LECTURE 23

Last time

Proof system P automatizable if $\exists$ algorithm A_P which
- takes unsat CNF F as input
- runs in time $\text{poly}(S(F) + S_P(F \vdash \bot))$
- outputs P-refutation of F

Want to cover result in [Alekhnovich-Razborov '08]
Resolution and tree-like resolution are not automatizable (unless some hypothesis in parameterized complexity fails to hold).

MONOTONE MINIMUM CIRCUIT SATISFYING ASSIGNMENT (MMCSA)

Instance: Monotone Boolean circuit $C(p_1, \dots, p_n)$
Solution: $\vec{x} \in \{0,1\}^n$ s.t. $C(\vec{x}) = 1$
Objective function: $k(\vec{x})$ = Hamming weight $wt(\vec{x})$ = number of 1s in $\vec{x}$

Denote optimal minimum value by $k(C)$
W[P]-complete problem — according to hypothesis in parameterized complexity should not have algorithm running in FPT-time $f(k(C)) \cdot \text{poly}(|C|)$

THEOREM 1 [AR08]

If either resolution or tree-like resolution is automatizable, then for any fixed $\varepsilon > 0$ exists algorithm Φ that:

- takes monotone circuit C as input
- runs in time $\exp(k(C)^{O(1)}) \; |C|^{O(1)}$
  [or $\exp(\text{poly}(k(C)) \; \text{poly}(|C|))$]
- approximates $k(C)$ to within factor $1+\varepsilon$.

How to prove theorem?

1. Bake MMCSA problem into other problem with nice properties
2. Express this new problem as CNF formula
3. Prove upper and lower bounds in resolution
4. Tie things together nicely

Last time: did 1 and 2, but failed miserably at 3.

This lecture: redo 2 so that 3 can succeed

Quick recap of 1

$A \subseteq \{0,1\}^m$ set of vectors

Matrix M A-admissible, or A-generated, if every column of M is a vector in A

DEF 2 (C, t)-$\exists$SAT

For every $m \times n$ t-generated matrix $M = (m_{ij})$ there exists a row $i \in [m]$ such that

$$C(m_{i,1}, \ldots, m_{i,n}) = 1$$

GAME PLAN (FROM LAST TIME)

1. Given C, find t so that (C, t)-$\exists$SAT true
2. Encode (C, t)-$\exists$SAT as unsat CNF formula claiming that (C, t)-$\exists$SAT false
3. Prove that refuting such formulas in resolution requires large width
4. Hit with random restriction so that all wide (large) clauses in short refutation disappears
5. Yields (short) refutation of same kind of formula without wide clauses — contradiction to (3)

Also need upper bound but that won't be too hard.

Formula needs to say things like

If
- $\vec{a} \in A$ chosen for column j, and
- C evaluated on row i, and
- $\vec{a}_i = 1$

then

Input p_j in C should be 1

How to encode such choices and implications?

Collection/set $C = \{c_0, c_1, \dots, c_{N-1}\}$ of size N

Variant 1

x_i = "element c_i chosen"

Some c_i chosen: $\bigvee_{i=0}^{N-1} x_i$ (a)

At most one c_i chosen: $\bigwedge_{0 \le i < j < N} (\bar{x}_i \vee \bar{x}_j)$ (b)

Exactly one element chosen: (a) + (b)

"If element c_i chosen, then D holds"

$x_i \to D \quad = \quad \bar{x}_i \vee D$

This approach didn't work for us — random restrictions destroy too much structure

Variant 2

Suppose for simplicity $N = 2^n$ so that $C = \{c_0, c_1, \dots, c_{2^n-1}\}$

Let variables $x_{n-1}, x_{n-2}, \dots, x_1, x_0$ be index i of chosen element written in binary

Any assignment σ sets $\vec{x} = (x_0, \dots, x_{n-1})$ in some way $\Rightarrow$ one and exactly one element chosen

Recall $x^\sigma = \begin{cases} \bar{x} & \text{if } \sigma = 0 \\ x & \text{if } \sigma = 1 \end{cases}$. Introduce shorthand

$\vec{x}^\sigma = x_0^{\sigma_0} \wedge x_1^{\sigma_1} \wedge \dots \wedge x_{n-1}^{\sigma_{n-1}}$

$\neg\vec{x}^\sigma = x_0^{1-\sigma_0} \vee x_1^{1-\sigma_1} \vee \dots \vee x_{n-1}^{1-\sigma_{n-1}}$

