

Supercritical Tradeoffs between Size and Height in Resolution

Sam Buss

Lisbon, Computability in Europe (CiE)
Special Session on Proof Complexity & SAT
July 17, 2025

Talk outline:

- Resolution proofs. What is supercriticality?
- Known supercritical results on size vs height for resolution.
- Applications (some only potential) of supercriticality.
- Details of a supercritical tradeoff

Resolution is a proof system that refutes sets of clauses (i.e., a proof system for DNF formulas).

- **Variables:** $x_i, y_i, \dots$; range over *True/False*.
- **Literals:** Variables x , negated variables $\bar{x}$.
- **Clauses:** Finite sets of literals. - Interpreted as disjunctions.
- **CNF formula:** Finite set of clauses.
- **Resolution rule:**
$$\frac{C \vee x \quad D \vee \bar{x}}{C \cup D}$$
- **Refutation** of a set of clauses: Derivation that ends with the empty clause $\emptyset$.
- **Proof complexity:** The complexity of refutations can be measured in terms of size, height (depth), width, space, etc.
This talk: dag-like refutations under size/height measures.
 - **Size:** the number of clauses in the refutation
 - **Height (aka Depth):** The maximum length of any directed path through the refutation.

Importance of resolution:

- Fundamental for propositional proof search. A basis for most SAT solvers (namely, CDCL with restarts).
- Fundamental for SMT solvers.
- Asymptotic proof complexity fairly well understood from a theoretical point of view.
E.g., lower bounds for PHP, Tseitin, etc.
- Sharper lower bounds, supercritical tradeoffs can give lower bounds for monotone circuits, for cutting planes, and for communication complexity (via lifting).
- This talk: Size-Height tradeoffs. (Joint w/ Neil Thapen)
A simpler approach to supercritical lower bounds.
- And potential applications for separating CDCL without restarts from full resolution.

Supercritical lower bounds

“Supercritical lower bounds” refer generally to any lower bound that trades off different measurements of proof complexity, especially if some of the lower bounds are significantly larger than a-priori lower bounds.

[Berkholz'12] - height vs. width

[Beame-Beck-Impagliazzo'16] - size vs. space

[Razborov'16] - treelike-size/depth vs. width
and many others.

For this paper:

* **Supercritical** means tradeoffs where resolution height is greater than the number of *variables*.

* **Truly supercritical**: tradeoffs where height is greater than the number of *clauses*. [dR-F-J-N-P'24], [G-M-R-S'24]

Supercritical lower bounds:

[Fleming-Pitassi-Robere'22] & [Buss-Thapen'24] (ind.)

Thm: (version from [BT'24])

Let $k \leq n/\log^2 n$ there are CNF's with $O(n)$ variables of size $O(n^{k+2})$, that

- have refutations of size $O(n^{k+2})$, and
- any refutation of size $\leq 2^{n/k^{1/8}}$ has height $\geq n^k$.

Taking k large, this gives exponential height proofs in terms of the number of variables. Even for $k = O(1)$ it gives supercritical height lower bounds.

Both proofs start with principles for which (slightly) narrow width resolution refutations require large height. They then use “variable compression” or a “random-XOR gadget” to reduce the number of variables whilst preserving the height.

Truly supercritical lower bounds

[de Rezende-Fleming-Janett-Nordström-Pang-P'24]
[Göös-Maystre-Risse-Sokolev'24]

Thm: [dR-F-J-N-P'24] There are CNF's of size s with n variables and with resolution refutations of poly size $m = n^{O(1)}$ such that any refutation of size $< m^{1.4}$ requires height at least $s^{2.4}$.

Thm. [G-M-R-S'24] There are 3-CNFs with n variables with quasipolynomial size resolution refutations such that any refutation of size $2^{n^{o(1)}}$ requires height $n^{\Omega(1)}$.

Proofs again use lower bounds on small width refutations plus random-XOR gadgets and indexing (IND) gadgets.

Applications: lower bound tradeoffs for monotone circuit size/depth.

Also: related lower bounds on cutting planes

A CPLS-style unsatisfiable CNF, $\Gamma_{a,b}$

Def'n. For $a, b \in \mathbb{N}$, the CNF $\Gamma_{a,b}$ has the clauses

- The unit clause $\neg G_{0,0}$.
- For $i \in [a] \setminus \{a-1\}$ and $x, x' \in [b]$, the clause

$$(f(x) = x') \wedge \neg G_{i,x} \rightarrow \neg G_{i+1,x'}$$

where $G_{i,j}$'s are variables, and $f(\cdot) : [b] \rightarrow [b]$ is a function defined bitwise by variables $f_{x,j}$ giving the j -th bit of $f(x)$. Thus $(f(x) = x')$ is a conjunction of $\log b$ variables $\pm f_{x,j}$.

