Some Challenge Problems for Resolution and One Also for AC^0-Frege

Sam Buss

Shonan Village
October 2, 2023

Motivation: Superpolynomial separations for AC^0 proofs.

Original Motivations:

1. **Frege proofs** - Propositional proofs using Modus Ponens
   Elegantly formalizable in the sequent calculus (LK).
2. **Depth of a formula:** Counts the alternation of $\wedge$'s and $\vee$'s.
   (Use negations only on variables.)
3. **CNF formulas** can be viewed as depth 1 formulas. (Or zero.)
   Resolution refutations as depth zero LK refutations.
4. **Open:** Are there CNFs with short depth $k+1$ Frege proofs, but require exponential size depth k Frege proofs?
5. **Best result so far:** Quasipolynomial ($2^{(\log n)^{O(1)}}$) separation for the pigeonhole principle. [Krajicek-Impagliazzo'02].

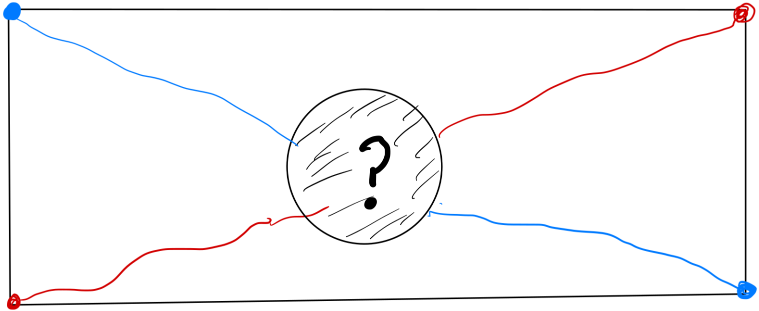
This talk:

1. **Three proposals**. Two easy for resolution (!), one open.
2. **Challenge problems.** Proposed as challenges for SAT solvers.

Part I. st-Connectivity Tautologies

st-Connectivity (non-crossing) tautologies:

Two paths cannot cross diagonally without intersecting.



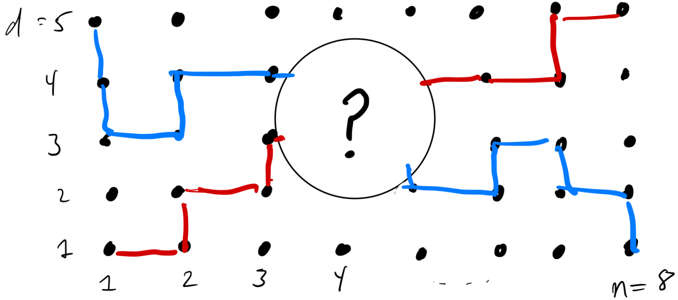
The red and blue paths must intersect somewhere

[B'06], "Polynomial-size Frege and Resolution Proofs of st-Connectivity and Hex Tautologies

Part I. st-Connectivity Tautologies

st-Connectivity (non-crossing) tautologies:

Two paths cannot cross diagonally without intersecting.



A width d grid graph. d is constant. n varies.

Propositional variables: r_e and b_e indicating e on red/blue path. Edge e is an unordered pair $\{\langle i,j\rangle,\langle i{+}1,j\rangle\}$ or $\{\langle i,j\rangle,\langle i,j{+}1\rangle\}$. $i \leq n$ and $j \leq d$.

[B'06], "Polynomial-size Frege and Resolution Proofs of st-Connectivity and Hex Tautologies

Clauses for Grid Graph st-Connectivity: (GridStConn)

- End points of red and blue paths (8 clauses):
  - $\mathit{OneOf}(\{r_e : \langle 1,1\rangle \in e\})$, $\mathit{OneOf}(\{r_e : \langle n,d\rangle \in e\})$
  - $\mathit{OneOf}(\{b_e : \langle 1,d\rangle \in e\})$, $\mathit{OneOf}(\{b_e : \langle n,1\rangle \in e\})$
- Intermediate points v on paths ($O(nd)$ clauses):
  - $\mathit{ZeroOrTwoOf}(\{r_e : v \in e\})$ and $\mathit{ZeroOrTwoOf}(\{b_e : v \in e\})$[1]
- Paths are vertex disjoint ($O(nd)$ clauses):
  - $\overline{r_e} \vee \overline{b}_f$, for $e \cap f \neq \emptyset$.

