Exponential separations using guarded extension variables

Emre Yolcu

Computer Science Department
Carnegie Mellon University

with Marijn Heule

eyolcu@cs.cmu.edu

Resolution

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

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

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

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

to derive the empty clause, which is trivially false.

Throughout this talk, "proof" means "refutation" since

proof of unsatisfiability $\equiv$ 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

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

Tree-like:

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

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:

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

Sequence-like:

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

Proof complexity

Concerned with the quantity

$$s_P(\Gamma) := \text{“size of a smallest } P\text{-proof of } \Gamma\text{”}.$$

Proof complexity

Concerned with the quantity

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

Let P and Q be proof systems.

- P *simulates* Q if there exists some c such that $\mathsf{s}_P(\Gamma) \leq \mathsf{s}_Q(\Gamma)^c$ for all Γ.
- P *is separated from* Q if there exists some sequence $(\Gamma_n)_{n\in\mathbb{N}}$ such that $\mathsf{s}_P(\Gamma_n) = n^{O(1)}$ while $\mathsf{s}_Q(\Gamma_n) = 2^{\Omega(n)}$.

More examples of proof systems

resolution, DNF-resolution, sequent calculus,
Frege, extended Frege, substitution Frege, AC^0*-Frege,*
cutting planes, Nullstellensatz, polynomial calculus,
Sherali–Adams, Lasserre, Lovász–Schrijver, . . .

They capture various restricted forms of mathematical reasoning: logical, geometric, algebraic, etc.

Without loss of generality...

Without loss of generality...

...assume that $x \geq y$.

(when proving a result that is unchanged by swapping x and y)

Without loss of generality...

...assume that $x \geq y$.
(when proving a result that is unchanged by swapping x and y)

...assume that the vertex is colored red.
(when the colors are interchangeable)

Without loss of generality...

...assume that $x \geq y$.
(when proving a result that is unchanged by swapping x and y)

...assume that the vertex is colored red.
(when the colors are interchangeable)

...assume that the random variable has expected value zero.
(when the general case can be recovered by translation)

Without loss of generality...

...assume that $x \geq y$.
(when proving a result that is unchanged by swapping x and y)

...assume that the vertex is colored red.
(when the colors are interchangeable)

...assume that the random variable has expected value zero.
(when the general case can be recovered by translation)

$\vdots$

Without loss of generality...

...assume that $x \geq y$.
(when proving a result that is unchanged by swapping x and y)

...assume that the vertex is colored red.
(when the colors are interchangeable)

...assume that the random variable has expected value zero.
(when the general case can be recovered by translation)

$\vdots$

This talk: Proof systems that capture reasoning WLOG

Example: deriving clauses WLOG

Let Γ be some formula such that the literal $\neg x$ does not appear in it. (for a really simple example, take $\Gamma = x \vee y$).

Example: deriving clauses WLOG

Let Γ be some formula such that the literal $\neg x$ does not appear in it. (for a really simple example, take $\Gamma = x \vee y$).

Without loss of generality, we may add the clause x to Γ:

Γ *is satisfiable if and only if* $\Gamma \wedge x$ *is satisfiable.*

Example: deriving clauses WLOG

Let Γ be some formula such that the literal $\neg x$ does not appear in it. (for a really simple example, take $\Gamma = x \vee y$).

Without loss of generality, we may add the clause x to Γ:

Γ *is satisfiable if and only if* $\Gamma \wedge x$ *is satisfiable.*

Note that x is not necessarily logically implied by Γ:

There might be an assignment satisfying Γ *without setting* $x = 1$.

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.

Lemma

A clause C is redundant with respect to a formula Γ if and only if there exists a partial assignment τ such that

$$\Gamma \wedge \neg C \models (\Gamma \wedge C)|_\tau.$$

Sufficient syntactic criteria for redundancy

Definition (Blocked clause)

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

$$C' \vee D' \text{ is tautological.}$$

Sufficient syntactic criteria for redundancy

Definition (Blocked clause)

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

$$C' \vee D' \text{ is tautological.}$$

Example (Blocked clause)

$$C = x \vee y \vee \neg z$$
$$\Gamma = (\neg x \vee \neg y) \wedge (\neg x \vee z) \wedge (y \vee z)$$

C is blocked for x with respect to Γ.

