies the unsatisfiability of a formula.

Properties of a Desirable Proof System for SAT

1. Succinctness: Proofs of unsatisfiability should be short strings that certify the unsatisfiability of formulas.
2. Efficient Checkability: It should be easy to verify that a proof is correct, i.e., that it certifies the unsatisfiability of a formula.
3. Practicability: SAT solvers should be able to produce proofs.

Properties of a Desirable Proof System for SAT

1. Succinctness: Proofs of unsatisfiability should be short strings that certify the unsatisfiability of formulas.
2. Efficient Checkability: It should be easy to verify that a proof is correct, i.e., that it certifies the unsatisfiability of a formula.
3. Practicability: SAT solvers should be able to produce proofs.
   - ➥ State-of-the-art techniques should be expressible in the system.

Properties of a Desirable Proof System for SAT

1. Succinctness: Proofs of unsatisfiability should be short strings that certify the unsatisfiability of formulas.
2. Efficient Checkability: It should be easy to verify that a proof is correct, i.e., that it certifies the unsatisfiability of a formula.
3. Practicability: SAT solvers should be able to produce proofs.
   - ➥ State-of-the-art techniques should be expressible in the system.
4. (Soundness and completeness.)

Traditional Proofs vs. Interference-Based Proofs

- In traditional proof systems, everything that is inferred, is implied by the premises.

$$\frac{C \lor I \qquad \bar{I} \lor D}{C \lor D}\ \text{(res)} \qquad\qquad \frac{A \qquad A \rightarrow B}{B}\ \text{(mp)}$$

Traditional Proofs vs. Interference-Based Proofs

- In traditional proof systems, everything that is inferred, is implied by the premises.

$$\frac{C \lor I \qquad \bar{I} \lor D}{C \lor D}\,(\text{res}) \qquad \frac{A \qquad A \rightarrow B}{B}\,(\text{mp})$$

➥ When inferring something, we reason about the presence of facts.

  - If certain premises are present, infer the conclusion.

Traditional Proofs vs. Interference-Based Proofs

- In traditional proof systems, everything that is inferred, is implied by the premises.

$$\frac{C \vee I \qquad \bar{I} \vee D}{C \vee D}\ \text{(res)} \qquad\qquad \frac{A \qquad A \rightarrow B}{B}\ \text{(mp)}$$

➥ When inferring something, we reason about the presence of facts.

  - If certain premises are present, infer the conclusion.

- Different approach: Allow not only implied conclusions.

Traditional Proofs vs. Interference-Based Proofs

- In traditional proof systems, everything that is inferred, is implied by the premises.

$$\frac{C \vee I \qquad \bar{I} \vee D}{C \vee D}\ (\text{res}) \qquad \frac{A \qquad A \rightarrow B}{B}\ (\text{mp})$$

➥ When inferring something, we reason about the presence of facts.

  - If certain premises are present, infer the conclusion.

- Different approach: Allow not only implied conclusions.
  - Require only that the addition of facts preserves satisfiability.

Traditional Proofs vs. Interference-Based Proofs

- In traditional proof systems, everything that is inferred, is implied by the premises.

$$\frac{C \vee l \qquad \bar{l} \vee D}{C \vee D}\ \text{(res)} \qquad \frac{A \qquad A \rightarrow B}{B}\ \text{(mp)}$$

➥ When inferring something, we reason about the presence of facts.

  - If certain premises are present, infer the conclusion.

- Different approach: Allow not only implied conclusions.
  - Require only that the addition of facts preserves satisfiability.
  - Reason also about the absence of facts.

Traditional Proofs vs. Interference-Based Proofs

- In traditional proof systems, everything that is inferred, is implied by the premises.

$$\frac{C \lor l \qquad \bar{l} \lor D}{C \lor D}\ (\text{res}) \qquad\qquad \frac{A \qquad A \to B}{B}\ (\text{mp})$$

➥ When inferring something, we reason about the presence of facts.

  - If certain premises are present, infer the conclusion.

- Different approach: Allow not only implied conclusions.
  - Require only that the addition of facts preserves satisfiability.
  - Reason also about the absence of facts.

  ➥ This leads to interference-based proof systems.

Interference-Based Proof Systems

- Interference-based proof systems generalize traditional proof systems.

Interference-Based Proof Systems

- Interference-based proof systems generalize traditional proof systems.
- An interference-based proof is a sequence of clauses.

Interference-Based Proof Systems

- Interference-based proof systems generalize traditional proof systems.
- An interference-based proof is a sequence of clauses.
  - Idea: The clauses are added to the initial formula step-by-step.

Interference-Based Proof Systems

- Interference-based proof systems generalize traditional proof systems.
- An interference-based proof is a sequence of clauses.
  - Idea: The clauses are added to the initial formula step-by-step.
  - Added clauses need not be implied, but their addition must preserve satisfiability:

Interference-Based Proof Systems

- Interference-based proof systems generalize traditional proof systems.
- An interference-based proof is a sequence of clauses.
  - Idea: The clauses are added to the initial formula step-by-step.
  - Added clauses need not be implied, but their addition must preserve satisfiability:

➥ If the formula is satisfiable, then the formula obtained by adding the clause is also satisfiable.

Interference-Based Proof Systems

- Interference-based proof systems generalize traditional proof systems.
- An interference-based proof is a sequence of clauses.
  - Idea: The clauses are added to the initial formula step-by-step.
  - Added clauses need not be implied, but their addition must preserve satisfiability:

➥ If the formula is satisfiable, then the formula obtained by adding the clause is also satisfiable.

➥ If the (unsatisfiable) empty clause, $\emptyset$, can be added, then the original formula must be unsatisfiable.

Interference-Based Proof Systems

- Interference-based proof systems generalize traditional proof systems.
- An interference-based proof is a sequence of clauses.
  - Idea: The clauses are added to the initial formula step-by-step.
  - Added clauses need not be implied, but their addition must preserve satisfiability:

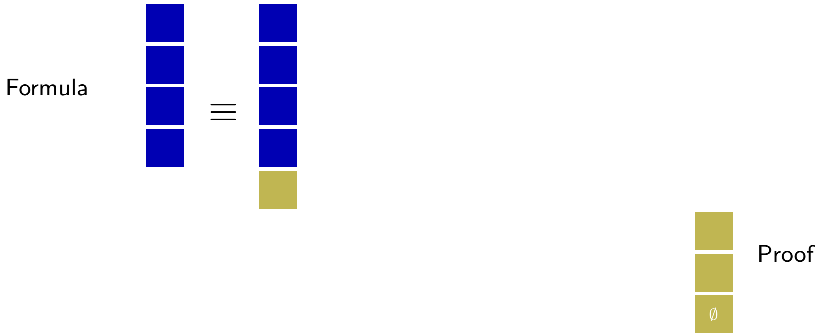
➥ If the formula is satisfiable, then the formula obtained by adding the clause is also satisfiable.

