Lower bounds for set-blocked clauses proofs

Emre Yolcu

Computer Science Department
Carnegie Mellon University

eyolcu@cs.cmu.edu

Resolution

Refutes a propositional formula in conjunctive normal form (i.e., a set of clauses) by using the single rule

$$\frac{A \vee x \qquad B \vee \overline{x}}{A \vee B}$$

to derive the empty clause, which is trivially false.

Resolution

Refutes a propositional formula in conjunctive normal form (i.e., a set of clauses) by using the single rule

$$\frac{A \vee x \qquad B \vee \overline{x}}{A \vee B}$$

to derive the empty clause, which is trivially false.

Throughout this talk, “proof” means “refutation”:

$$\text{proof of unsatisfiability} \quad \equiv \quad \text{refutation of satisfiability}$$

Example: resolution proof

$$\Gamma = (\overline{x} \vee \overline{z}) \wedge (\overline{y} \vee z) \wedge (x \vee y \vee \overline{z}) \wedge (x \vee \overline{y}) \wedge (y \vee z)$$

Example: resolution proof

$$\Gamma = (\overline{x} \lor \overline{z}) \land (\overline{y} \lor z) \land (x \lor y \lor \overline{z}) \land (x \lor \overline{y}) \land (y \lor z)$$

Tree-like:

$$\dfrac{\dfrac{x \lor \overline{y} \qquad \dfrac{\dfrac{\overline{y} \lor z \qquad y \lor z}{z} \qquad x \lor y \lor \overline{z}}{x \lor y}}{x} \qquad \dfrac{\dfrac{\overline{y} \lor z \qquad y \lor z}{z} \qquad \overline{x} \lor \overline{z}}{\overline{x}}}{\bot}$$

Example: resolution proof

$$\Gamma = (\overline{x} \vee \overline{z}) \wedge (\overline{y} \vee z) \wedge (x \vee y \vee \overline{z}) \wedge (x \vee \overline{y}) \wedge (y \vee z)$$

Tree-like:

$$\dfrac{\dfrac{x \vee \overline{y} \qquad \dfrac{\dfrac{\overline{y} \vee z \qquad y \vee z}{z} \qquad x \vee y \vee \overline{z}}{x \vee y}}{x} \qquad \dfrac{\dfrac{\overline{y} \vee z \qquad y \vee z}{z} \qquad \overline{x} \vee \overline{z}}{\overline{x}}}{\bot}$$

Sequence-like:

$$\overline{x} \vee \overline{z},\ \overline{y} \vee z,\ x \vee y \vee \overline{z},\ x \vee \overline{y},\ y \vee z,\ z,\ x \vee y,\ x,\ \overline{x},\ \bot$$

Motivation: proof complexity

As an attempt to understand the coNP vs. NP problem, proof complexity studies propositional proof systems.

Motivation: proof complexity

As an attempt to understand the coNP vs. NP problem, proof complexity studies propositional proof systems.

Mainly concerned with the quantity

$$\mathsf{s}_P(\Gamma) := \text{“size of the smallest } P\text{-proof of } \Gamma\text{”}.$$

Motivation: proof complexity

As an attempt to understand the coNP vs. NP problem, proof complexity studies propositional proof systems.

Mainly concerned with the quantity

$$\mathsf{s}_P(\Gamma) := \text{“size of the smallest } P\text{-proof of } \Gamma\text{”}.$$

Also concerned with comparing proof systems:

- *P simulates Q* if every Q-proof can be converted into an at most polynomially larger P-proof of the same formula.
- *P separates from Q* if there is an infinite sequence of formulas admitting poly-size P-proofs while requiring exp-size Q-proofs.

Motivation: SAT solvers

When a SAT solver claims unsatisfiability, it emits a proof.
Modern SAT solvers essentially* search for resolution proofs.

Motivation: SAT solvers

When a SAT solver claims unsatisfiability, it emits a proof.

Modern SAT solvers essentially* search for resolution proofs.

Several recent solvers apply preprocessing that is not expressible in resolution.

Motivation: SAT solvers

When a SAT solver claims unsatisfiability, it emits a proof.
Modern SAT solvers essentially* search for resolution proofs.

Several recent solvers apply preprocessing that is not expressible in resolution.

Resolution is rather weak, so there has been a push for using stronger systems as the basis for proof search.

Motivation: SAT solvers

When a SAT solver claims unsatisfiability, it emits a proof.
Modern SAT solvers essentially* search for resolution proofs.