Sufficient syntactic criteria for redundancy

Definition (Blocked clause)

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

$$C' \vee D' \text{ is tautological.}$$

Redundancy of a blocked clause

For the assignment τ setting $x = 1$, we have $\Gamma \wedge \neg C \models (\Gamma \wedge C)|_\tau$.

Sufficient syntactic criteria for redundancy

Definition (Blocked clause)

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

$$C' \vee D' \text{ is tautological.}$$

Redundancy of a blocked clause

For the assignment τ setting $x = 1$, we have $\Gamma \wedge \neg C \models (\Gamma \wedge C)|_\tau$.

Why? Consider some total assignment α that satisfies $\Gamma \wedge \neg C$.
$\Gamma \wedge C$ is satisfied by $\alpha \circ \tau$, which is α with $\alpha(x)$ flipped.

Sufficient syntactic criteria for redundancy

Definition (Blocked clause)

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

$$C' \vee D' \text{ is tautological.}$$

Redundancy of a blocked clause

For the assignment τ setting $x = 1$, we have $\Gamma \wedge \neg C \models (\Gamma \wedge C)|_\tau$.

Why? Consider some total assignment α that satisfies $\Gamma \wedge \neg C$.
$\Gamma \wedge C$ is satisfied by $\alpha \circ \tau$, which is α with $\alpha(x)$ flipped.

What about $\neg x \vee D'$? There is some $y \in C'$ such that $\neg y \in D'$.
$\alpha \circ \tau$ still sets y to $\alpha(y) = 0$.

Sufficient syntactic criteria for redundancy

Definition (Blocked clause)

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

$$C' \vee D' \text{ is tautological.}$$

Definition (Resolution asymmetric tautology*)

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

$$C' \vee D' \text{ is tautological or subsumed by } \Gamma.$$

Sufficient syntactic criteria for redundancy

Definition (Blocked clause)

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

$$C' \vee D' \text{ is tautological.}$$

Definition (Set-blocked clause)

A clause $C = L \vee C'$ is an *SBC* for L with respect to a formula Γ if, for every clause D in Γ such that $D \cap \neg L \neq \varnothing$ and $D \cap L = \varnothing$,

$$(C \setminus L) \vee (D \setminus \neg L) \text{ is tautological.}$$

Inference rules

Definition

A *proof* in a proof system P of a formula Γ is a sequence $\Gamma_1, \ldots, \Gamma_N$ of formulas such that

- $\Gamma = \Gamma_1$, $\bot \in \Gamma_N$, and
- Γ_{i+1} follows from Γ_i by some rule of P.

Inference rules

Definition

A *proof* in a proof system P of a formula Γ is a sequence $\Gamma_1, \ldots, \Gamma_N$ of formulas such that

- $\Gamma = \Gamma_1$, $\bot \in \Gamma_N$, and
- Γ_{i+1} follows from Γ_i by some rule of P.

Rules

- *Resolution:* If $A \vee x$ and $B \vee \neg x$ are in Γ_i, add $A \vee B$ to Γ_i.

Inference rules

Definition

A *proof* in a proof system P of a formula Γ is a sequence $\Gamma_1, \ldots, \Gamma_N$ of formulas such that

- $\Gamma = \Gamma_1$, $\bot \in \Gamma_N$, and
- Γ_{i+1} follows from Γ_i by some rule of P.

Rules

- *Resolution:* If $A \vee x$ and $B \vee \neg x$ are in Γ_i, add $A \vee B$ to Γ_i.
- *Extension:* Add to Γ_i the three clauses expressing $x \leftrightarrow p \wedge q$, where p, q are arbitrary literals and x is a *new* variable.

Inference rules

Definition

A *proof* in a proof system P of a formula Γ is a sequence $\Gamma_1, \ldots, \Gamma_N$ of formulas such that

- $\Gamma = \Gamma_1$, $\bot \in \Gamma_N$, and
- Γ_{i+1} follows from Γ_i by some rule of P.