- The unit clause $G_{a-1,x}$ for each $x \in [b]$.

$\Gamma_{a,b}$ has $O(ab + b \log b)$ variables $G_{i,x}$ and $f_{x,j}$. It has size $O(ab^2)$.

Clauses in $\Gamma_{a,b}$:

- $\neg G_{0,0}$ (upper left node)
- $(f(x) = x') \wedge \neg G_{i,x} \rightarrow \neg G_{i+1,x'}$
- $G_{a-1,x}$ (bottom nodes)

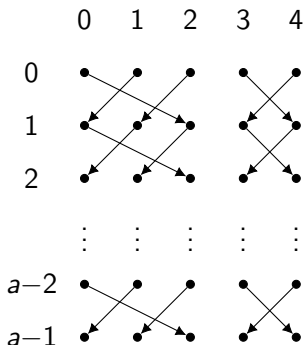
Informally:

- Something is false in top row,
- “Something is false” preserved from row to row by f ,
- Everything true in bottom row.

Arrows show $f(x)$.

The same function f is used at each row.

$G_{i,x}$ - value at row i and column x .



$\Gamma_{a,b,c}$ - a partial Or-ification of $\Gamma_{a,b}$

Def'n. For $c \geq 2$, form the **or-ification** $\Gamma_{a,b,c}$ of $\Gamma_{a,b}$, by replacing each $G_{i,x}$ with a disjunction of c new variables

$$G_{i,x}^0 \vee G_{i,x}^1 \vee \dots \vee G_{i,x}^{c-1}.$$

The clauses of $\Gamma_{a,b,c}$ are:

- The unit clauses $\neg G_{0,0}^y$ for $y < c$.
- For $i \in [a] \setminus \{a-1\}$ and $x, x' \in [b]$, the “clause”

$$(f(x) = x') \wedge \left(\bigwedge_y \neg G_{i,x}^y \right) \rightarrow \left(\bigwedge_y \neg G_{i+1,x'}^y \right)$$

- The clause $\bigvee_{y < c} G_{a-1,x}^y$ for each $x \in [b]$.

$\Gamma_{a,b}$ has $O(abc + b \log b)$ variables. It has size $O(ab^2c)$.

$\Gamma_{a,b,c}$ - a partial Or-ification of $\Gamma_{a,b}$

Def'n. For $c \geq 2$, form the **or-ification** $\Gamma_{a,b,c}$ of $\Gamma_{a,b}$, by replacing each $G_{i,x}$ with a disjunction of c new variables

$$G_{i,x}^0 \vee G_{i,x}^1 \vee \dots \vee G_{i,x}^{c-1}.$$

The clauses of $\Gamma_{a,b,c}$ are:

- The unit clauses $\neg G_{0,0}^y$ for $y < c$.
- For $i \in [a] \setminus \{a-1\}$ and $x, x' \in [b]$, **and** $y \in [c]$, the clause

$$(f(x) = x') \wedge \left(\bigwedge_y \neg G_{i,x}^y \right) \rightarrow \neg G_{i+1,x'}^y$$

- The clause $\bigvee_{y < c} G_{a-1,x}^y$ for each $x \in [b]$.

$\Gamma_{a,b}$ has $O(abc + b \log b)$ variables. It has size $O(ab^2c)$.

Thm: The set of clauses $\Gamma_{a,b,c}$ has resolution refutations of

- Size $O(abc)$ and
- Width $2c + \log b - 1$ and
- Height $O(a \cdot (c + \log b))$.

Proof idea: working from the bottom row (the row $i = a-1$) to the top row ($i = 0$), prove, for each $i \in [a]$ and $x \in [b]$, the clauses

$$G_{i,x}^0 \vee G_{i,x}^1 \vee \dots \vee G_{i,x}^{c-1}.$$

In each step, the proof splits into cases depending on the value of $f(x)$ (the variables $f_{x,j}$).

A contradiction is reached at $i = 0$ and $x = 0$. □

Height lower bounds:

We want to claim the height $\geq a$ cannot be improved.

Def'n. There are $2b$ **blocks** of variables:

- The variables $G_{i,x}^c$ for some fixed $x \in [b]$.
- The variables for $f(x)$ (i.e., $f_{x,j}$ for some fixed $x \in [b]$).

Def'n. The **block-width** of a clause is the number of blocks from which it uses literals.

The **block-width** of a refutation R is the maximum block-width of the clauses in R .

Thm. Any refutation of $\Gamma_{a,b,c}$ with block-width $\leq b/2$ must have a branch on which variables from at least a blocks are resolved on. Hence, such a refutation has height $\geq a$.