*Several recent solvers apply preprocessing that is not expressible in resolution.

Resolution is rather weak, so there has been a push for using stronger systems as the basis for proof search.

It is relatively easy to search for resolution proofs because proof lines are extremely simple, which enables unit propagation.

Motivation: SAT solvers

When a SAT solver claims unsatisfiability, it emits a proof.
Modern SAT solvers essentially* search for resolution proofs.

Several recent solvers apply preprocessing that is not expressible in resolution.

Resolution is rather weak, so there has been a push for using stronger systems as the basis for proof search.

It is relatively easy to search for resolution proofs because proof lines are extremely simple, which enables unit propagation.

In short: We want stronger proof systems with simple proof lines.
They should not be "too strong"; otherwise, proof search is difficult.

“Without loss of generality”

“Without loss of generality”

You have probably written something like the following somewhere:

> “WLOG, assume that $x \geq y$.”

“Without loss of generality”

You have probably written something like the following somewhere:

> “WLOG, assume that $x \geq y$.”
>
> “WLOG, assume that the vertex is colored red.”

“Without loss of generality”

You have probably written something like the following somewhere:

“WLOG, assume that $x \geq y$.”

“WLOG, assume that the vertex is colored red.”

“WLOG, assume that the random variable has mean zero.”

“Without loss of generality”

You have probably written something like the following somewhere:

“WLOG, assume that $x \geq y$.”

“WLOG, assume that the vertex is colored red.”

“WLOG, assume that the random variable has mean zero.”

$$\vdots$$

"Without loss of generality"

You have probably written something like the following somewhere:

> "WLOG, assume that $x \geq y$."
>
> "WLOG, assume that the vertex is colored red."
>
> "WLOG, assume that the random variable has mean zero."

$$\vdots$$

A major weakness of all of the "weak" proof systems is their inability to formalize the use of such phrases.

"Without loss of generality"

You have probably written something like the following somewhere:

> "WLOG, assume that $x \geq y$."
>
> "WLOG, assume that the vertex is colored red."
>
> "WLOG, assume that the random variable has mean zero."

$$\vdots$$

A major weakness of all of the "weak" proof systems is their inability to formalize the use of such phrases.

Extended resolution is a system that simulates this ability. It consists of the resolution rule + a rule to define new variables:

$x \leftrightarrow p \wedge q$, where p, q are arbitrary literals and x is "fresh"

Redundancy

Definition

A clause C is *redundant* with respect to a formula Γ if

$$\Gamma \text{ and } \Gamma \wedge C \text{ are equisatisfiable.}$$

Redundancy

Definition

A clause C is *redundant* with respect to a formula Γ if

$$\Gamma \text{ and } \Gamma \wedge C \text{ are equisatisfiable.}$$

Weaker than logical implication: does not require $\Gamma \models C$.

Semantic property: cannot necessarily be checked efficiently.

Redundancy

Definition

A clause C is *redundant* with respect to a formula Γ if

$$\Gamma \text{ and } \Gamma \wedge C \text{ are equisatisfiable.}$$

Weaker than logical implication: does not require $\Gamma \models C$.

Semantic property: cannot necessarily be checked efficiently.

Proofs must be verifiable in polynomial time, so we work with restricted versions of redundancy given by syntactic conditions.

Example: blocked clauses

Definition

A clause $C = x \vee C'$ is *blocked* for x with respect to a formula Γ if, for every clause D of the form $\overline{x} \vee D'$ in Γ,

$C' \vee D'$ contains a pair of complementary literals.

Example: blocked clauses

Definition

A clause $C = x \vee C'$ is *blocked* for x with respect to a formula Γ if, for every clause D of the form $\overline{x} \vee D'$ in Γ,

$$C' \vee D' \text{ contains a pair of complementary literals.}$$

Resolution rule + the ability to add blocked clauses results in the proof system called *blocked clauses*.

Example: blocked clauses

Definition

A clause $C = x \vee C'$ is *blocked* for x with respect to a formula Γ if, for every clause D of the form $\overline{x} \vee D'$ in Γ,

$$C' \vee D' \text{ contains a pair of complementary literals.}$$

Resolution rule + the ability to add blocked clauses results in the proof system called *blocked clauses*.

This work: Lower bounds and separations for proof systems that are based on different generalizations* of blocked clauses.

