Proof Complexity

Proof system for language $L \subseteq \{0,1\}^*$ is a TM P that on input $\langle x, \pi \rangle$ runs in time $(|x| + |\pi|)^{O(1)}$ s.t.

$$x \in L \iff \exists \pi : P(x, \pi) = 1$$

possible: $|\pi| \ggg |x|$

Propositional PS: Above for $L = \text{UNSAT}$

(or $L = \text{TAUT} = \{ \text{DNF } \varphi : \forall x : \varphi(x) = 1 \}$)

P **poly-bounded** $\iff$ $\forall x \in \text{UNSAT}$ $\exists \pi, |\pi| \leq |x|^{O(1)} : P(x, \pi) = 1$

coNP-complete

Claim: $NP = coNP$ iff

$\exists$ poly-bounded proof sys. $\blacksquare$

Belief: $\nexists$ p-bounded PS. $(NP \neq coNP)$

"**Cook's programme**": Prove that increasingly strong proof systems are not p-bounded

---

Resolution

Let $\varphi \in \text{UNSAT}$.

Res refutation of φ is $\pi = (C_1, \dots, C_s)$ where

- $C_s = \emptyset$
- $\forall i \in [s]$ either
  - C_i clause of φ

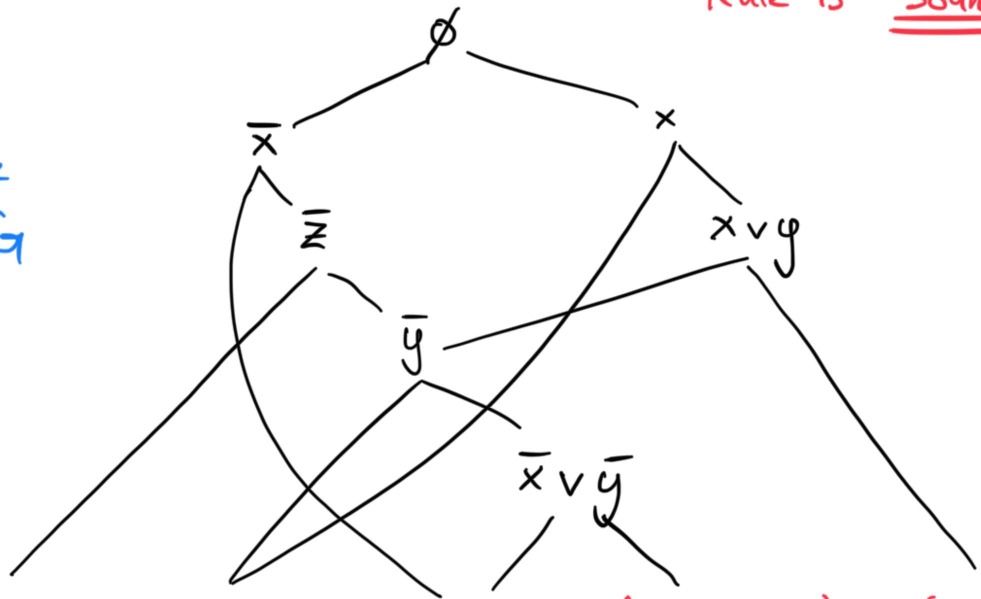
– C_i obtained from $C_j, C_{j'}$, $j, j' < i$ by rule

$$\frac{C_j = (\overset{\text{clause}}{A} \vee \overset{\text{var}}{x_k}) \qquad (B \vee \overline{x_k}) = C_{j'}}{(A \vee B) = C_i}$$

Resolution Rule

Rule is <u>sound</u>

Ex: Visualise as DAG



$\varphi = \; (y \vee \overline{z}) \quad (x \vee \overline{y}) \quad (z \vee \overline{x}) \quad (\overline{x} \vee \overline{y} \vee \overline{z}) \quad (x \vee y \vee z)$

Thm [Haken 85]: Res is not p-bounded. In particular Pigeonhole Principle requires Res-ref size $2^{\Omega(n)}$

Def π is tree-like if underlying DAG is tree

Def $\underset{n \text{ vars}}{\varphi} \in \text{UNSAT}$ defines $\text{Search}(\varphi)$:

<u>Input:</u> $x \in \{0,1\}^n$

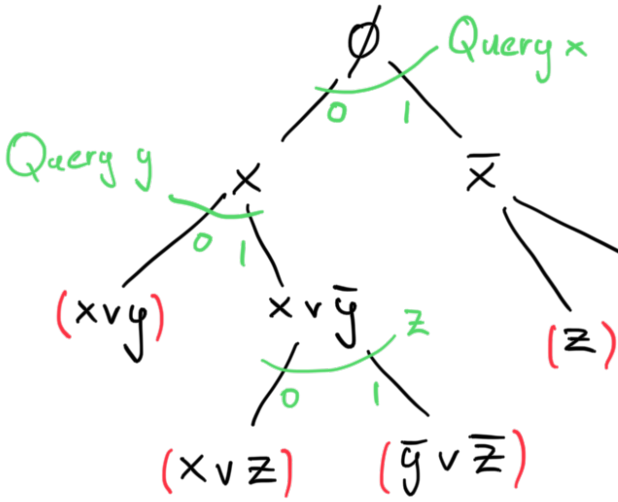
<u>Output:</u> Any clause C of φ with $C(x) = 0$

Lemma Tree-like ref π of φ

|||

Decision tree solving Search(φ)

" $\Rightarrow$ "



DT: Trace path down proof.

Invariant: current clause falsified

Q: How to prove l.b. on tree $\underline{\text{size}}$?

\# nodes

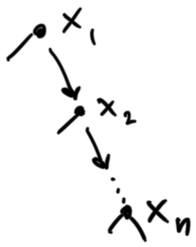
$$\text{depth} \le \text{size} \le 2^{\text{depth}}$$

today

last week

Ex: $\varphi = (x_1) \wedge (x_1 \to x_2) \wedge \cdots \wedge (x_{n-1} \to x_n) \wedge (\bar{x}_2)$

$x = 1111\boxed{10}0011\boxed{10}00$ 2 sols



depth $= n$
size $= O(n)$

depth $\log n$
size $O(n)$

Tree-vs-Adversary game for Search(φ)

In each round:

- Tree queries x_i
- Adv either chooses value of x_i

  $\boxed{\text{or}}$ lets Tree choose it $\leftarrow$ Adv scores point

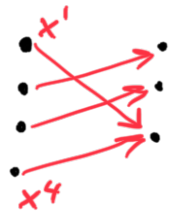
Game ends when Tree solves Search(φ)

Lemma: $\exists$ Adv that can score $\geq r$ points
$\Rightarrow$ DT size $\geq 2^r$ [Exercise]

Def PHP_n^{n+1} formula (binary version)

- $(n+1)\log n$ variables: $\forall i \in [n+1]$

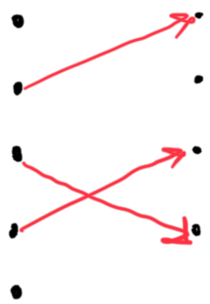
  $x^i \in \{0,1\}^{\log n} \equiv [n]$



- CNF: $\bigwedge_{i \neq j} \underbrace{[x^i \neq x^j]}_{\text{width } \log n \Rightarrow \text{poly}(n)\text{-size CNF}}$

Thm: PHP_n^{n+1} requires tree-like Res size $\geq 2^{\Omega(n)}$

Proof: Give Adv strategy scoring $\Omega(n)$ points



depth $\geq n$: easy

score $\geq \frac{n}{2}$: let tree choose one bit of each x^i (still $n/2$ possible holes)