Thm. Any refutation of $\Gamma_{a,b,c}$ with block-width $\leq b/2$ must have a branch on which variables from at least a blocks are resolved on.

Proof sketch. We use a Prover-Delayer game, where the Prover is trying to falsify a clause from $\Gamma_{a,b,c}$ and the Delayer is trying to prevent this.

At each step in the game, the Prover can either:

- Query a block of variables, and the Delayer must reveal some values for the block's variables.
- Or, forget all values of variables of a block.

There is a direct correspondence between winning strategies for the Prover and refutations for $\Gamma_{a,b,c}$. Basically, the Prover follows a branch in the refutation from the concluding empty clause up to an initial clause.

Claim. If Prover never has values for $> b/2$ blocks at once, then the Prover must query at least a distinct blocks. $\square$

Variable compression — $\Phi_{b,c,d}$

We now reduce the number of variables whilst preserving a resolution height lower bound.

Let $a = d^c$. Integers $i \in [a]$ are written in base d with c digits:

$$i = (i_0 i_1 \cdots i_{c-2} i_{c-1})_d.$$

Create new variables $H_{i',x}^y$ where $i' \in [d]$.

Define $\Phi_{b,c,d}$ from $\Gamma_{a,b,c}$ by

replacing each variable $G_{i,x}^y$ with $H_{i_y,x}^y$.

E.g.: $\bigvee_{y \in [c]} G_{0,x}^y$ becomes $H_{0,x}^0 \vee H_{0,x}^1 \vee \cdots \vee H_{0,x}^{c-2} \vee H_{0,x}^{c-1}$
 $\bigvee_{y \in [c]} G_{1,x}^y$ becomes $H_{0,x}^0 \vee H_{0,x}^1 \vee \cdots \vee H_{0,x}^{c-2} \vee H_{1,x}^{c-1}$

etc.

Thm: The set of clauses $\Phi_{b,c,d}$ has

- $bcd + b \log b$ variables and
- fewer than $d^c b^2 c$ clauses,

resolution refutations of

- Size $O(d^c b^2 c)$ and
- Width $2c + \log b - 1$ and
- Height $O(d^c \cdot (c + \log b))$.

Thm. Any refutation of $\Phi_{b,c,d}$ with block-width $\leq b/2$ must have a branch on which variables from at least $a = d^c$ blocks are resolved on.

Hence, such a refutation has height $\Omega(d^c) = \Omega(a)$.

Next: Improve this width/height tradeoff to a size/height tradeoff.

Thm. Any refutation of $\Phi_{b,c,d}$ of size $2^{b^{1/4}}$ must have a branch on which variables from at least $a = d^c$ blocks are resolved on.

Proof sketch: Given a refutation of size $\leq 2^{b^{1/4}}$, hit it with a random restriction that removes approximately $3b^{-2/3}$ of the columns at random:

- For each column x , with probability $b^{-2/3}$, choose a random free column x' and set $f(x) = f(x') = x'$.
- For each column x , with probability $b^{-2/3}$, set $f(x) = x$ and set the variables $H_{i',x}^y$ to true for a randomly chosen half of the y 's and to false for the rest of the y 's.

Claim: With high probability, this random restriction reduces the block-width of the refutation to $< b/4$.

So, by the previous theorem, the refutation height is $\geq a$. □

Sample tradeoff:

Thm. Let $1 \leq k < n/\log^2 n$. There is a CNF with

- n variables,
- $O(n^{k+2})$ clauses,
- a refutation of size $O(n^{k+2})$

such that

- any refutation of size $< 2^{(n/k)^{1/8}}$ has height at least n^k .

Pf: Use $\Phi_{b,c,d}$ with $c = k \log n$ and $b = n/c$ and $d = 2$. □

Conjectures/Questions for CDCL without restarts

CDCL - Clause-Driven Conflict Learning.

CDCL is the core reasoning system for most SAT Solvers. It is simulated by resolution, and when combined with Restarts it polynomially simulates resolution.

[Marques-Silva-Sakallah'99], [Beame-Kautz-Sabharwal'04]

Pipatsisawat-Darwiche'08 and Atserias-Fichte-Thurley'11]

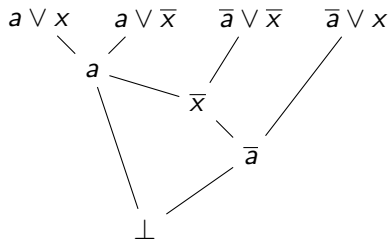
Open Question: Does CDCL *without* Restarts polynomially simulate Resolution?