"If element $\sigma \in \{0,1\}^n = [0, N-1]$ (Go for σ) chosen, then D holds"

$$\vec{x}^{\sigma} \to D \;=\; \neg \vec{x}^{\sigma} \vee D$$

(Note: this is a clause)

If σ chosen, then all literals in $\neg \vec{x}^{\sigma}$ false, so D must hold or else the whole clause is falsified

Now better properties w.r.t. random restrictions

Can set several bits to 0 and 1 uniformly at random without revealing too much about choice

But not good enough — we want to be able to set many bits randomly without revealing anything

Variant 3

shift to $C = \{c_1, \ldots, c_N\}$

Choose $s \gg \log N$

Choose some onto (surjective) function $f: \{0,1\}^s \to [N]$

Element c_i chosen if we have assignment $\sigma \in f^{-1}(i)$

(automatically get)

Again one and exactly one element chosen.

"If c_i chosen, then D holds"

$$\bigwedge_{\sigma \in f^{-1}(i)} \left(\vec{x}^{\sigma} \to D \right) = \bigwedge_{\sigma \in f^{-1}(i)} \left(\neg \vec{x}^{\sigma} \vee D \right)$$

↖ CNF formula

If $s \gg \log N$ large enough, and if f behaves nicely under random restrictions, then we are in business!

We will need the following fact.

OBSERVATION 3
For any $n \in \mathbb{N}^+$, the CNF formula over variables $x_1, x_2, \ldots, x_n$ consisting of the clauses

$$\{ \neg \vec{x}^\sigma \mid \sigma \in \{0,1\}^n \}$$

is refutable in tree-like resolution in length $2^{n+1} - 1$

Proof Build the ~~decision~~ search tree that first queries x_1, then in both children of this node queries x_2, et cetera, down to the last level above the leaves where variable x_n is queried. This is a tree of size $2^{n+1} - 1$. Now use the equivalence between ~~decision~~ search trees and tree-like resolution.

Encode choices of n column vectors from A by functions $F_j : \{0,1\}^s \to A$, $j \in [n]$

Introduce r copies $C_1, \ldots, C_r$ of circuit C

Every row of matrix evaluated (correctly) on some copy C_i

For every row i encode choice of circuit copy C_i by function $f_i : \{0,1\}^s \to [r]$, $i \in [m]$

Functions f_i can be partial

DEFINITION [OF FORMULA $\Sigma(C, A, \vec{F}, \vec{f})$]

$C(p_1, \ldots, p_n)$ monotone circuit with output v_{out}

$A \subseteq \{0,1\}^m$ set of vectors

$F_j : \{0,1\}^s \to A$, $j \in [n]$; $f_i : \{0,1\}^s \to [r]$, $i \in [m]$

surjective (onto) functions, f_i possibly partial

Note: All of these fixed; given as inputs; define formula

Variables:

x_j^v $j \in [n]$, $v \in [s]$ "inputs to F_j's"

y_i^v $i \in [m]$, $v \in [s]$ "inputs to f_i's"

(refer to c as "controls")

$z_{i,v}^{(c)}$ $c \in [r]$, $i \in [m]$, v node in C

"value of node v in circuit copy C_c when evaluated on row i"

Notational shorthands:

$[col_j = \vec{a}]$ = "$F_j(x_j^1, \ldots, x_j^s) = \vec{a}$"

$[ctrl_i = c]$ = "$f_i(y_i^1, \ldots, y_i^s)$ is defined and is $= c$"

Recall "$(x \wedge y) \to z$" shorthand for $\bar{x} \vee \bar{y} \vee z$

Then $\tau(C, t, \vec{F}, \vec{f})$ is the expansion to CNF of the following Boolean predicates

(i) $(y_i^1, \ldots, y_i^s) \in Dom(f_i)$ $\quad i \in [m]$

"f_i only evaluated on inputs where it is defined" (so every row gets some circuit copy)

(ii) $\left([col_j = \vec{a}] \wedge [ctrl_i = c]\right) \rightarrow z_{i,p_j}^c$ $\quad c \in [r]$, $j \in [n]$

all $\vec{a} \in t$ and $i \in [m]$ s.t. $a_j = 1$

"if column j is $\vec{a}$, ith row evaluated in circuit copy c, and $a_i = 1$, then input variable p_j is set to 1 in this circuit copy"

(iii) $\left([ctrl_i = c] \wedge \left(z_{i,u}^c \circ z_{i,v}^c\right)\right) \rightarrow z_{i,w}^c$

$i \in [m]$, $c \in [r]$

w internal node labelled by $\circ \in \{\wedge, \vee\}$ and with wires (edges) from u, v

"if row i evaluated on circuit copy c, then internal node w evaluated correctly given inputs"

(iv) $[ctrl_i = c] \rightarrow \overline{z}_{i,v_{out}}^c$

"if row i evaluated on circuit copy c, then the output of this circuit copy is 0"

OBSERVATION 5

If (C, t) - $\exists SAT$ true, then $\tau(C, t, \vec{F}, \vec{f})$ unsat for any total functions $\vec{F}$ and partial functions $\vec{f}$.