➥ If the (unsatisfiable) empty clause, $\emptyset$, can be added, then the original formula must be unsatisfiable.

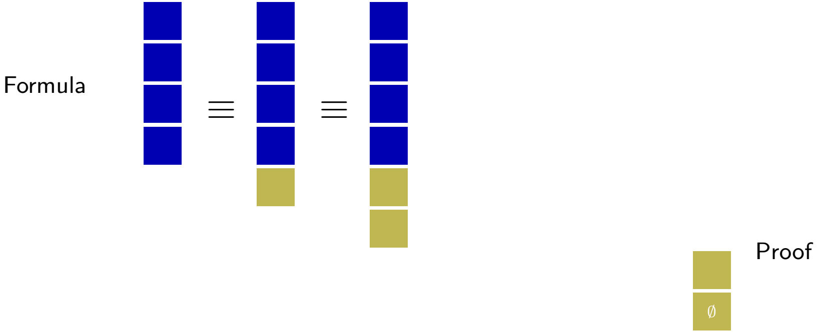
  - The empty clause is unsatisfiable because it has no literal that could be true.

Interference-Based Proofs

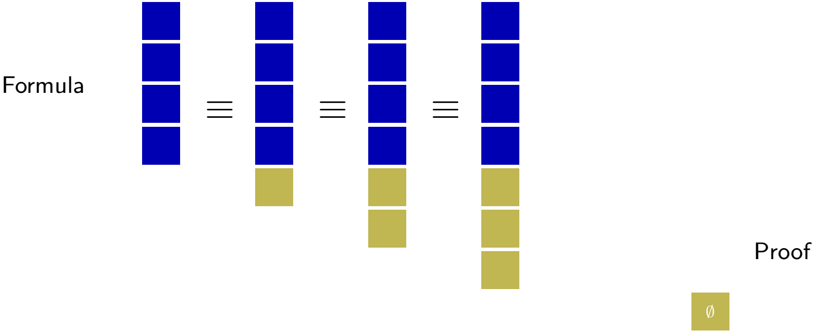
Interference-Based Proofs



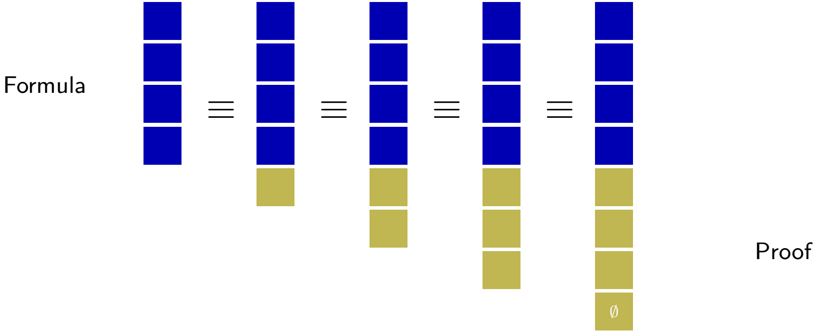
Interference-Based Proofs



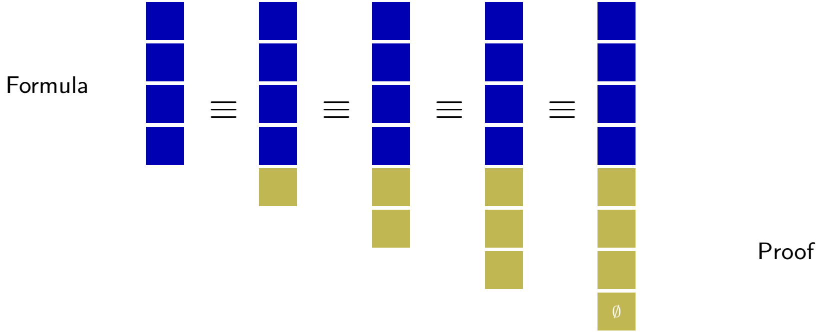
Interference-Based Proofs



Interference-Based Proofs



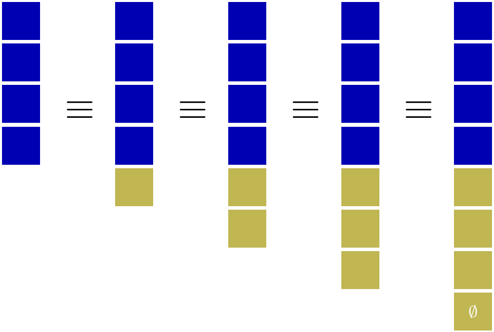
Interference-Based Proofs



- It should be efficiently checkable whether clause additions preserve satisfiability.

Interference-Based Proofs

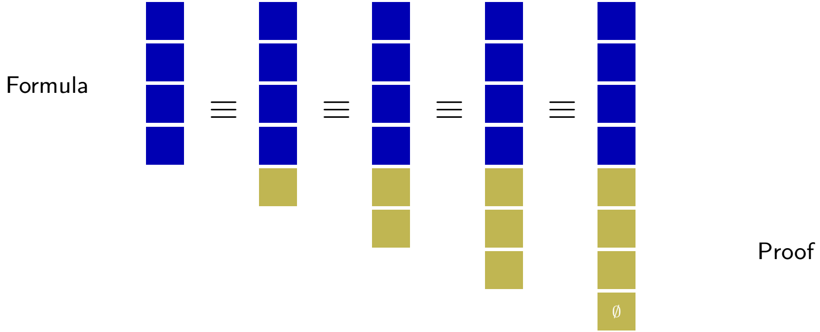
Formula



Proof

- It should be efficiently checkable whether clause additions preserve satisfiability.
- Clauses whose addition preserves satisfiability are called redundant.

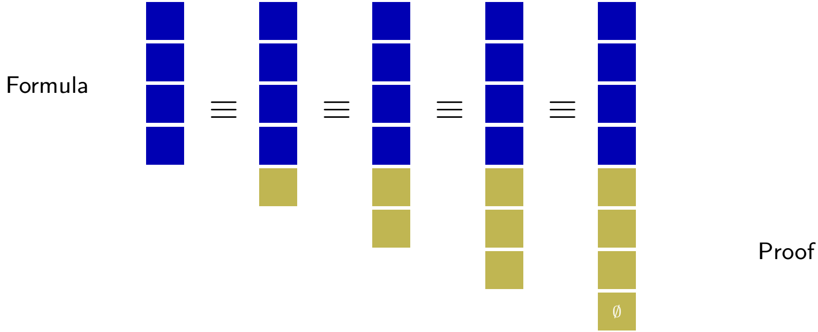
Interference-Based Proofs



- It should be efficiently checkable whether clause additions preserve satisfiability.
- Clauses whose addition preserves satisfiability are called redundant.