Example: blocked clauses

Definition

A clause $C = x \vee C'$ is *blocked* for x with respect to a formula Γ if, for every clause D of the form $\overline{x} \vee D'$ in Γ,

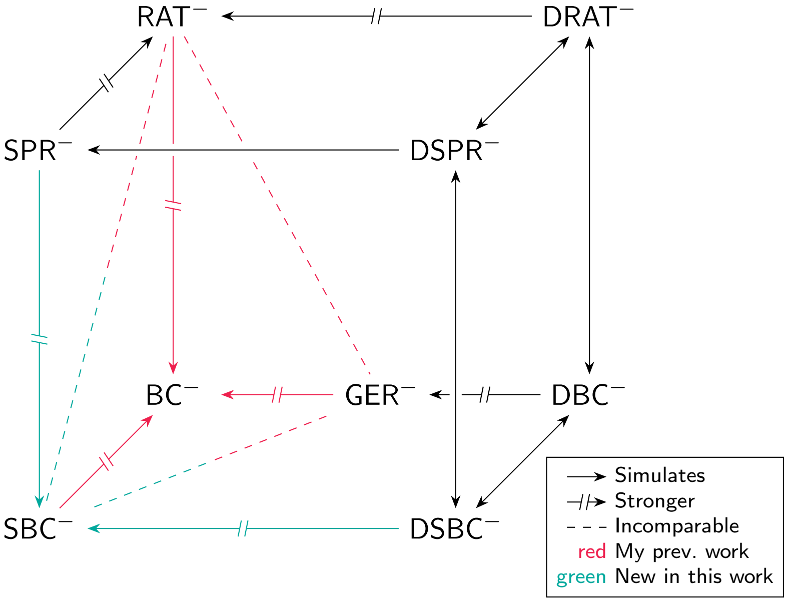
$$C' \vee D' \text{ contains a pair of complementary literals.}$$

Resolution rule + the ability to add blocked clauses results in the proof system called *blocked clauses*.

This work: Lower bounds and separations for proof systems that are based on different generalizations* of blocked clauses.

"Set-blocked clauses" generalize blocked clauses by allowing x to be a set of literals and tweaking the rest of the definition accordingly.

Results


RAT$^-$
DRAT$^-$
SPR$^-$
DSPR$^-$
BC$^-$
GER$^-$
DBC$^-$
SBC$^-$
DSBC$^-$
Simulates
Stronger
Incomparable
red My prev. work
green New in this work

Sketch of the lower bound

Γ will be a CNF that encodes a version of the pigeonhole principle.
An assignment for Γ is *light* if it does not set too many variables.

Sketch of the lower bound

Γ will be a CNF that encodes a version of the pigeonhole principle.
An assignment for Γ is *light* if it does not set too many variables.

1. Reduce lower bounds against set-blocked clauses proofs of Γ

Sketch of the lower bound

Γ will be a CNF that encodes a version of the pigeonhole principle.
An assignment for Γ is *light* if it does not set too many variables.

1. Reduce lower bounds against set-blocked clauses proofs of Γ
   to lower bounds against resolution proofs of $\Gamma \cup \Sigma$

Sketch of the lower bound

Γ will be a CNF that encodes a version of the pigeonhole principle.
An assignment for Γ is *light* if it does not set too many variables.

1. Reduce lower bounds against set-blocked clauses proofs of Γ
   to lower bounds against resolution proofs of $\Gamma \cup \Sigma$
   for every small set Σ of clauses that are derivable from Γ
   by a sequence of set-blocked clause additions.

Sketch of the lower bound

Γ will be a CNF that encodes a version of the pigeonhole principle.
An assignment for Γ is *light* if it does not set too many variables.

1. Reduce lower bounds against set-blocked clauses proofs of Γ
   to lower bounds against resolution proofs of $\Gamma \cup \Sigma$
   for every small set Σ of clauses that are derivable from Γ
   by a sequence of set-blocked clause additions.
2. Every such Σ consists solely of "complex" clauses.

Sketch of the lower bound

Γ will be a CNF that encodes a version of the pigeonhole principle. An assignment for Γ is *light* if it does not set too many variables.

1. Reduce lower bounds against set-blocked clauses proofs of Γ
   to lower bounds against resolution proofs of $\Gamma \cup \Sigma$
   for every small set Σ of clauses that are derivable from Γ
   by a sequence of set-blocked clause additions.
2. Every such Σ consists solely of "complex" clauses.
3. For every light assignment ρ, every resolution proof of $\Gamma|_\rho$ contains some complex clause.