Some remarks

(1) By "correct evaluation", we actually just mean that 1s are propagated through circuit correctly

Intuitively, the formula $\tau(C, t, \vec{F}, \vec{f})$ says that the circuit evaluates to false, and since C is monotone it can never be a good idea to introduce spurious 1s

Formally, suppose we have satisfying assignment to $\tau(C, t, \vec{F}, \vec{f})$. Then we can flip any "spurious" $z^C_{i,v} = 1$ to 0 without falsifying the formula

(2) How to "expand to CNF"? Some concrete examples (in addition to our discussion above)

For clauses (i) suppose f_i not defined on $\vec{b} \in (0,1)^s$

Recall notation $x^b = \begin{cases} \bar{x} & \text{if } b=0 \\ x & \text{if } b=1 \end{cases}$

To rule out that $\vec{b} \in \{0,1\}^s$ is fed to f_i, we add the clause

$$\neg y_i^{\vec{b}} = (y_i^1)^{1-b_1} \vee (y_i^2)^{1-b_2} \vee \cdots \vee (y_i^s)^{1-b_s}$$

For a concrete example, suppose we don't want the string $\vec{b} = (0\,1\,0\,1\,0\,1\,\ldots)$. Then the clause will be

$$y_i^1 \vee \bar{y}_i^2 \vee y_i^3 \vee \bar{y}_i^4 \vee \cdots$$

The truth value of $[col_j = c]$ is determined by assignments to variables y_i^b

f_i is fixed, so we know for which $b \in \{0,1\}^s$ we have $f_i(\vec{b}) = c$

To encode "$[ctrl_i = c] \to$ clause D" consider all such vectors $\vec{b}$ and generate the CNF formula

$$\bigwedge_{\substack{\vec{b} \in \{0,1\}^s \\ f_i(\vec{b}) = c}} \left((y_i^1)^{1-b_1} \vee \dots \vee (y_i^s)^{1-b_s} \vee D \right)$$

To encode "$([col_j = \vec{a}] \wedge [ctrl_i = c]) \to$ clause D" generate the CNF formula

$$\bigwedge_{\substack{\vec{b}' \in \{0,1\}^s \\ F_j(\vec{b}') = \vec{a}}} \bigwedge_{\substack{\vec{b}'' \in \{0,1\}^s \\ f_i(\vec{b}'') = c}} \left(\begin{array}{l} (x_j^1)^{1-b'_1} \vee \dots \vee (x_j^s)^{1-b'_s} \vee \\ (y_i^1)^{1-b''_1} \vee \dots \vee (y_i^s)^{1-b''_s} \vee \\ D \end{array} \right)$$

The point of $F_1, \ldots, F_n$ and $f_1, \ldots, f_m$ is that we want to be able to

(a) hit x_j^v and y_i^v with random restrictions, but

(b) still maintain the full choice of vectors in A for columns j and copies C_c for rows i

DEF 6 An onto (possibly partial) function $g : \{0,1\}^s \to R$ is r-SURJECTIVE if for any restriction ρ with $|Dom(\rho)| \leq r$ it holds that $g|_\rho$ is still onto.

We need two measures on how 1s can be distributed in rows and 0s distributed in columns for any A-admissible matrix (or maybe we should say A-generated matrix?).

DEF 7 Given $A \subseteq \{0,1\}^m$ we let:

- $d_1(A)$ = max d s.t. for any d vectors $\vec{a}^{(1)}, \ldots, \vec{a}^{(d)} \in A$ $\exists$ position i s.t. $\vec{a}_i^{(1)} = \vec{a}_i^{(2)} = \cdots = \vec{a}_i^{(d)} = 1$

- $d_0(A)$ = max d s.t. for any d disjnct positions $i_1, \ldots, i_d \in [m]$ $\exists$ vector $\vec{a} \in A$ s.t. $\vec{a}_{i_1} = \cdots = \vec{a}_{i_d} = 0$.