➥ Idea: Allow only the addition of clauses that fulfill an efficiently checkable redundancy criterion.

Interference-Based Proofs



- It should be efficiently checkable whether clause additions preserve satisfiability.
- Clauses whose addition preserves satisfiability are called redundant.

➥ Idea: Allow only the addition of clauses that fulfill an efficiently checkable redundancy criterion.

  - Example: Addition of resolution asymmetric tautologies (RATs).

DRAT: An Interference-Based Proof System

- Popular example of an interference-based proof system: DRAT

DRAT: An Interference-Based Proof System

- Popular example of an interference-based proof system: DRAT
- DRAT allows the addition of so-called resolution asymmetric tautologies (RATs) to a formula (whatever that means).

DRAT: An Interference-Based Proof System

- Popular example of an interference-based proof system: DRAT
- DRAT allows the addition of so-called resolution asymmetric tautologies (RATs) to a formula (whatever that means).
  - It can be efficiently checked if a clause is a RAT.

DRAT: An Interference-Based Proof System

- Popular example of an interference-based proof system: DRAT
- DRAT allows the addition of so-called resolution asymmetric tautologies (RATs) to a formula (whatever that means).
  - It can be efficiently checked if a clause is a RAT.
  - RATs are not necessarily implied by the formula.

DRAT: An Interference-Based Proof System

- Popular example of an interference-based proof system: DRAT
- DRAT allows the addition of so-called resolution asymmetric tautologies (RATs) to a formula (whatever that means).
  - It can be efficiently checked if a clause is a RAT.
  - RATs are not necessarily implied by the formula.
  - But RATs are redundant: their addition preserves satisfiability.

DRAT: An Interference-Based Proof System

- Popular example of an interference-based proof system: DRAT
- DRAT allows the addition of so-called resolution asymmetric tautologies (RATs) to a formula (whatever that means).
  - It can be efficiently checked if a clause is a RAT.
  - RATs are not necessarily implied by the formula.
  - But RATs are redundant: their addition preserves satisfiability.
  - A RAT check involves reasoning about the absence of facts.

DRAT: An Interference-Based Proof System

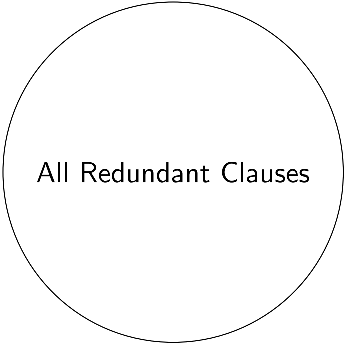
- Popular example of an interference-based proof system: DRAT
- DRAT allows the addition of so-called resolution asymmetric tautologies (RATs) to a formula (whatever that means).
  - It can be efficiently checked if a clause is a RAT.
  - RATs are not necessarily implied by the formula.
  - But RATs are redundant: their addition preserves satisfiability.
  - A RAT check involves reasoning about the absence of facts.
    - A clause is a RAT w.r.t. a formula if the formula contains no clause such that ...

DRAT: An Interference-Based Proof System

- Popular example of an interference-based proof system: DRAT
- DRAT allows the addition of so-called resolution asymmetric tautologies (RATs) to a formula (whatever that means).
  - It can be efficiently checked if a clause is a RAT.
  - RATs are not necessarily implied by the formula.
  - But RATs are redundant: their addition preserves satisfiability.
  - A RAT check involves reasoning about the absence of facts.
    - A clause is a RAT w.r.t. a formula if the formula contains no clause such that ...
- Are there more general types of redundant clauses than RATs?

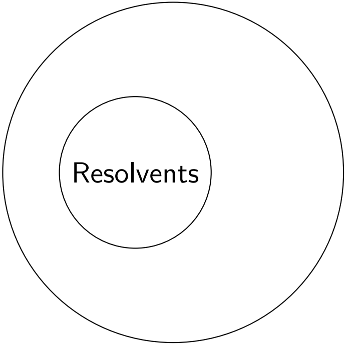
Redundant Clauses

- Strong proof systems allow the addition of many redundant clauses.



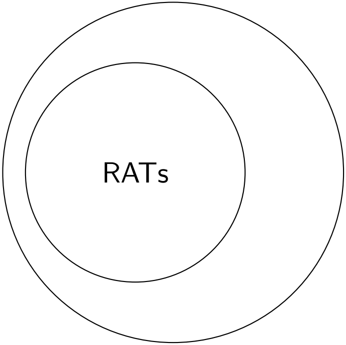
Redundant Clauses

- Strong proof systems allow the addition of many redundant clauses.



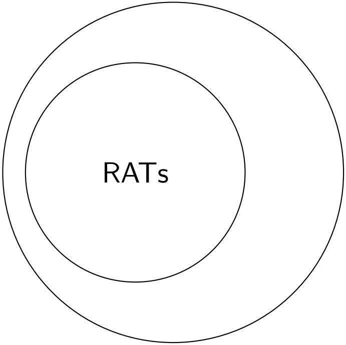
Redundant Clauses

- Strong proof systems allow the addition of many redundant clauses.



Redundant Clauses

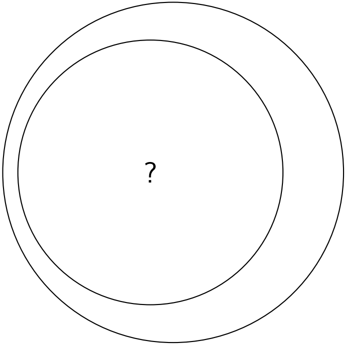
- Strong proof systems allow the addition of many redundant clauses.



- Are there stronger redundancy notions that are efficiently checkable?

Redundant Clauses

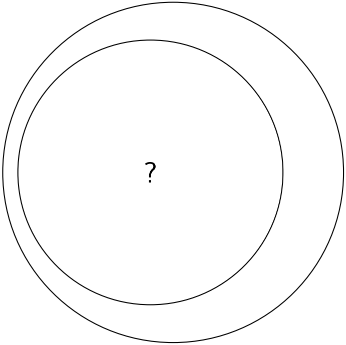
- Strong proof systems allow the addition of many redundant clauses.



- Are there stronger redundancy notions that are efficiently checkable?

Redundant Clauses

- Strong proof systems allow the addition of many redundant clauses.



- Are there stronger redundancy notions that are efficiently checkable?

➥ Short Proofs Without New Variables

Short Proofs Without New Variables: Main Contributions

- We introduced new clause-redundancy notions:
  - Propagation-redundant (PR) clauses
  - Set-propagation-redundant (SPR) clauses
  - Literal-propagation-redundant (LPR) clauses

Short Proofs Without New Variables: Main Contributions

- We introduced new clause-redundancy notions:
  - Propagation-redundant (PR) clauses
  - Set-propagation-redundant (SPR) clauses
  - Literal-propagation-redundant (LPR) clauses
- LPR clauses coincide with RAT.

