Week 9

Announcements:

- HW3 posted
- Test 2 grades released today or tomorrow

Week 9

Today:

(1) Complexity of Approximation Algs, PCP Theorem

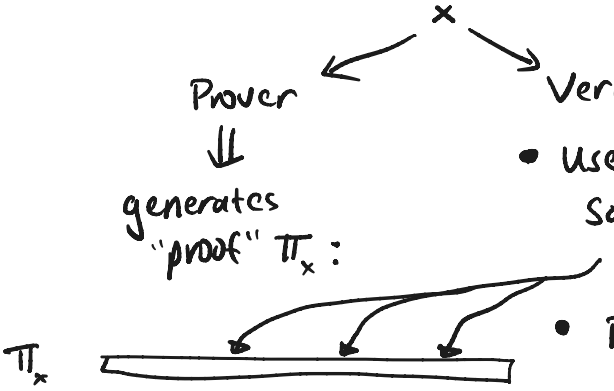
(2) Intro Circuit Complexity

PCP theorem

PCP theorem is a "robust" characterization of the class NP.

Idea: comes from Interactive Proofs

Algs/protocols are one-round, and we want verifier to use few random bits and to probe provers proof at only a few locations



Verifier

- uses random bits to query some c locations of π ← $O(\log n)$
- Prover sends values of π at these locations ← $q = O(1)$
- Verifier accepts or rejects

More formally

Verifier $V(x, \pi, b_1 \ldots b_{r(n)})$ is probabilistic and runs in time polynomial in $|x|$



V is $(r(n), q(n))$ - restricted if:

1. V has access to $O(r(n))$ random bits $b_1 \ldots b_{r(n)}$

2. $\forall x \in \{0,1\}^n$, $\forall \pi$,
   $V(x, \pi, \vec{b})$ makes at most $O(q(n))$ queries into π

Defn $L \in PCP[r(n), q(n)]$ if $\exists$ an $(r(n), q(n))$-restricted verifier V such that:

(1) completeness: $x \in L \Rightarrow \exists \pi$ s.t. $\Pr_{\vec{b}}[V(x, \pi, \vec{b}) = 1] = 1$

(2) Soundness: $x \notin L \Rightarrow \forall \pi \Pr_{\vec{b}}[V(x, \pi, \vec{b}) = 1] \leq \frac{1}{2}$

$\frac{1}{2}$ is soundness error

Theorem (PCP Theorem) $NP = PCP[\log n, 1]$

Note: can make Soundness $0 < \varepsilon < \frac{1}{2}$

(but number of queries increases)

Compare to defn of RP:

Verifier $V(x, \vec{u})$ — V run in time $\text{poly}(|x|)$, $|u|$ is at most $\text{poly}(|x|)$

$x \in L \Rightarrow \forall u \; [V(x, \vec{u}) = 1]$

$x \notin L \Rightarrow \Pr_{\vec{u}} [V(x, \vec{u}) = 1] \leq \frac{1}{2}$

PCP $(\log n, 1)$: 2 sources of nondet:

$\vec{r}, \Pi$ $\quad |\Pi| \leq \text{poly}(|x|)$

$\vec{r}$ length $O(\log n)$

* V can only look at $O(1)$ locations of Π

Theorem (PCP Theorem) $NP = PCP[\log n, 1]$

Note: can make Soundness ε:

$\frac{1}{2} \xrightarrow{\text{to}} \varepsilon$: V': Run Verifier $t = \log_2(\frac{1}{\varepsilon})$ times on the same proof π using independent random strings; accept iff every run accepts.

then $x \notin L \Rightarrow \forall \pi \Pr[V' \text{ accepts}] \leq 2^{-t} \leq \varepsilon$

$\varepsilon \rightarrow \frac{1}{2}$: If $\varepsilon \leq \frac{1}{2}$ it already implies $\frac{1}{2}$ bound

If $\frac{1}{2} < \varepsilon < 1$: repeat constant number of times until $\varepsilon^t \leq \frac{1}{2}$