Recall: If $k(C) \leq d_1(A)$, then (C, A)-$\exists$SAT definitely true

Intuition: $d_0(A)$ large $\Rightarrow$ hard to prove we will be able to find needed 1s to feed into circuit

Recall our lower bound proof strategy

(i) Prove lower bound on resolution refutation width

(ii) Argue with the help of random restrictions that (i) implies length lower bounds for resolution refutations

We will need to "hack" the width measure a bit

DEF 8 Controlled width

Let D clause over Vars $(\tau(C, t, \vec{F}, \vec{f}))$.

Define

$W_x(D)$ = # variables x_j^v in D

$W_y(D)$ = # variables y_i^v in D

$W_c(D)$ = # variables $z_{i,v}^c$ in D
(note that c is a fixed value!)

Then the CONTROLLED WIDTH of a clause D is

$$\tilde{W}(D) = W_x(D) + W_y(D) + r \cdot \min_{c \in [r]} W_c(D)$$

The controlled width of refuting $\tau(C, t, \vec{F}, \vec{f})$ is

$$\min_{\pi : \tau(C,t,\vec{F},\vec{f}) \vdash \bot} \left\{ \max_{D \in \pi} \left\{ \tilde{W}(D) \right\} \right\}$$

Clearly, for any clause D it holds that $\widetilde{W}(D) \leq W(D)$ (why?)

Recall that $k(C)$ is the minimum Hamming weight of an input $\vec{x} \in \{0,1\}^*$ s.t. $C(\vec{x}) = 1$

We argued before (though very handwavingly) that if we get our encoding of (C,t)-$\exists$SAT into CNF right, and if the sun, moon, and stars are properly aligned, then a resolution proof refuting the claim that (C,t)-$\exists$SAT is false should have to cycle through $|A|^{k(C)}$ possible counter-examples. Now we can make this precise.

MAIN TECHNICAL LEMMA (LEMMA 9)

Let $C(p_1, \ldots, p_n)$ monotone circuit, $A \subseteq \{0,1\}^m$, $F_1, \ldots, F_n : \{0,1\}^s \to A$, $f_1, \ldots, f_m : \{0,1\}^s \to [r]$ be r-surjective functions; f_i is possibly partial

m, r, s arbitrary | parameters | positive integer

Then

(a) If $k(C) \leq d_1(A)$, $\mathcal{L}_{\mathcal{P}}(\tau(C, A, \vec{F}, \vec{f}) \vdash \bot) = 2^{\Omega(s(k(C)+1))}$

(b) $\widetilde{W}(\tau(C, A, \vec{F}, \vec{f}) \vdash \bot) \geq \frac{r}{2} \min\{k(C), d_0(A)\}$

(c) $\mathcal{L}_{\mathcal{R}}(\tau(C, A, \vec{F}, \vec{f}) \vdash \bot) \geq \exp\left(\Omega\left(\frac{r^2}{s} \min\{k(C), d_0(A)\}\right)\right)$

Part (a) We just need to formalize the proof we did last lecture within the framework of tree-like resolution.

Let $k = k(C)$ and fix $\vec{\alpha} \in \{0,1\}^n$ s.t. $C(\vec{\alpha}) = 1$ and $wt(\vec{\alpha}) = k$. W.l.o.g. (because of symmetry) suppose $\vec{\alpha}_1 = \dots = \vec{\alpha}_k = 1$, $\vec{\alpha}_{k+1} = \dots = \vec{\alpha}_n = 0$, $\alpha = (\underbrace{1, \dots, 1}_{k}, \underbrace{0, \dots, 0}_{n-k})$

Consider any $\sigma^{(1)}, \dots, \sigma^{(k)} \in \{0,1\}^s$

We want to derive

$$\neg \vec{x}_1^{\sigma^{(1)}} \vee \dots \vee \neg \vec{x}_k^{\sigma^{(k)}} \qquad (1)$$

in tree-like resolution in length at most $|C| \cdot 2^{s+1}$. Then we can apply Observation 3 to derive contradiction in total length $\leq |C| \cdot 2^{s+1} \cdot 2^{ks+1}$

For every $\sigma^{(j)}$ we have $F_j(\sigma^{(j)}) =$ some vector in A, say $\vec{a}_I^{(j)}$ (to avoid notational clutter)

Given columns $\vec{a}_{i_1}^{(1)}, \dots, \vec{a}_{i_k}^{(k)}$, there is a row i in the matrix such that the first k positions are all 1. Fix this i. We want to derive

$$\neg \vec{x}_1^{\sigma^{(1)}} \vee \dots \vee \neg \vec{x}_k^{\sigma^{(k)}} \vee \neg \vec{y}_i^{\tau} \qquad (2)$$

for all $\tau \in \{0,1\}^s$. If we can do this in tree like resolution in length at most $|C|$, then we can get from (2) to (1) in length $|C| \, 2^{s+1}$ by an easy adaptation of Observation 3.