Short Proofs Without New Variables: Main Contributions

- We introduced new clause-redundancy notions:
  - Propagation-redundant (PR) clauses
  - Set-propagation-redundant (SPR) clauses
  - Literal-propagation-redundant (LPR) clauses
- LPR clauses coincide with RAT.
- SPR clauses strictly generalize RATs.

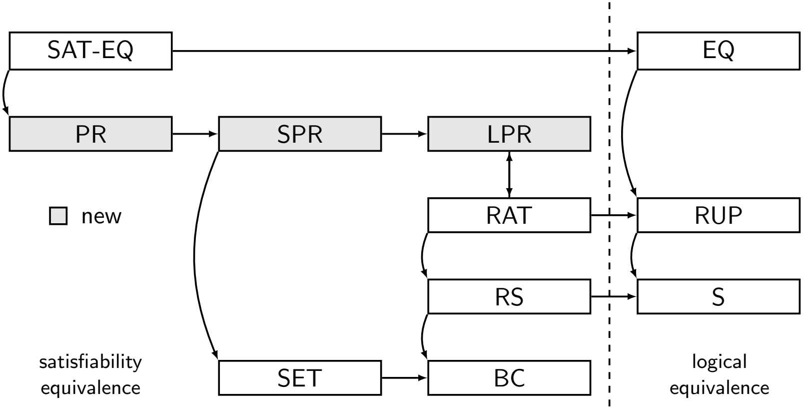
Short Proofs Without New Variables: Main Contributions

- We introduced new clause-redundancy notions:
  - Propagation-redundant (PR) clauses
  - Set-propagation-redundant (SPR) clauses
  - Literal-propagation-redundant (LPR) clauses
- LPR clauses coincide with RAT.
- SPR clauses strictly generalize RATs.
- PR clauses strictly generalize SPR clauses.

Short Proofs Without New Variables: Main Contributions

- We introduced new clause-redundancy notions:
  - Propagation-redundant (PR) clauses
  - Set-propagation-redundant (SPR) clauses
  - Literal-propagation-redundant (LPR) clauses
- LPR clauses coincide with RAT.
- SPR clauses strictly generalize RATs.
- PR clauses strictly generalize SPR clauses.
- The redundancy notions provide the basis for new proof systems.

New Landscape of Redundancy Notions



Stronger Proof Systems: What Are They Good For?

- The new proof systems can give short proofs of formulas that are considered hard.

Stronger Proof Systems: What Are They Good For?

- The new proof systems can give short proofs of formulas that are considered hard.
- We have short SPR and PR proofs for the well-known pigeon hole formulas.

Stronger Proof Systems: What Are They Good For?

- The new proof systems can give short proofs of formulas that are considered hard.
- We have short SPR and PR proofs for the well-known pigeon hole formulas.
  - Pigeon hole formulas have only exponential-size resolution proofs.

Stronger Proof Systems: What Are They Good For?

- The new proof systems can give short proofs of formulas that are considered hard.
- We have short SPR and PR proofs for the well-known pigeon hole formulas.
  - Pigeon hole formulas have only exponential-size resolution proofs.
  - If the addition of new variables via definitions is allowed, there are polynomial-size proofs.

Stronger Proof Systems: What Are They Good For?

- The new proof systems can give short proofs of formulas that are considered hard.
- We have short SPR and PR proofs for the well-known pigeon hole formulas.
  - Pigeon hole formulas have only exponential-size resolution proofs.
  - If the addition of new variables via definitions is allowed, there are polynomial-size proofs.
    - So-called extended resolution proofs.

Stronger Proof Systems: What Are They Good For?

- The new proof systems can give short proofs of formulas that are considered hard.
- We have short SPR and PR proofs for the well-known pigeon hole formulas.
  - Pigeon hole formulas have only exponential-size resolution proofs.
  - If the addition of new variables via definitions is allowed, there are polynomial-size proofs.
    - So-called extended resolution proofs.
- Our proofs do not require new variables.
  - ➥ Search space of possible clauses is finite.

Stronger Proof Systems: What Are They Good For?

- The new proof systems can give short proofs of formulas that are considered hard.
- We have short SPR and PR proofs for the well-known pigeon hole formulas.
  - Pigeon hole formulas have only exponential-size resolution proofs.
  - If the addition of new variables via definitions is allowed, there are polynomial-size proofs.
    - So-called extended resolution proofs.
- Our proofs do not require new variables.
  - ➥ Search space of possible clauses is finite.
  - ➥ Makes search for such clauses easier.

Short Proofs Without New Variables: Conclusion

- We introduced new redundancy notions for SAT.

Short Proofs Without New Variables: Conclusion

- We introduced new redundancy notions for SAT.
- The redundancy notions strictly generalize RAT.

Short Proofs Without New Variables: Conclusion

- We introduced new redundancy notions for SAT.
- The redundancy notions strictly generalize RAT.
- Proof systems based on these redundancy notions are strong.

Short Proofs Without New Variables: Conclusion

- We introduced new redundancy notions for SAT.
- The redundancy notions strictly generalize RAT.
- Proof systems based on these redundancy notions are strong.
  - They allow for short proofs without new variables.

Short Proofs Without New Variables: Conclusion

- We introduced new redundancy notions for SAT.
- The redundancy notions strictly generalize RAT.
- Proof systems based on these redundancy notions are strong.
  - They allow for short proofs without new variables.

Short Proofs Without New Variables: Conclusion

- We introduced new redundancy notions for SAT.
- The redundancy notions strictly generalize RAT.
- Proof systems based on these redundancy notions are strong.
  - They allow for short proofs without new variables.
- Proofs for the pigeon hole formulas are hand-crafted.

Short Proofs Without New Variables: Conclusion

- We introduced new redundancy notions for SAT.
- The redundancy notions strictly generalize RAT.
- Proof systems based on these redundancy notions are strong.
  - They allow for short proofs without new variables.
- Proofs for the pigeon hole formulas are hand-crafted.

➥ Open problem: Automatically generate such short proofs.

What the Reviewers Say

Reviewer 1:
"I find the topic interesting and I believe the authors did a great job when writing the paper (. . .) I believe this paper contains solid work that should be presented at CADE."

What the Reviewers Say

Reviewer 1:
"*I find the topic interesting and I believe the authors did a great job when writing the paper* (. . .) *I believe this paper contains solid work that should be presented at CADE.*"

Reviewer 2:
"*The paper is very well written, is easy to follow and a pleasure to read. The authors address an important problem.*"

What the Reviewers Say

Reviewer 1:
"I find the topic interesting and I believe the authors did a great job when writing the paper (...) I believe this paper contains solid work that should be presented at CADE."

Reviewer 2:
"The paper is very well written, is easy to follow and a pleasure to read. The authors address an important problem."