To formulate this rigorously, need to formalize CDCL without restarts. There are two principal methods:

- a. **Pool Resolution.** Resolution refutations with a depth-first regular traversal. [Van Gelder'05].
- b. **RegWRTL.** A more restricted system (apparently?), based a closer match to CDCL-style conflict clause learning. [Buss-Hoffmann-Johannsen'08].

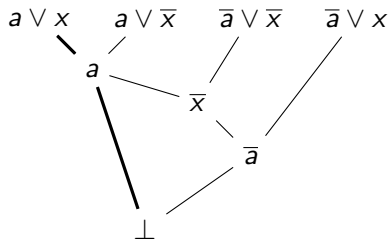
Def'n. A *pool resolution refutation* is a resolution refutation that admits a depth-first regular traversal. [Van Gelder'05]

Example:



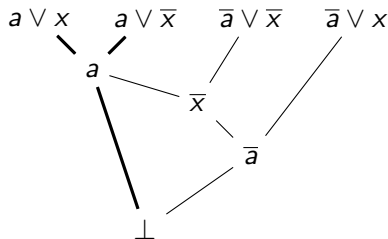
Def'n. A *pool resolution refutation* is a resolution refutation that admits a depth-first regular traversal. [Van Gelder'05]

Example:



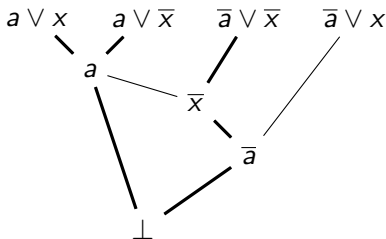
Def'n. A *pool resolution refutation* is a resolution refutation that admits a depth-first regular traversal. [Van Gelder'05]

Example:



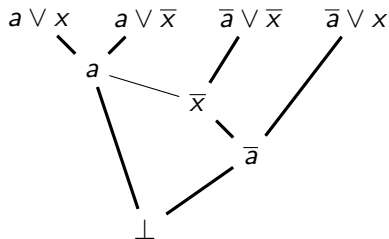
Def'n. A *pool resolution refutation* is a resolution refutation that admits a depth-first regular traversal. [Van Gelder'05]

Example:



Def'n. A *pool resolution refutation* is a resolution refutation that admits a depth-first regular traversal. [Van Gelder'05]

Example:



Earlier candidates for separating pool resolution (or RegWTRL) proofs arose from work separating regular resolution from resolution. But these have all failed to give superpolynomial separations of pool resolution (or RegWTRL) proofs from resolution, including

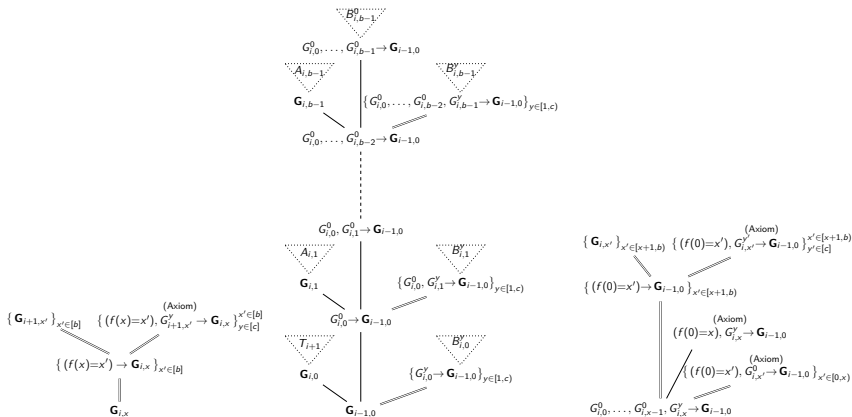
- guarded graph/pebbling principles [Bonet.-B.-Johannsen'08], [Bonet-Buss'12] and
- the Stone tautologies [B.-Kołodziejczyk'14].

— —

Because they require resolving on the same variables many times, short proofs of formulas with (super)critical size/depth tradeoffs require highly irregular proofs.

Thus these provide another set of candidates, that have not been well explored yet.

Thm: [BT'24] The $\Gamma_{a,b,c}$ CNFs have polynomial size pool and RegWRTL refutations.



Open Question: Do the $\Phi_{b,c,d}$ CNF's have polynomial size pool refutations or RegWRTL separations?

What about formulas with supercritical size/depth tradeoffs?

- Do the lifted versions of the bracket tautologies of [GMRS'25] have polynomial size pool refutations or RegWRTL refutations?
- Is there a general principle that implies lifting with index gadgets (or, other gadgets) cannot be used to give superpolynomial separations of pool (or RegWRTL) resolution from general resolution?

Thank you!

—

Obrigado!