* Note: this increases q = # queries

$q^{\text{new}} = t \cdot q$

Theorem (PCP Theorem) $NP = PCP[\log n, 1]$

Comments

1. Since number of random bits used by Verifier is $O(\log n)$, V can be derandomized: simulate $V(x, \pi, *)$ on all $2^{r(n)} = \text{poly}(n)$ random bits and accept iff all runs accept.
   So one direction follows from this ($PCP[\log n, 1] \subseteq NP$)
2. Other direction: $NP \subseteq PCP[\log n, 1]$ hard. Breakthrough result!

   Applications:
   - Hardness of approx for almost all important NP approx problems
   - Crypto - SNARKs
   - Error correcting codes
   - Complexity theory - PRGs
   - Quantum analogs

Theorem (PCP Theorem) $NP = PCP[\log n, 1]$

Comments

Proof of PCP theorem highly non trivial.

Uses:
- Idea of arithmetizing (as we saw for #SAT $\in$ IP)
- property testing

Dinur

Equivalent Form of PCP Theorem

(Connecting it to hardness of approximation)

Defn (Max-KCSP)

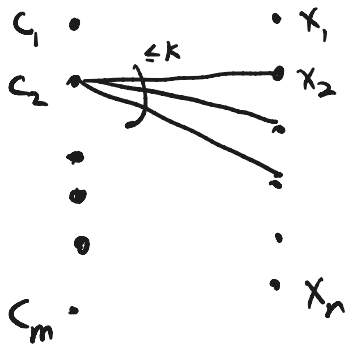
A Boolean $KCSP_n$ instance consists of Boolean variables $x_1, \ldots, x_n$ and a list of constraints $C_1 \ldots C_m$.

Each C_i is an arbitrary boolean predicate on a set $S_i \subseteq \{x_1, \ldots, x_n\}$, $|S_i| \leq K$

Examples: $x_1 \vee \neg x_2 \vee x_3$

$x_2 \oplus x_3 \oplus x_5 = 1$

Constraint graph:



MaxkCSP: on input $C = \{C_1, \ldots, C_m\}$ each C_i a kCSP over $x_1, \ldots, x_n$

find assignment $a \in \{0,1\}^n$ satisfying max fraction of constraints

Theorem (Equivalent formulation of PCP Theorem)
The following 2 statements are equivalent:

(1) [$NP = PCP(\log n, 1)$] $\forall L \in NP$, $L \in PCP(\log n, 1)$ ← recall: $O(\log n)$ random bits, $O(1)$ queries

(2) [Hardness of Approximation of Max kCSP]
$\exists K \in \mathbb{N}$, $0 < \epsilon < 1$ s.t. $\forall L \in NP$ there is a polytime reduction mapping x to a kCSP instance I_x satisfying:

$x \in L \Rightarrow OPT(I_x) = 1$

$x \notin L \Rightarrow OPT(I_x) \leq \epsilon$

Special case of Max 3CSP is

Max 3SAT : Input is a CNF formula $C_1 \ldots C_m$ over $x_1 \ldots x_n$

Find $\alpha \in \{0,1\}^n$ that satisfies the maximal possible number of clauses.

Proof $\boxed{(1) \Rightarrow (2)}$: $\exists k,\ 0 < \varepsilon < 1$ such that:

$L \in PCP[\log n, 1] \Rightarrow \exists$ ptime reduction $x \to I_x$ satisfying

$x \in L \Rightarrow OPT(I_x) = 1$

$x \notin L \Rightarrow OPT(I_x) \leq \varepsilon$

Proof

on input x, construct kCSP instance I_x as follows:

Let $R = c\log n$ be number of random bits used by V on x, $|x| = n$

Let q = # queries made by V

$\therefore$ the length of Provers proof $|\Pi|$ on x is $\leq q \cdot n^c$

I_x: Variables: $y_1, \ldots, y_{|\Pi|}$; constraints $\{C_r \mid r \in \{0,1\}^R\}$