Reviewer 3:
"The presented proof system is novel and powerful, the results in the paper are interesting, and the paper fits the scope of CADE."

What Jayadev Misra Says

Jayadev Misra:
"*Good work is more important than good reviews.*"

A Little Blocked Literal Goes a Long Way: Overview

- Deals with proofs for quantified Boolean formulas (QBFs).

A Little Blocked Literal Goes a Long Way: Overview

- Deals with proofs for quantified Boolean formulas (QBFs).
- Short overview on QBF and corresponding proof systems.

A Little Blocked Literal Goes a Long Way: Overview

- Deals with proofs for quantified Boolean formulas (QBFs).
- Short overview on QBF and corresponding proof systems.
- We show that QRAT (the QBF generalization of DRAT) can polynomially simulate long-distance resolution.

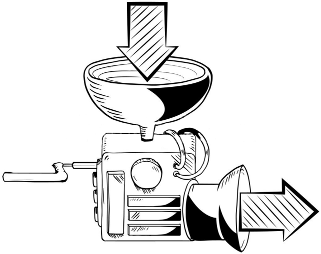
A Little Blocked Literal Goes a Long Way: Overview

- Deals with proofs for quantified Boolean formulas (QBFs).
- Short overview on QBF and corresponding proof systems.
- We show that QRAT (the QBF generalization of DRAT) can polynomially simulate long-distance resolution.
- We have an implementation and evaluation of the simulation.

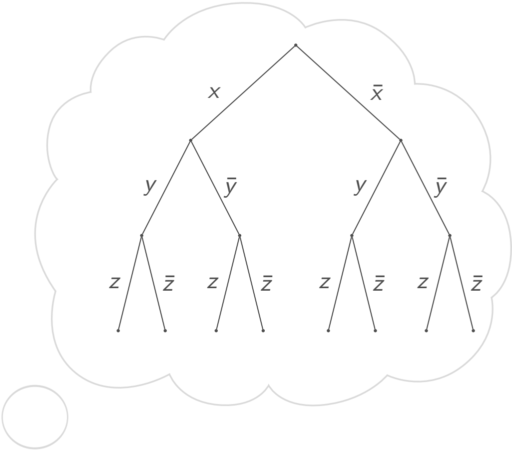
Satisfiability of Quantified Boolean Formulas (QSAT)

"For every truth value of x,
does there exist a truth value of y,
such that ..."

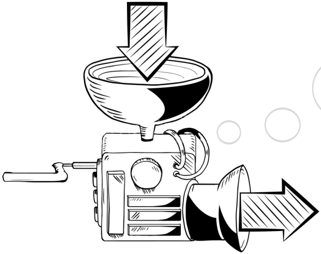
$$\forall x \exists y \forall z \ (x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$$



Satisfiability of Quantified Boolean Formulas (QSAT)

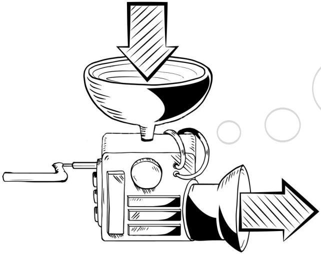


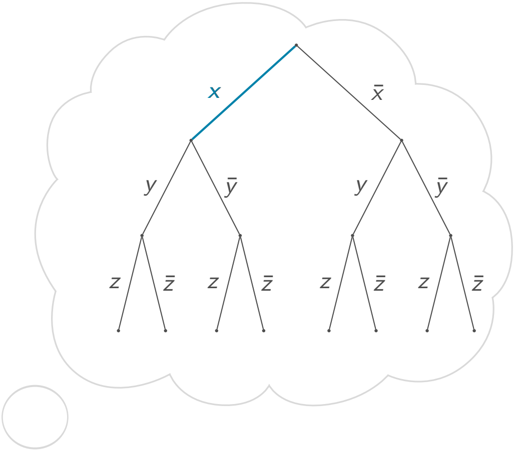
$\forall x \exists y \forall z \; (x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$



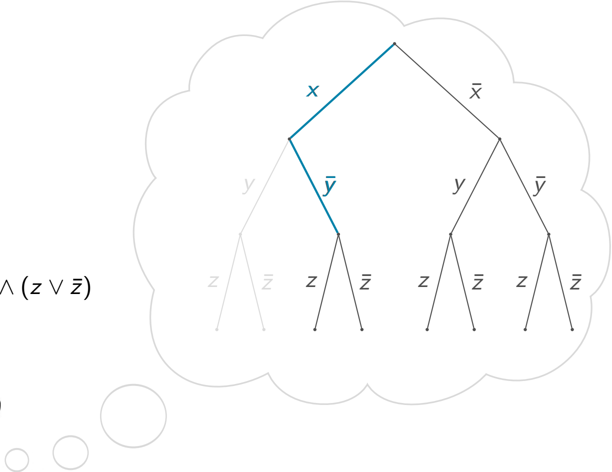
Satisfiability of Quantified Boolean Formulas (QSAT)

$\forall x \exists y \forall z \ (x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$

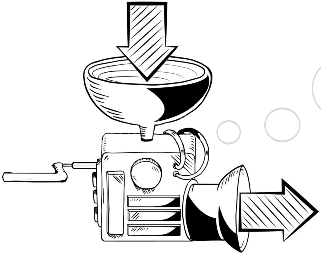




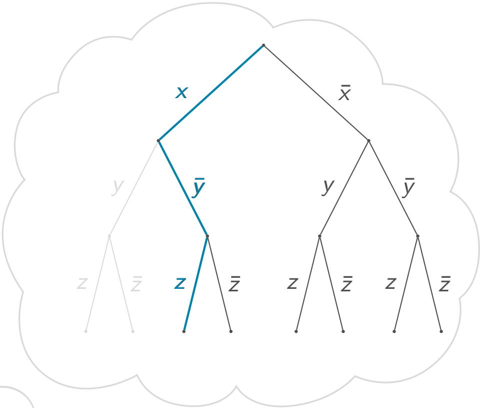
Satisfiability of Quantified Boolean Formulas (QSAT)



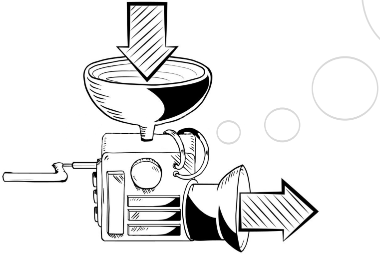
$\forall x \exists y \forall z \ (x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$