Rules

- *Resolution:* If $A \vee x$ and $B \vee \neg x$ are in Γ_i, add $A \vee B$ to Γ_i.
- *Extension:* Add to Γ_i the three clauses expressing $x \leftrightarrow p \wedge q$, where p, q are arbitrary literals and x is a *new* variable.
- *Redundancy:* Add to Γ_i a syntactically redundant clause.

Inference rules

Definition

A *proof* in a proof system P of a formula Γ is a sequence $\Gamma_1, \ldots, \Gamma_N$ of formulas such that

- $\Gamma = \Gamma_1$, $\bot \in \Gamma_N$, and
- Γ_{i+1} follows from Γ_i by some rule of P.

Rules

- *Resolution:* If $A \vee x$ and $B \vee \neg x$ are in Γ_i, add $A \vee B$ to Γ_i.
- *Extension:* Add to Γ_i the three clauses expressing $x \leftrightarrow p \wedge q$, where p, q are arbitrary literals and x is a *new* variable.
- *Redundancy:* Add to Γ_i a syntactically redundant clause.
- *Deletion:* Remove a clause from Γ_i.

Proof systems

- ER — resolution + extension
- BC — resolution + blocked clause addition
- BC^{-} — BC without new variables
- DBC — BC with deletion

  ⋮

- SPR — resolution + "SBC $\times$ RAT" addition

Proof systems

- ER — resolution + extension
- BC — resolution + blocked clause addition
- BC^- — BC without new variables
- DBC — BC with deletion

  ⋮
- SPR — resolution + "SBC $\times$ RAT" addition

"Fact"

BC generalizes ER $\implies$ BC^- is more interesting

Thus, we focus on the versions without new variables.

Lost properties

Formulas $\Gamma \subseteq \Delta$, clause C, partial assignment α not satisfying C.

- *Monotonicity:* $\Gamma \vdash C \implies \Delta \vdash C$

Lost properties

Formulas $\Gamma \subseteq \Delta$, clause C, partial assignment α not satisfying C.

- *Monotonicity:* $\Gamma \vdash C \implies \Delta \vdash C$
- *Strong soundness:* $\Gamma \vdash C \implies \Gamma \not\models C$

Lost properties

Formulas $\Gamma \subseteq \Delta$, clause C, partial assignment α not satisfying C.

- *Monotonicity:* $\Gamma \vdash C \implies \Delta \vdash C$
- *Strong soundness:* $\Gamma \vdash C \implies \Gamma \not\models C$
- *Strong closure:* $\Gamma \vdash C \implies \Gamma|_\alpha \vdash C|_\alpha$

Lost properties

Formulas $\Gamma \subseteq \Delta$, clause C, partial assignment α not satisfying C.

- *Monotonicity:* $\Gamma \vdash C \implies \Delta \vdash C$
- *Strong soundness:* $\Gamma \vdash C \implies \Gamma \not\models C$
- *Strong closure:* $\Gamma \vdash C \implies \Gamma|_\alpha \vdash C|_\alpha$

Loss of monotonicity makes *deletion* important.

Proof systems with arbitrary deletion: DBC$^-$, DRAT$^-$, etc.

Lost properties

Formulas $\Gamma \subseteq \Delta$, clause C, partial assignment α not satisfying C.

- *Monotonicity:* $\Gamma \vdash C \implies \Delta \vdash C$
- *Strong soundness:* $\Gamma \vdash C \implies \Gamma \not\models C$
- *Strong closure:* $\Gamma \vdash C \implies \Gamma|_\alpha \vdash C|_\alpha$

Loss of monotonicity makes *deletion* important.

Proof systems with arbitrary deletion: DBC^-, DRAT^-, etc.

Definition (Generalized extended resolution)

A GER proof of a formula Γ is a BC proof of some subset $\Gamma' \subseteq \Gamma$ such that every clause from $\Gamma \setminus \Gamma'$ is derived in the proof as blocked.