If $\tau \notin Dom(f_i)$, then we immediately get (2) by weakening from any axiom of type (i) so suppose $f(\tau) = c$.

Let V^1 = all nodes in C evaluating to 1 under $\vec{\alpha} = (1^k, 0^{n-k})$. Let V^1 sorted in topological order (wrt C) so that

$$V^1 = \{v_1 = p_1, v_2 = p_2, \ldots, v_k = p_k, v_{k+1}, v_{k+2}, \ldots, v_t = v_{out}\}$$

By reverse induction derive

$$\neg \vec{y}_i^{\tau} \vee \bigvee_{\ell=1}^{\mu} \bar{z}_{i,v_\ell}^{c} \qquad (3)$$

for $\mu = t, t-1, t-2, \ldots, k+1, k$.

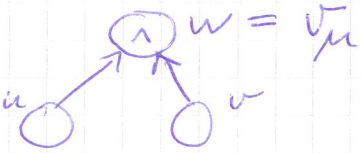
Base case

$$\neg \vec{y}_i^{\tau} \vee \bar{z}_{i,v_{out}}^{c} \in [col_i = c] \rightarrow \bar{z}_{i,v_{out}}^{c} \qquad (4)$$

is an axiom of type (iv). Use weakening to derive (3).

Inductive step

Suppose μ is an AND-gate.



Then

$$\neg \vec{y}_i^{\tau} \vee \bar{z}_{i,u}^{c} \vee \bar{z}_{i,v}^{c} \vee z_{i,w}^{c} \qquad (5)$$

is an axiom of type (iii) and u, v appear in V^1 with indices smaller than μ. Resolve away z_{i,v_μ}^{c}.

OR-gate is similar

Finally we get

$$\neg \vec{y}_i^{\,\tau} \vee \bigvee_{\ell=1}^{k} \bar{z}_{i,\ell}^{\,c}$$

Since $\forall\, \vec{a}^{(j)}$ $j = 1,\dots,k$ it holds that $\vec{a}^{(j)}_i = 1$ — that is how we chose i — we can resolve with axioms of type (ii)

Note that

$$\neg \vec{x}_j^{\,\sigma^{(j)}} \in \left[col_j = \vec{a}^{(j)} \right]$$

for all $j \in [k]$. This yields clauses of the form (2). Since we did one resolution step for every node in V^1, the length of the derivation of each clause (2) is at most $|C|$.

Part (6) Use a [BWOL]-style complexity measure approach. I.e., define $\mu : \{\text{clauses}\} \to \mathbb{N}$ such that

(i) μ (axiom) ≤ 1

(ii) μ subadditive $\mu(B \vee C) \leq \mu(B \vee x) + \mu(C \vee \bar{x})$

(iii) μ ($\perp$) large

(iv) Prove $\mu(D)$ medium-large $\Rightarrow$ D wide

Note that all axioms of $\tau(C, t, \vec{F}, \vec{f})$ refer to one specific row i. Let

$R_i = \{\text{all axioms speaking about row } i\}$

Let

$$\mu(D) = \min\{|I| : I \subseteq m, \bigcup_{i \in I} R_i \vdash D\}$$

By definition, (i) & (ii) immediate.

Want to prove: $\boxed{\mu(\mathcal{I}) > d_0(\mathcal{A})}$

Fix any $I \subseteq [m]$, $|I| \leq d_0(\mathcal{A})$

Need to find assignment satisfying

$$\bigcup_{i \in I} R_i$$

Pick $\vec{a}^* \in \mathcal{A}$ s.t. $\vec{a}^*_i = 0 \quad \forall i \in I$

(possible since $|I| \leq d_0(\mathcal{A})$)

Set $\vec{x}_j = \sigma$ s.t. σ is for all $j \in [n]$ / for some σ s.t. $F_j(\sigma) = \vec{a}$

(possible thanks to surjectivity).

Set $\vec{y}_i$ in any way that satisfies (i)

Set $z^c_{i,\vec{v}} = 0$ for all $i, \vec{v}, c$.