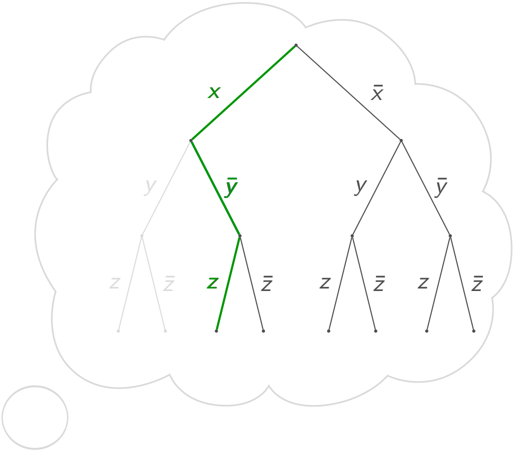
Satisfiability of Quantified Boolean Formulas (QSAT)



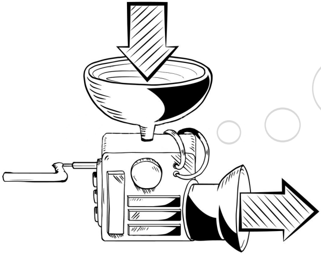
$\forall x \exists y \forall z \; (x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$



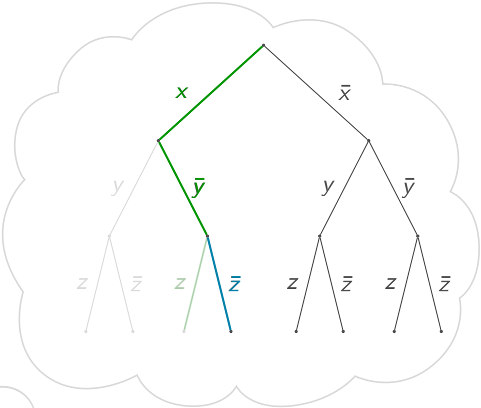
Satisfiability of Quantified Boolean Formulas (QSAT)



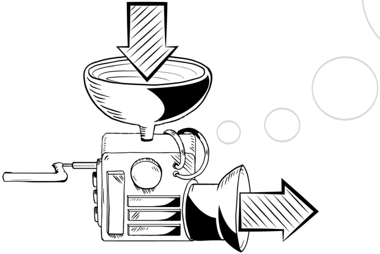
$\forall x \exists y \forall z\ (x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$



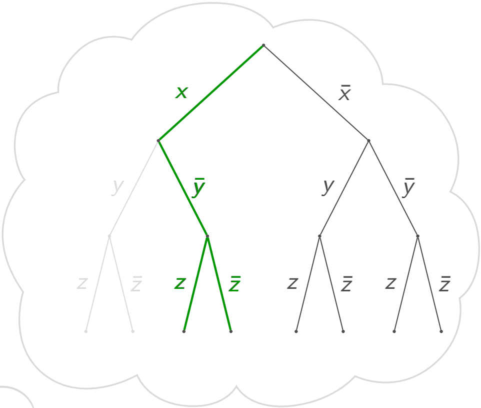
Satisfiability of Quantified Boolean Formulas (QSAT)



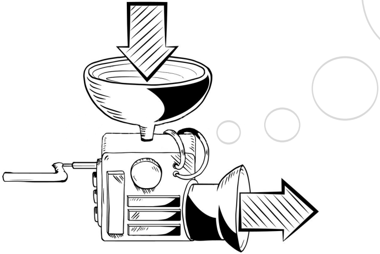
$\forall x \exists y \forall z \; (x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$



Satisfiability of Quantified Boolean Formulas (QSAT)

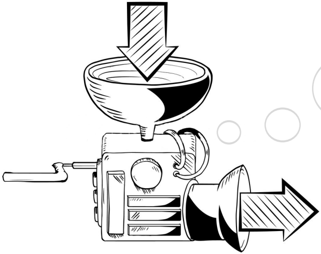


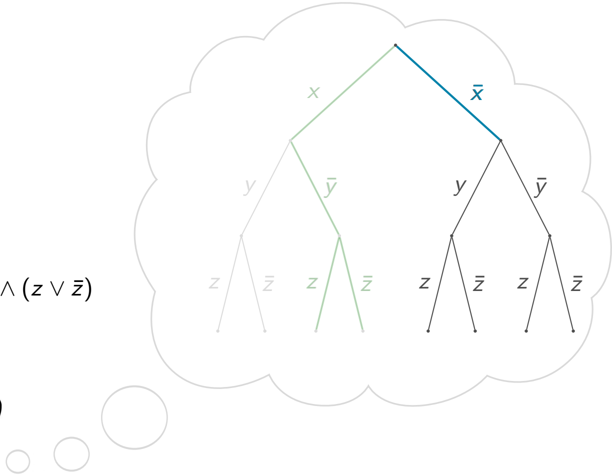
$\forall x \exists y \forall z\ (x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$



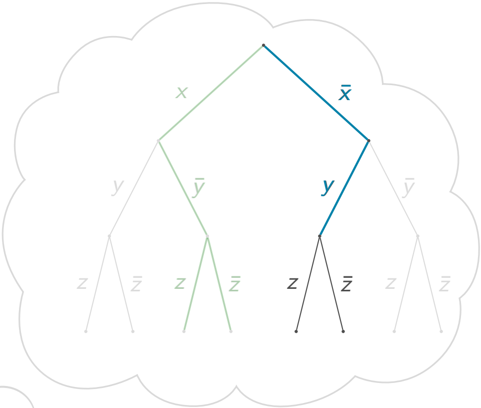
Satisfiability of Quantified Boolean Formulas (QSAT)

$\forall x \exists y \forall z \ (x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$

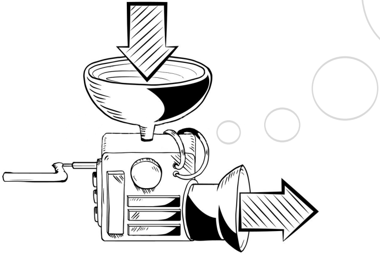




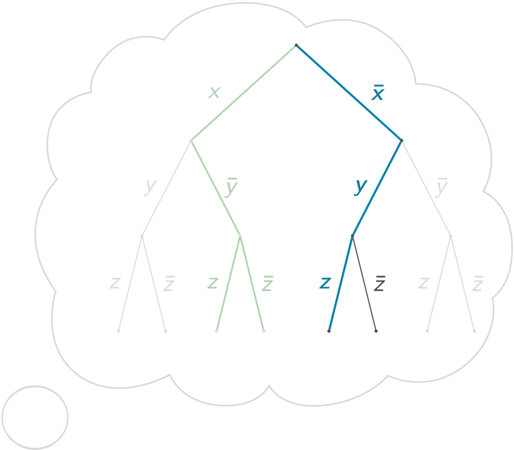
Satisfiability of Quantified Boolean Formulas (QSAT)



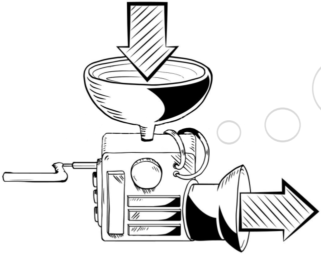
$\forall x \exists y \forall z \; (x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$