Theorem: Fix $d \in \mathbb{N}$. The st-Connectivity Decision Problem of whether there is a path from s to t is many-one complete for Π_d-Boolean circuits. ($\Pi_d =$ "depth d".)
[Barrington-Lu-Militerson-Skyun'98]

Nonetheless . . .

[1] OneOf is Xor ($\oplus$). $\mathit{ZeroOrTwoOf}$ is $\overline{\mathit{Xor}}$.

Theorem: GridStConn has resolution refutations of size $poly(n \cdot 2^d)$. These are polynomial size for d a constant.

Two ways of viewing the proof:

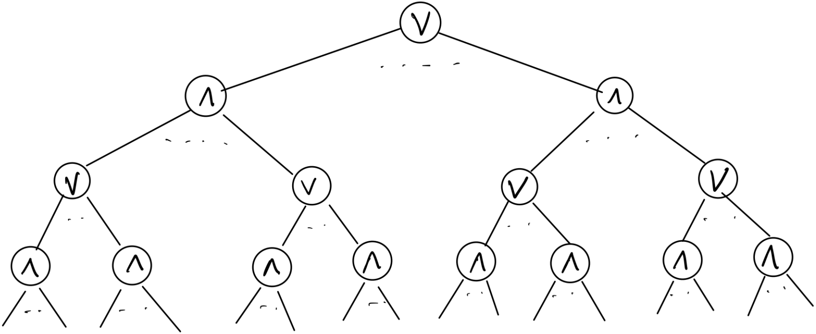
- **Induction style argument:** Scan from left-to-right ruling out appropriate patterns of crossing sequences of red and blue paths.
- **Bounded Tree Width / Decision Tree argument:** Divide-and-conquer by querying the middle column of edges. Then branch left or right depending on the crossing sequence of red and blue edges. Recurse until reach a contradiction.

Challenge Problem: Do SAT solvers efficiently find proofs of GridStConn for d constant? What if d is allowed to vary?

Theorem: When $d = n$, the PHP tautologies are reducible to GridStConn. Thus resolution refutations must be exponential size.

Part II. AC^0 Equivalence and Implication Tautologies

Boolean formula of depth d and constant unbounded fanin f:

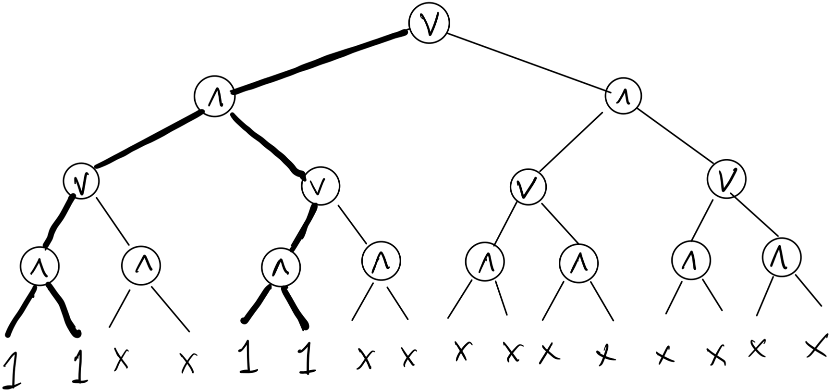


Alternating $\vee$'s and $\wedge$'s, with $d = 4$ and $f = 2$.