By design, ~~there are~~ all axioms (ii) will have $\vec{a} \neq \vec{a}^*$ and so are satisfied

Axioms (iii) & (iv) satisfied since all z-variables are set to false

This proves $\mu(\mathcal{I}) > d_0(\mathcal{A})$.

Hence, any resolution refutation of D $\tau(C, t, \vec{F}, \vec{f})$ contains a clause with

$$\frac{1}{2} d_0(t) < \mu(D) \leq d_0(t) \qquad (6)$$

We need to show this implies

$$\widetilde{W}(D) \geq \frac{r}{2} \min\{k(C), d_0(t)\} \qquad (7)$$

Fix a clause D satisfying (6)
Fix minimal $I \subseteq [m]$ s.t.

$$\bigvee_{i \in I} R_i \vdash D$$

We are done if for every $i \in I$ one of the following holds

1. D contains $\geq r$ variables in $\{y_i^v \mid v \in [s]\}$
2. For every control $c \in [r]$ D contains at least one variable among $\{z_{iv}^c \mid v \text{ node in } C\}$

(1) contributes $\geq r$ to $W_y(D)$. (2) contributes $\geq r$ to $r \cdot \min_{c \in [r]} W_c(D)$.

Suppose for $i_0 \in I$ neither (1) nor (2) holds. Then $\exists\, c_0 \in [r]$ s.t. no variable $z_{i_0, v}^{c_0}$ appears in D. Fix an assignment α that satisfies

$$\bigvee_{i \in I \setminus \{i_0\}} R_i \text{ and falsifies } D$$

(which exists by minimality)

Let J_0 consist of those $j \in [m]$ for which D contains at least r variables from $\{x_j^v \mid v \in [s]\}$. If $|J_0| \geq k(C)$ then (7) certainly follows and we are done.

If this is not the case, though, we will show how to flip values in x to satisfy all of $\bigvee_{i \in I} R_i$ without changing that D is falsified. This will yield a contradiction.

Use again that we have $\vec{a}^* \in A$ s.t. $\vec{a}_i^* = 0$ for all $i \in I_0$.

STEP 1 For every $j \notin J_0$, change values of $\{x_j^v \mid v \in [s]\} \setminus \text{Vars}(D)$ so that

$$F_j(\vec{x}_j) = \vec{a}^*$$

This uses r-surjectivity.

STEP 2 Change values of $\{y_{i_0}^v \mid v \in [s]\} \setminus \text{Vars}(D)$ so that $f_{i_0}(\vec{y}_{i_0}) = c_0$

This also uses r-surjectivity

STEP 3 Reassign every $z_{i_0, v}^{c_0}$ to value computed by node v when C is fed the characteristic vector of J_0 (i.e. 1 in coordinate $j \in [n]$ iff $j \in J_0$)

$z_{i_0, p_j}^{c_0} = 1$ for $j \in J_0$, but $z_{i_0, v_{out}}^{c_0} = 0$ since $|J_0| < k(C)$

Let α' be the assignment altered according to steps 1, 2, & 3.

Claim α' satisfies $\bigvee_{i \in I} R_i$ but falsifies D.

Proof The latter part is clear — we never touched variables in D.

R_{i_0}-axioms are OK — we know position 1 can only appear in J_0, and circuit copy C_{i_0} is evaluated correctly even assuming that we have all 1s there.

For $i \in I \setminus \{i_0\}$, axioms (i), (iii) & (iv) are OK — we never touched these variables.

What about (ii)? Intuitively, we should be OK since we are just flipping inputs from 1 to 0.

Case analysis

(a) $j \in J_0$, $i \neq i_0$: No variables changed — OK

(b) $j \notin J_0$: The chosen column might have changed but we have just zeroed out 1s. So if $z_{i,j} = 0$ before, then that's because $\bar{\alpha}$ chose a vector $F(\vec{x}_j) = \vec{a}$ with $\vec{a}_i = 0$, and this holds for $\vec{a}^*$ as well.

c) $j \in J_0$, $i = i_0$: Such axioms get satisfied during step 2, since C_{i_0} computes correctly on the characteristic vector of J_0.

We have established (7), and part (6) follows.

Part (c) We want to hit $\tau(C, t, \vec{F}, \vec{f})$ with a random restriction that will kill all wide clauses in a short refutation.

By the r-surjectivity, however, we get back some instance

$$\tau(C, t, \vec{F}', \vec{f}')$$

for which we have a width lower bound, and this shows that the refutation we started with cannot have been too short. This will have to wait till next time, though...