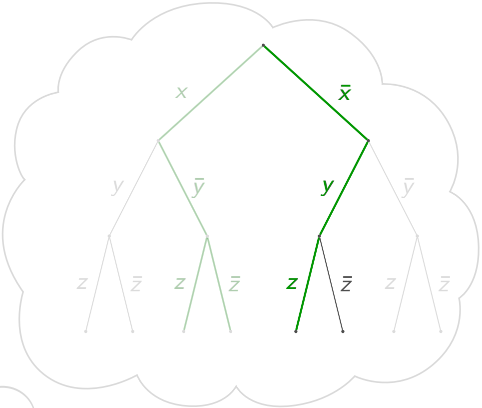
Satisfiability of Quantified Boolean Formulas (QSAT)



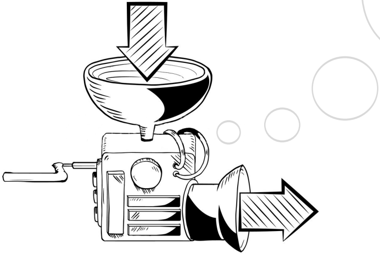
$\forall x \exists y \forall z \; (x \lor y) \land (\bar{x} \lor \bar{y}) \land (z \lor \bar{z})$



Satisfiability of Quantified Boolean Formulas (QSAT)

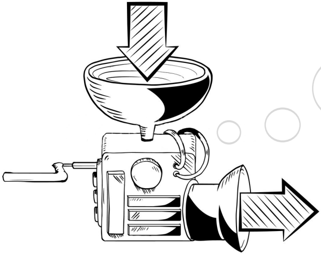


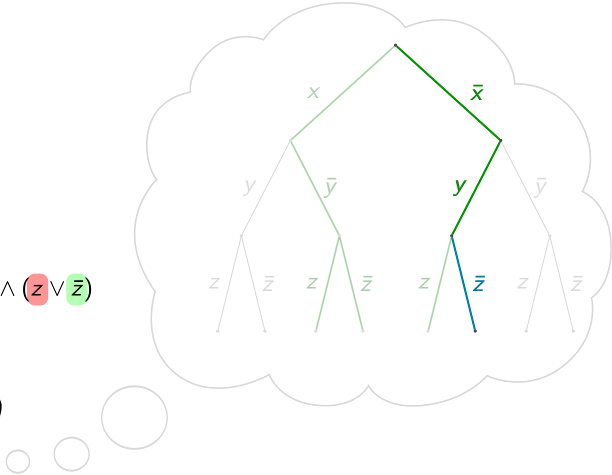
$\forall x \exists y \forall z \ (x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$



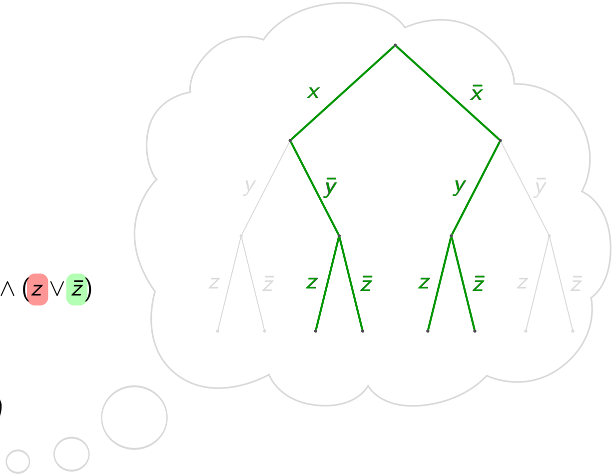
Satisfiability of Quantified Boolean Formulas (QSAT)

$\forall x \exists y \forall z\ (x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$

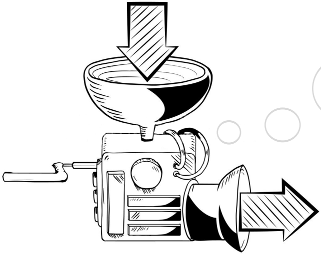




Satisfiability of Quantified Boolean Formulas (QSAT)



$\forall x \exists y \forall z \; (x \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (z \vee \bar{z})$



Satisfiable

Proof Systems for QBF: LQ-Res vs. QRAT

- Proof systems for QBF are similar to proof systems for SAT.

Proof Systems for QBF: LQ-Res vs. QRAT

- Proof systems for QBF are similar to proof systems for SAT.
- There are several resolution systems.

Proof Systems for QBF: LQ-Res vs. QRAT

- Proof systems for QBF are similar to proof systems for SAT.
- There are several resolution systems.
- Most popular system: long-distance resolution (LQ-Res)

Proof Systems for QBF: LQ-Res vs. QRAT

- Proof systems for QBF are similar to proof systems for SAT.
- There are several resolution systems.
- Most popular system: long-distance resolution (LQ-Res)
  - Allows for short proofs both in theory and in practical solving.

Proof Systems for QBF: LQ-Res vs. QRAT

- Proof systems for QBF are similar to proof systems for SAT.
- There are several resolution systems.
- Most popular system: long-distance resolution (LQ-Res)
  - Allows for short proofs both in theory and in practical solving.
- Other approach: QRAT (interference-based!)
  - QRAT is the QBF generalization of RAT.

Proof Systems for QBF: LQ-Res vs. QRAT

- Proof systems for QBF are similar to proof systems for SAT.
- There are several resolution systems.
- Most popular system: long-distance resolution (LQ-Res)
  - Allows for short proofs both in theory and in practical solving.
- Other approach: QRAT (interference-based!)
  - QRAT is the QBF generalization of RAT.
  - Perfect for certifying the correctness of the preprocessing.

Proof Systems for QBF: LQ-Res vs. QRAT

- Proof systems for QBF are similar to proof systems for SAT.
- There are several resolution systems.
- Most popular system: long-distance resolution (LQ-Res)
  - Allows for short proofs both in theory and in practical solving.
- Other approach: QRAT (interference-based!)
  - QRAT is the QBF generalization of RAT.
  - Perfect for certifying the correctness of the preprocessing.
- It was unclear how LQ-Res and QRAT are related.

Proof Systems for QBF: LQ-Res vs. QRAT

- Proof systems for QBF are similar to proof systems for SAT.
- There are several resolution systems.
- Most popular system: long-distance resolution (LQ-Res)
  - Allows for short proofs both in theory and in practical solving.
- Other approach: QRAT (interference-based!)
  - QRAT is the QBF generalization of RAT.
  - Perfect for certifying the correctness of the preprocessing.
- It was unclear how LQ-Res and QRAT are related.
  - If there is a short LQ-Res proof of a QBF, is there also a short QRAT proof?