Some known results for generalizations of BC^-

Upper bounds

- Pigeonhole principle
- Bit pigeonhole principle
- Parity principle
- Clique-coloring principle
- Tseitin tautologies
- OR-ification, XOR-ification, lifting with index gadgets

Some known results for generalizations of BC^-

Upper bounds

- Pigeonhole principle
- Bit pigeonhole principle
- Parity principle
- Clique-coloring principle
- Tseitin tautologies
- OR-ification, XOR-ification, lifting with index gadgets

Lower bounds

- Pigeonhole principle
- Bit pigeonhole principle

Some known results for generalizations of BC^-

Upper bounds

- Pigeonhole principle
- Bit pigeonhole principle
- Parity principle
- Clique-coloring principle
- Tseitin tautologies
- OR-ification, XOR-ification, lifting with index gadgets

Lower bounds

- Pigeonhole principle
- Bit pigeonhole principle

Simulations

- Deletion collapses the systems
- No known simulations without deletion (except obvious ones)

In a *relaxed* sense, BC^- simulates ER

(without new variables!)

Lemma

Suppose that a formula Γ has an ER proof of size m and that $X = (y \vee x_1 \vee \cdots \vee x_m) \wedge y$ has no variables in common with Γ. Then $\Gamma \wedge X$ has a BC^- proof of size $O(m)$.

In a *relaxed* sense, BC^- simulates ER

(without new variables!)

Lemma

Suppose that a formula Γ has an ER proof of size m and that $X = (y \vee x_1 \vee \cdots \vee x_m) \wedge y$ has no variables in common with Γ. Then $\Gamma \wedge X$ has a BC^- proof of size $O(m)$.

Proof. Consider a use of the extension rule in the ER proof that introduces $x_i \leftrightarrow p \wedge q$. WLOG, the literals p and q are not new.

Add the clauses

$$\neg x_i \vee \neg y \vee p \qquad \neg x_i \vee \neg y \vee q \qquad x_i \vee \neg p \vee \neg q$$

in sequence as blocked clauses. Resolve against y. $\square$

In a *relaxed* sense, BC^- simulates ER

(without new variables!)

Lemma

Suppose that a formula Γ has an ER proof of size m and that $X = (y \vee x_1 \vee \cdots \vee x_m) \wedge y$ has no variables in common with Γ. Then $\Gamma \wedge X$ has a BC^- proof of size $O(m)$.

The idea of this result will be useful in proving separations across various generalizations of BC^-.

Moreover, the proofs of the separations will be completely modular, using previously known upper and lower bounds as black boxes.

Guarded extension variables

Let Γ be a formula with an ER proof of size $m = |\Gamma|^{O(1)}$.
Recall $X = (y \vee x_1 \vee \cdots \vee x_m) \wedge y$, which made $\Gamma \wedge X$ easy for BC^-.

Guarded extension variables

Let Γ be a formula with an ER proof of size $m = |\Gamma|^{O(1)}$.
Recall $X = (y \vee x_1 \vee \cdots \vee x_m) \wedge y$, which made $\Gamma \wedge X$ easy for BC^-.

To separate P and Q, incorporate extension variables into Γ in ways that are useful to only one of the two systems.

Guarded extension variables

Let Γ be a formula with an ER proof of size $m = |\Gamma|^{O(1)}$.
Recall $X = (y \vee x_1 \vee \cdots \vee x_m) \wedge y$, which made $\Gamma \wedge X$ easy for BC^-.

To separate P and Q, incorporate extension variables into Γ in ways that are useful to only one of the two systems.

Idea: Guard the variables by clauses to make them hard to access.
P will somehow use the included variables to simulate the ER proof.
Q will be unable to achieve any speedup using the included variables.

Guarded extension variables

Let Γ be a formula with an ER proof of size $m = |\Gamma|^{O(1)}$.
Recall $X = (y \vee x_1 \vee \cdots \vee x_m) \wedge y$, which made $\Gamma \wedge X$ easy for BC^-.

To separate P and Q, incorporate extension variables into Γ in ways that are useful to only one of the two systems.