For fixed d, evaluating the truth of a formula is Σ_d complete.

Part II. AC^0 Equivalence and Implication Tautologies

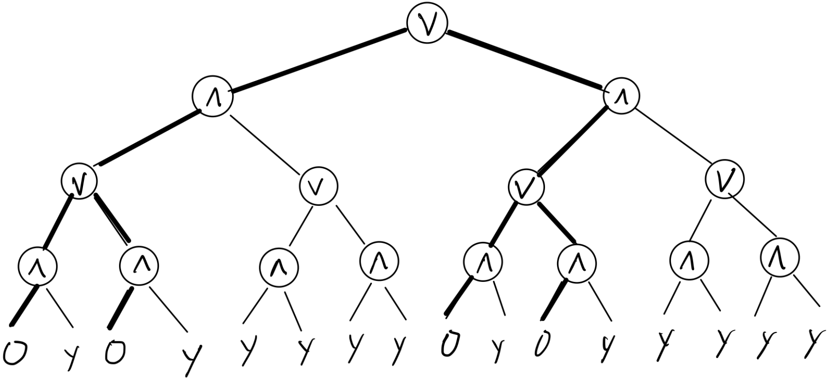
An obviously true Boolean formula:



The **obviously true** formula.
The first inputs to ∨'s evaluate to true (1) (as needed).
Any path that always takes the first input from an ∨ reaches "1".
The x's are "Don't Care" values.

Part II. AC^0 Equivalence and Implication Tautologies

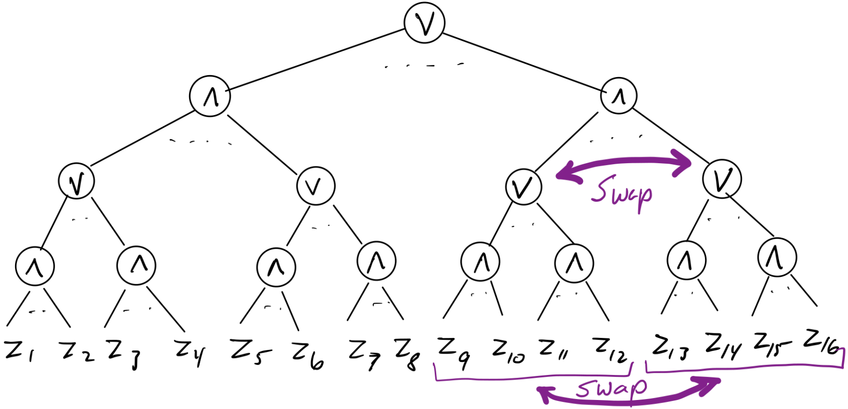
An obviously false Boolean formula:



The **obviously false** formula.
The first inputs to ∧'s evaluate to false (0) (as needed).
Any path that always takes the first input from an ∧ reaches "0".
The y's are "Don't Care" values.

Swaps as primitive equivalence steps

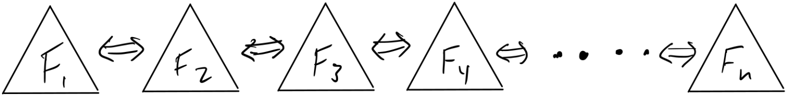
Swapping the order of inputs does not change truth value:



A swap π of one pair of inputs to one gate is a **primitive swap**.
Here $\pi(i) = i$ for $i \leq 8$, $\pi(9+j) = 13+j$ and $\pi(13+j) = 9+j$.

AC^0 Formula Equivalence Tautologies (FmlaEquiv)

Conceptually: A sequence of n formulas, each equivalent to the next, starting with a true formula and ending with a false formula. The formulas have depth d and fanin f. [2]



- F_1 is obviously true.
- F_n is obviously false.
- Each F_{i+1} is obtained from F_i by a primitive swap.