Proof Systems for QBF: LQ-Res vs. QRAT

- Proof systems for QBF are similar to proof systems for SAT.
- There are several resolution systems.
- Most popular system: long-distance resolution (LQ-Res)
  - Allows for short proofs both in theory and in practical solving.
- Other approach: QRAT (interference-based!)
  - QRAT is the QBF generalization of RAT.
  - Perfect for certifying the correctness of the preprocessing.
- It was unclear how LQ-Res and QRAT are related.
  - If there is a short LQ-Res proof of a QBF, is there also a short QRAT proof?
  - Short = polynomial with respect to the size of the formula.

Proof Systems for QBF: LQ-Res vs. QRAT

- Proof systems for QBF are similar to proof systems for SAT.
- There are several resolution systems.
- Most popular system: long-distance resolution (LQ-Res)
  - Allows for short proofs both in theory and in practical solving.
- Other approach: QRAT (interference-based!)
  - QRAT is the QBF generalization of RAT.
  - Perfect for certifying the correctness of the preprocessing.
- It was unclear how LQ-Res and QRAT are related.
  - If there is a short LQ-Res proof of a QBF, is there also a short QRAT proof?
  - Short = polynomial with respect to the size of the formula.
  - Our answer: Yes!

Simulating LQ-Res With QRAT

- How to show that there is a short QRAT proof for every short LQ-Res proof?

Simulating LQ-Res With QRAT

- How to show that there is a short QRAT proof for every short LQ-Res proof?

➥ Answer: With a simulation procedure.

Simulating LQ-Res With QRAT

- How to show that there is a short QRAT proof for every short LQ-Res proof?

➥ Answer: With a simulation procedure.

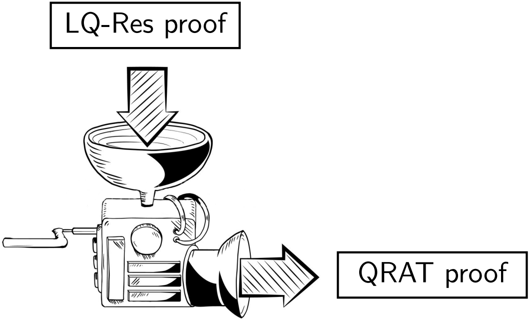
  - Takes as input an LQ-Res proof and transforms it into a short QRAT proof.

Simulating LQ-Res With QRAT

- How to show that there is a short QRAT proof for every short LQ-Res proof?

➥ Answer: With a simulation procedure.

  - Takes as input an LQ-Res proof and transforms it into a short QRAT proof.



Simulation Procedure: Results

- Our simulation procedure produces a QRAT proof with at most a quadratic blow-up in size.

Simulation Procedure: Results

- Our simulation procedure produces a QRAT proof with at most a quadratic blow-up in size.
- We implemented the procedure, the tool is called `ld2qrat`.

Simulation Procedure: Results

- Our simulation procedure produces a QRAT proof with at most a quadratic blow-up in size.
- We implemented the procedure, the tool is called `ld2qrat`.
  - Takes a long-distance proof in the so-called QPR format.
  - Outputs a QRAT proof.

Simulation Procedure: Results

- Our simulation procedure produces a QRAT proof with at most a quadratic blow-up in size.
- We implemented the procedure, the tool is called `ld2qrat`.
  - Takes a long-distance proof in the so-called QPR format.
  - Outputs a QRAT proof.
  - Several optimizations to reduce proof size.

Simulation Procedure: Results

- Our simulation procedure produces a QRAT proof with at most a quadratic blow-up in size.
- We implemented the procedure, the tool is called `ld2qrat`.
  - Takes a long-distance proof in the so-called QPR format.
  - Outputs a QRAT proof.
  - Several optimizations to reduce proof size.
  - Resulting proofs are reasonably short.

Simulation Procedure: Results

- Our simulation procedure produces a QRAT proof with at most a quadratic blow-up in size.
- We implemented the procedure, the tool is called `ld2qrat`.
  - Takes a long-distance proof in the so-called QPR format.
  - Outputs a QRAT proof.
  - Several optimizations to reduce proof size.
  - Resulting proofs are reasonably short.
- With the tool it is now possible to merge a QRAT proof of a preprocessor with a long-distance proof of a search-based solver.

Simulation Procedure: Results

- Our simulation also gave insight for constructing short QRAT proofs by hand.

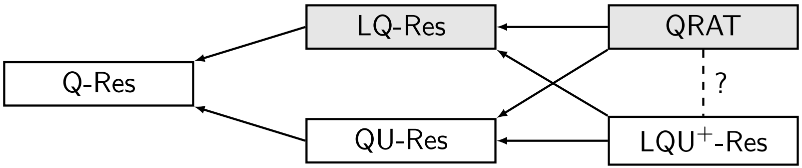
Simulation Procedure: Results

- Our simulation also gave insight for constructing short QRAT proofs by hand.
  - Formulas well-known for having short LQ-Res proofs but being hard for other proof systems: Kleine Büning formulas

Simulation Procedure: Results

- Our simulation also gave insight for constructing short QRAT proofs by hand.
  - Formulas well-known for having short LQ-Res proofs but being hard for other proof systems: Kleine Büning formulas
  - We have hand-crafted QRAT proofs of these formulas that are shorter than the LQ-Res proofs.

New Proof-Complexity Landscape for QBF



- Open question: Can QRAT also simulate LQU$^+$-Res, a system that is stronger than LQ-Res?

A Little Blocked Literal . . . : Conclusion

- We shed light on the relationship between LQ-Res and QRAT

A Little Blocked Literal . . . : Conclusion

- We shed light on the relationship between LQ-Res and QRAT
  - LQ-Res is a popular system for QBF solving.

A Little Blocked Literal . . . : Conclusion

- We shed light on the relationship between LQ-Res and QRAT
  - LQ-Res is a popular system for QBF solving.
  - QRAT is the best system for QBF preprocessing.

A Little Blocked Literal . . . : Conclusion

- We shed light on the relationship between LQ-Res and QRAT
    - LQ-Res is a popular system for QBF solving.
    - QRAT is the best system for QBF preprocessing.
- QRAT turns out to be stronger than LQ-Res.

A Little Blocked Literal . . . : Conclusion

- We shed light on the relationship between LQ-Res and QRAT
  - LQ-Res is a popular system for QBF solving.
  - QRAT is the best system for QBF preprocessing.
- QRAT turns out to be stronger than LQ-Res.
- Our new tool allows to transform LQ-Res proofs into QRAT proofs.

But I did not spend my whole time writing papers. (Fortunately.)

Found A Fantastic Collaborator/Supervisor

Found A Fantastic Collaborator/Supervisor/Friend

Had Also a Lot of Fun With His Husband

Lived Together With Magnificent Roommates

Had a Great Time With Lindy and Devon

And Last But Not Least: Met a Cool Group!