Idea: Guard the variables by clauses to make them hard to access.
P will somehow use the included variables to simulate the ER proof.
Q will be unable to achieve any speedup using the included variables.

Example

With respect to the formula $(\neg x \vee y) \wedge (\neg x \vee \neg y)$, any clause blocked for x has to include both $\neg y$ and y.

Example: lower bound

Lemma

$f(\Gamma) \coloneqq \Gamma \wedge \bigwedge_{i=1}^{m} \left[(x_i \vee \Gamma) \wedge (\neg x_i \vee \Gamma) \right]$ is no easier than Γ for BC^{-}.

Example: lower bound

Lemma

$f(\Gamma) := \Gamma \wedge \bigwedge_{i=1}^{m} \left[(x_i \vee \Gamma) \wedge (\neg x_i \vee \Gamma)\right]$ is no easier than Γ for BC^{-}.

Proof.

1. View a BC^{-} proof of $f(\Gamma)$ as a resolution proof of $f(\Gamma) \wedge \Delta$, where Δ is derived by a sequence of blocked clause additions.

Example: lower bound

Lemma

$f(\Gamma) := \Gamma \wedge \bigwedge_{i=1}^{m} \left[(x_i \vee \Gamma) \wedge (\neg x_i \vee \Gamma)\right]$ is no easier than Γ for BC^{-}.

Proof.

1. View a BC^{-} proof of $f(\Gamma)$ as a resolution proof of $f(\Gamma) \wedge \Delta$, where Δ is derived by a sequence of blocked clause additions.
2. No clause in Δ can be blocked for some x_i wrt $f(\Gamma)$.

Example: lower bound

Lemma

$f(\Gamma) := \Gamma \wedge \bigwedge_{i=1}^{m} \left[(x_i \vee \Gamma) \wedge (\neg x_i \vee \Gamma)\right]$ is no easier than Γ for BC^-.

Proof.

1. View a BC^- proof of $f(\Gamma)$ as a resolution proof of $f(\Gamma) \wedge \Delta$, where Δ is derived by a sequence of blocked clause additions.
2. No clause in Δ can be blocked for some x_i wrt $f(\Gamma)$.
3. Since $\Gamma \subseteq f(\Gamma)$, every clause in Δ is in particular blocked wrt Γ.

Example: lower bound

Lemma

$f(\Gamma) := \Gamma \wedge \bigwedge_{i=1}^{m} \left[(x_i \vee \Gamma) \wedge (\neg x_i \vee \Gamma)\right]$ is no easier than Γ for BC^-.

Proof.

1. View a BC^- proof of $f(\Gamma)$ as a resolution proof of $f(\Gamma) \wedge \Delta$, where Δ is derived by a sequence of blocked clause additions.
2. No clause in Δ can be blocked for some x_i wrt $f(\Gamma)$.
3. Since $\Gamma \subseteq f(\Gamma)$, every clause in Δ is in particular blocked wrt Γ.
4. For the assignment $\alpha(x_i) = 1$, we have $(f(\Gamma) \wedge \Delta)|_\alpha = \Gamma \wedge \Delta'$, where Δ' is possible to derive from Γ in BC^-.

Example: lower bound

Lemma

$f(\Gamma) := \Gamma \wedge \bigwedge_{i=1}^{m} \left[(x_i \vee \Gamma) \wedge (\neg x_i \vee \Gamma)\right]$ is no easier than Γ for BC^-.

Proof.

1. View a BC^- proof of $f(\Gamma)$ as a resolution proof of $f(\Gamma) \wedge \Delta$, where Δ is derived by a sequence of blocked clause additions.
2. No clause in Δ can be blocked for some x_i wrt $f(\Gamma)$.
3. Since $\Gamma \subseteq f(\Gamma)$, every clause in Δ is in particular blocked wrt Γ.
4. For the assignment $\alpha(x_i) = 1$, we have $(f(\Gamma) \wedge \Delta)|_\alpha = \Gamma \wedge \Delta'$, where Δ' is possible to derive from Γ in BC^-.
5. Resolution is closed under restrictions, which implies that $f(\Gamma) \wedge \Delta$ is at least as hard for resolution as $\Gamma \wedge \Delta'$.