Sketch of the lower bound

Γ will be a CNF that encodes a version of the pigeonhole principle.
An assignment for Γ is *light* if it does not set too many variables.

1. Reduce lower bounds against set-blocked clauses proofs of Γ
   to lower bounds against resolution proofs of $\Gamma \cup \Sigma$
   for every small set Σ of clauses that are derivable from Γ
   by a sequence of set-blocked clause additions.
2. Every such Σ consists solely of "complex" clauses.
3. For every light assignment ρ, every resolution proof of $\Gamma|_\rho$ contains some complex clause.
4. Suppose that $\Gamma \cup \Sigma$ has a small resolution proof π for some Σ. Then, by the probabilistic method, there is a light assignment ρ that satisfies every complex clause in Σ or π.

Sketch of the lower bound

Γ will be a CNF that encodes a version of the pigeonhole principle.
An assignment for Γ is *light* if it does not set too many variables.

1. Reduce lower bounds against set-blocked clauses proofs of Γ
   to lower bounds against resolution proofs of $\Gamma \cup \Sigma$
   for every small set Σ of clauses that are derivable from Γ
   by a sequence of set-blocked clause additions.
2. Every such Σ consists solely of "complex" clauses.
3. For every light assignment ρ, every resolution proof of $\Gamma|_\rho$ contains some complex clause.
4. Suppose that $\Gamma \cup \Sigma$ has a small resolution proof π for some Σ. Then, by the probabilistic method, there is a light assignment ρ that satisfies every complex clause in Σ or π.
5. Resolution is closed under restrictions, so there is a resolution proof of $\Gamma|_\rho$ where no clause is complex, contradicting Item 3.

Strategy for the separations

Those proof systems are unusual in the following sense: if we relax our notion of simulation slightly, they become equivalent to ER.

The relaxation allows the translation of a formula along with its proof when moving between proof systems.

Strategy for the separations

Those proof systems are unusual in the following sense: if we relax our notion of simulation slightly, they become equivalent to ER.

The relaxation allows the translation of a formula along with its proof when moving between proof systems.

Recipe: Given proof systems P, Q, R, find a function f such that

1. if Γ is easy for R, then $f(\Gamma)$ is easy for P;
2. if Γ is hard for Q, then $f(\Gamma)$ is hard for Q.

$\Longrightarrow$ If R separates from Q, then P separates from Q.

Strategy for the separations

Those proof systems are unusual in the following sense: if we relax our notion of simulation slightly, they become equivalent to ER.

The relaxation allows the translation of a formula along with its proof when moving between proof systems.

Recipe: Given proof systems P, Q, R, find a function f such that

1. if Γ is easy for R, then $f(\Gamma)$ is easy for P;
2. if Γ is hard for Q, then $f(\Gamma)$ is hard for Q.

$\implies$ If R separates from Q, then P separates from Q.

Item 1 on its own is easy to achieve. The main difficulty is to avoid inadvertently making item 2 false.

References

[BT21] Sam Buss and Neil Thapen.
DRAT and propagation redundancy proofs without new variables.
Logical Methods in Computer Science, 17(2):12:1–12:31, 2021.

[HKB20] Marijn J. H. Heule, Benjamin Kiesl, and Armin Biere.
Strong extension-free proof systems.
Journal of Automated Reasoning, 64(3):533–554, 2020.

[KRHB20] Benjamin Kiesl, Adrián Rebola-Pardo, Marijn J. H. Heule, and Armin Biere.
Simulating strong practical proof systems with extended resolution.
Journal of Automated Reasoning, 64(7):1247–1267, 2020.

[KSTB18] Benjamin Kiesl, Martina Seidl, Hans Tompits, and Armin Biere.
Local redundancy in SAT: Generalizations of blocked clauses.
Logical Methods in Computer Science, 14(4:3):1–23, 2018.

[Kul99] Oliver Kullmann.
On a generalization of extended resolution.
Discrete Applied Mathematics, 96–97:149–176, 1999.

[YH23] Emre Yolcu and Marijn J. H. Heule.
Exponential separations using guarded extension variables.
In *Proceedings of the 14th Innovations in Theoretical Computer Science (ITCS)*, number 251 in Leibniz International Proceedings in Informatics, pages 101:1–101:22. Schloss Dagstuhl, 2023.