Propositional Variables:

- $x_{i,\ell}$ is the Boolean value of the ℓ-th input to F_i.
- $s_{i,p}$ means F_{i+1} is obtained from F_i by primitive swap π_p.
  p encodes a gate and two of its inputs.

$\ell \le f^d$ and $p \le (f^d - 1)/(d-1)$ and $i \le n$ (or $< n$ for $s_{i,p}$).

[2] FmlaEquiv is based on a suggestion of Krajicek.

Clauses of FmlaEquiv: Parameters d, f, n.

- $\boldsymbol{F_1}$ is obviously true. Unit clauses for $f^{d/2}$ many inputs of F_1 are set to 1.
- $\boldsymbol{F_n}$ is obviously false. Unit clauses for $f^{d/2}$ many inputs of F_n are set to 0.
- **Don't care values** of F_1 and F_n are set to 0 and to 1, respectively. (Only need one set of these.)
- **One primitive swap for each $i < n$:** $\bigvee_p s_{i,p}$ and $\overline{s_{i,p}} \vee \overline{s_{i,p'}}$ (for $p \neq p'$)
- **Primitive swap preserves truth & falsity**. If $s_{i,p}$ is true, then $x_{i,\ell} \leftrightarrow x_{i+1,\pi_p(\ell)}$.

FmlaEquiv is a **CNF formula** and is clearly unsatisfiable.

The straightforward proof of unsatisfiability involves showing, successively by induction on i, that F_i evaluates to true, and reaching a contradiction at F_n.
However, expressing truth requires depth d formulas.
Thus this proof cannot be carried out in resolution.

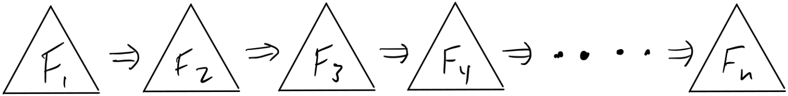
Theorem: [B-Ramyaa'18]
FmlaEquiv has polynomial size resolution refutations.

The proof is tricky, but the idea is that the extra "Don't Care" variables contain enough information to let resolution express the condition that the formula F_i is a permuted (via multiple swaps) version of F_1.

Challenge Problem for resolution: Do SAT solvers refute FmlaEquiv efficiently?

AC^0 Formula Implication Tautologies (FmlaImply)

Conceptually: A sequence of n formulas, each **implying** the next, starting with a true formula and ending with a false formula. The formulas have depth d and fanin f.



- F_1 is obviously true.
- F_n is obviously false.
- Each F_{i+1} is obtained from F_i by a primitive swap,
- Plus, possibly changing some 0 inputs to 1's.

So $F_i \to F_{i+1}$ is assumed, not $F_i \leftrightarrow F_{i+1}$.

Clauses of FmlaImply: Parameters d, f, n.

- $\boldsymbol{F_1}$ is obviously true. Unit clauses for $f^{d/2}$ many inputs of F_1 are set to 0.
- $\boldsymbol{F_n}$ is obviously true. Unit clauses for $f^{d/2}$ many inputs of F_n are set to 1.
- **Don't care values** are no longer important.
- **One primitive swap for each** $i < n$**:**
  $\bigvee_p s_{i,p}$ and $\overline{s_{i,p}} \vee \overline{s_{i,p'}}$ (for $p \neq p'$)
- **Primitive swap preserves truth implicationally**. If $s_{i,p}$ is true, then $x_{i,\ell} \to x_{i+1,\pi_p(\ell)}$.

Open Question: Does resolution have polynomial size refutations of FmlaImply?

Challenge Problem: How do SAT solvers perform on FmlaImply?

Thm: FmlaImply has poly-size, tree-like LK, depth $d-1$ refutations.

Open Question: Does FmlaImply give exponential separations for (tree-like LK) depth $d-2$ versus depth $d-1$ proof size?
(Note that these systems are much stronger than resolution.)

If yes, it gives similar separations for dag-like LK refutations.

Thank you!