Example: lower bound

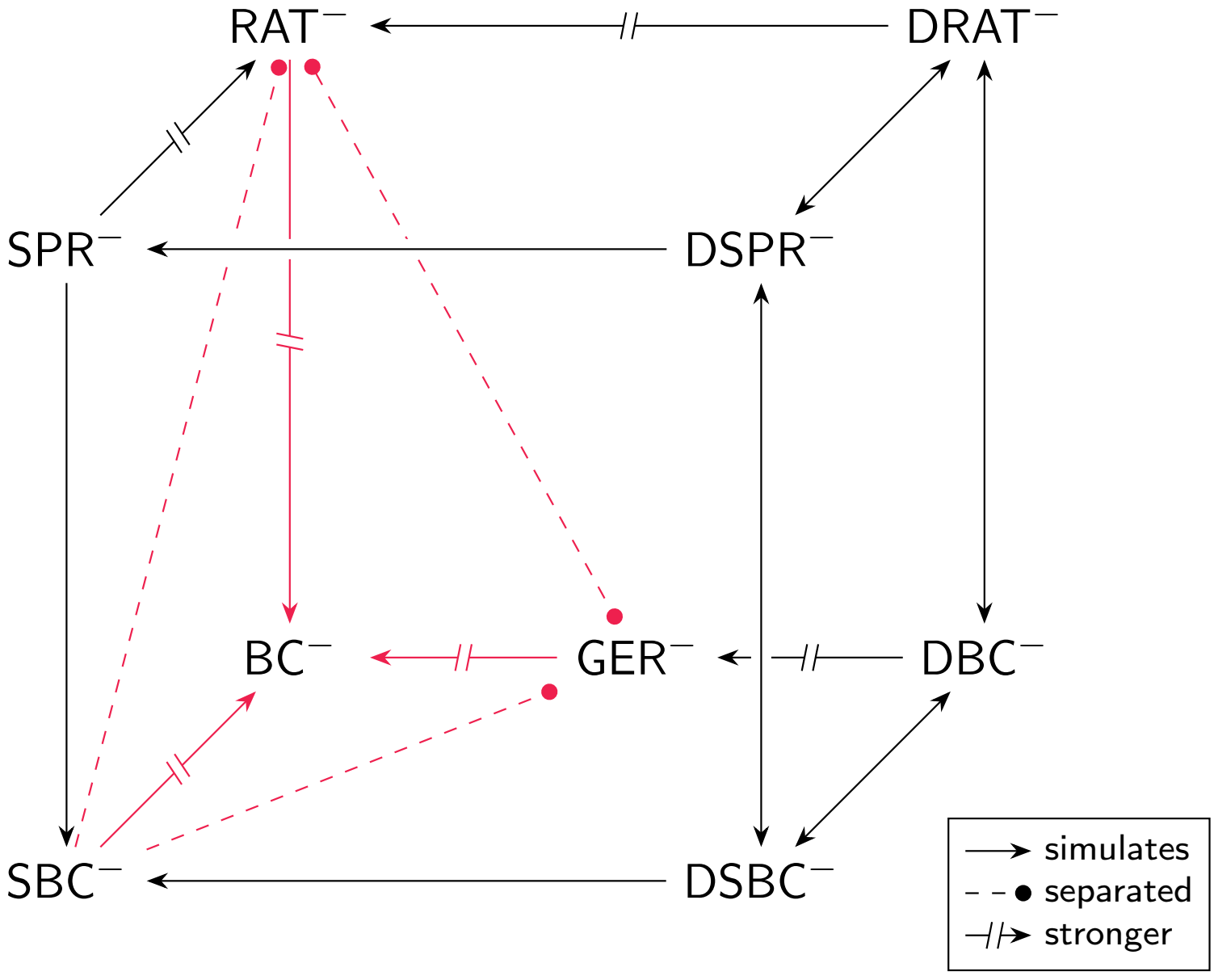
Lemma

$f(\Gamma) := \Gamma \wedge \bigwedge_{i=1}^{m} \left[(x_i \vee \Gamma) \wedge (\neg x_i \vee \Gamma)\right]$ is no easier than Γ for BC^-.

