Weak versions of extended resolution

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

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

to derive the empty clause.

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.

Throughout this talk, "proof" $\equiv$ "refutation."

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:

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

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:

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

Extended resolution (ER)

At any step, derive

$$x \leftrightarrow p \wedge q,$$

where p, q are arbitrary literals and x is a *new* variable.

Extended resolution (ER)

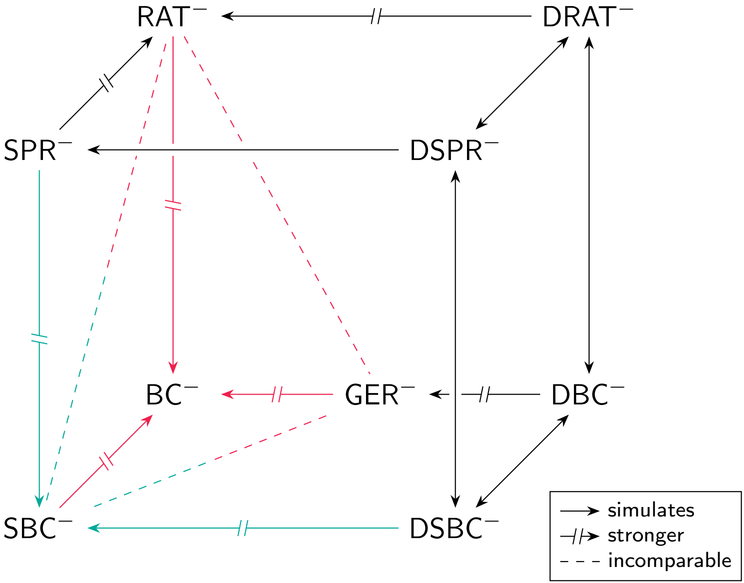
At any step, derive

$$x \leftrightarrow p \wedge q,$$

where p, q are arbitrary literals and x is a *new* variable.

This talk: Relative strengths of different weak versions of ER

Results


RAT$^-$
DRAT$^-$
SPR$^-$
DSPR$^-$
BC$^-$
GER$^-$
DBC$^-$
SBC$^-$
DSBC$^-$
simulates
stronger
incomparable

Why care?

They correspond to being able to say "without loss of generality" without needing to introduce new variables.

Why care?

They correspond to being able to say "without loss of generality" without needing to introduce new variables.

Upper bounds

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

Why care?

They correspond to being able to say “without loss of generality” without needing to introduce new variables.

Upper bounds

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

Without loss of generality, pigeon $n + 1$ is mapped to hole n...

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

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

Lemma

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

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

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,

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

Syntactic criteria for redundancy

Definition (Blocked clause)

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

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

Example (Blocked clause)

$$C = x \lor y \lor \neg z$$
$$\Gamma = (\neg x \lor \neg y) \land (\neg x \lor z) \land (y \lor z)$$

C is blocked for x with respect to Γ.

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

Claim. If assignment τ sets $x = 1$, then $\Gamma \wedge \neg C \models (\Gamma \wedge C)|_\tau$.

Syntactic criteria for redundancy

Definition (Blocked clause)

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

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

Redundancy of a blocked clause

Claim. If assignment τ sets $x = 1$, then $\Gamma \land \neg C \models (\Gamma \land C)|_\tau$.

Consider some total assignment α that satisfies $\Gamma \land \neg C$.

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

Claim. If assignment τ sets $x = 1$, then $\Gamma \wedge \neg C \models (\Gamma \wedge C)|_\tau$.

Consider some total assignment α that satisfies $\Gamma \wedge \neg C$.

Claim. $\Gamma \wedge C$ is satisfied by $\alpha \circ \tau$, which is α with $\alpha(x)$ flipped.

Syntactic criteria for redundancy

Definition (Blocked clause)

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

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

Redundancy of a blocked clause

Claim. If assignment τ sets $x = 1$, then $\Gamma \land \neg C \models (\Gamma \land C)|_\tau$.

Consider some total assignment α that satisfies $\Gamma \land \neg C$.

Claim. $\Gamma \land C$ is satisfied by $\alpha \circ \tau$, which is α with $\alpha(x)$ flipped.

Clauses of the form $\neg x \lor D'$ might be falsified by $\alpha \circ \tau$, but there is some $y \in C'$ such that $\neg y \in D'$ and $\alpha \circ \tau$ still sets y to $\alpha(y) = 0$.

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,

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

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 $D \cap L = \varnothing$,

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

The proof systems

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

  ⋮

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

  ⋮

Effectively*, BC^- simulates ER

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 $O(m)$.

Effectively*, BC$^-$ simulates ER

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. □

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

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

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

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

$f(\Gamma) := \Gamma \wedge \bigwedge_{i=1}^{m} \big[(x_i \vee \Gamma) \wedge (\neg x_i \vee \Gamma)\big]$ 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^-. $\square$

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 $m = |\Gamma|^{O(1)}$.

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

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

$f(\Gamma)$ 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 $m = |\Gamma|^{O(1)}$.

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

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

$g(\Gamma)$ 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 $m = |\Gamma|^{O(1)}$.

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

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

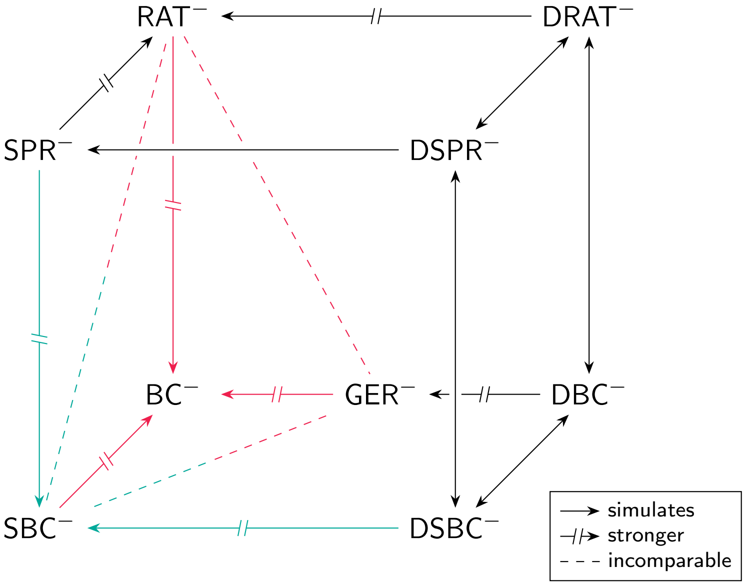
$h_s(\Gamma)$ is easy for GER^- (regardless of whether Γ is).

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

Open questions

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

Results


RAT⁻
DRAT⁻
SPR⁻
DSPR⁻
BC⁻
GER⁻
DBC⁻
SBC⁻
DSBC⁻
simulates
stronger
incomparable

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.