Proof.

1. View a BC^- proof of $f(\Gamma)$ as a resolution proof of $f(\Gamma) \wedge \Delta$, where Δ is derived by a sequence of blocked clause additions.
2. No clause in Δ can be blocked for some x_i wrt $f(\Gamma)$.
3. Since $\Gamma \subseteq f(\Gamma)$, every clause in Δ is in particular blocked wrt Γ.
4. For the assignment $\alpha(x_i) = 1$, we have $(f(\Gamma) \wedge \Delta)|_\alpha = \Gamma \wedge \Delta'$, where Δ' is possible to derive from Γ in BC^-.
5. Resolution is closed under restrictions, which implies that $f(\Gamma) \wedge \Delta$ is at least as hard for resolution as $\Gamma \wedge \Delta'$.
6. If $\Gamma \wedge \Delta'$ is easy for resolution, then Γ is easy for BC^-. $\square$

Summary of the new results



Separating constructions

Let Γ be a formula with an ER proof of size $m = |\Gamma|^{O(1)}$.

Separating constructions

Let Γ be a formula with an ER proof of size $m = |\Gamma|^{O(1)}$.

$\mathrm{GER}^- \not\geq \mathrm{RAT}^-$
$\mathrm{SBC}^- \not\geq \mathrm{RAT}^-$ (*)

$$f(\Gamma) := \Gamma \wedge \bigwedge_{i=1}^{m} \left[(x_i \vee \Gamma) \wedge (\neg x_i \vee \Gamma) \right]$$

$f(\Gamma)$ is easy for RAT^- (regardless of whether Γ is).

$f(\Gamma)$ is at least as hard as Γ for GER^- and SBC^-.

Separating constructions

Let Γ be a formula with an ER proof of size $m = |\Gamma|^{O(1)}$.

$\mathrm{RAT}^- \not\geq \mathrm{GER}^-$
$\mathrm{RAT}^- \not\geq \mathrm{SBC}^-$

$$g(\Gamma) := \Gamma \wedge \bigwedge_{i=1}^{m} \left[(\neg x_i \vee y_i) \wedge (x_i \vee \neg y_i) \right]$$

$g(\Gamma)$ is easy for both GER^- and SBC^- (for different reasons).

$g(\Gamma)$ is at least as hard as Γ for RAT^-.

Separating constructions

Let Γ be a formula with an ER proof of size $m = |\Gamma|^{O(1)}$.

$\mathrm{SBC}^- \not\geq \mathrm{GER}^-$ (*)

$$h_s(\Gamma) := \Gamma \wedge \bigwedge_{i=1}^{m} \bigwedge_{j=1}^{s} \left[(x_i \vee y_j \vee \neg z_j) \wedge (\neg x_i \vee y_j \vee \neg z_j) \right]$$
$$\wedge \bigwedge_{j=1}^{s} \left[(\neg y_j \vee z_j) \wedge (y_j \vee \Gamma) \wedge (\neg z_j \vee \Gamma) \right]$$

$h_s(\Gamma)$ is easy for GER^- (regardless of whether Γ is).

$h_s(\Gamma)$ is at least as hard as Γ for SBC^- with suitable choice of s.

Take home

Results

- Recipe for proving separations in a modular way
- Almost complete picture of the weakest generalizations of BC^{-}

Take home

Results

- Recipe for proving separations in a modular way
- Almost complete picture of the weakest generalizations of BC^-

Open questions

- Lower bounds for SBC^- or SPR^-
- Separations using natural combinatorial principles
- Any subsystem of Frege above resolution that DBC^- simulates
- Other uses of the high-level idea in proof complexity

References

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

[Kri85] Balakrishnan Krishnamurthy.
Short proofs for tricky formulas.
Acta Informatica, 22(3):253–275, 1985.

[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), 2018.

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