$C_r(\underbrace{y_{i_1} \ldots y_{i_q}}_{q \text{ locations queried by } V^{\Pi}(x,r)}) = 1$ iff V accepts x with randomness r and query answers $y_{i_1} \ldots y_{i_q}$

For all $\alpha \in \{0,1\}^{|\pi|}$ (assignments to $y_1 \cdots y_{|\pi|}$)

$$\Pr_r\left[V^{\pi=\alpha}(x,r)=1\right] = \text{fraction of constraints of } \mathcal{I}_x \text{ satisfied by } \alpha$$

$x \in L \Rightarrow \exists\, \alpha$ satisfying all constraints $\Rightarrow OPT(\mathcal{I}_x) = 1$
(by completeness of V)

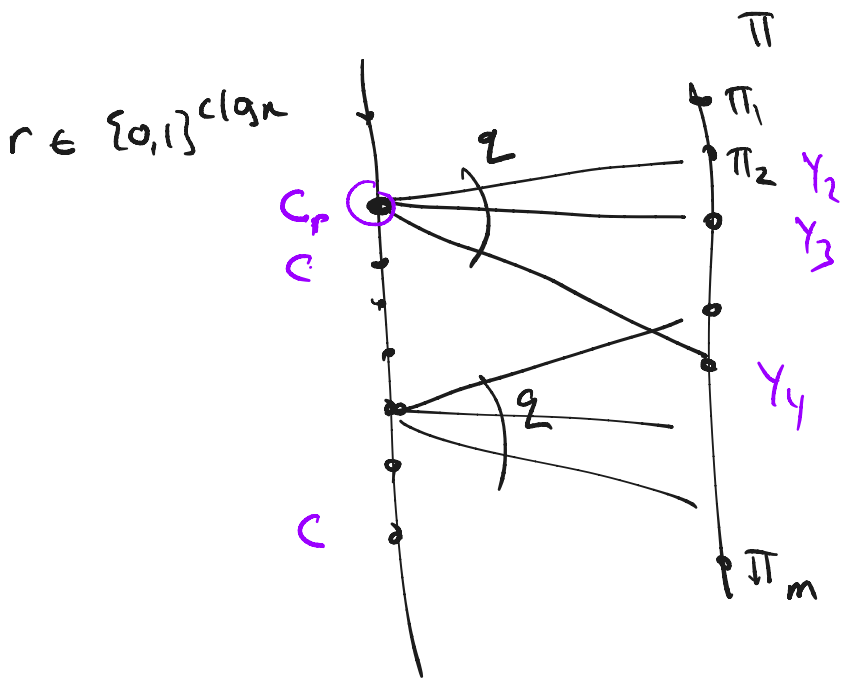
$x \notin L \Rightarrow \Pr_r\left[V^{\pi}(x,r)=1\right] \leq \varepsilon$

$\therefore \forall \alpha$, fraction of constraints of $\mathcal{I}_x$ satisfied by $\alpha \leq \varepsilon$

$\therefore OPT(\mathcal{I}_x) \leq \varepsilon$



Verifier on x



f_x

$\Pi = Y_1 \cdots Y_m$

$C_r(Y_2 Y_3 Y_4) =$

Same as

$V(x, r_\cdot)$

Proof (2) → (1) : Let $L \in NP$ and assume there is a ptime reduction f from $x \longrightarrow I_x$ (I_x instance of Max kCSP) satisfying:

$$x \in L \Rightarrow OPT(I_x) = 1 \; ; \; x \notin L \Rightarrow OPT(I_x) \leq \varepsilon$$

then $L \in PCP[\log n, 1]$

PCP Protocol

1. Verifier on x, $|x| = n$: computes $I_x = \overbrace{C_1 \wedge \dots \wedge C_m}^{\text{kCSP constraints over } n' \text{ vars}}$

   since reduction ptime, m, n' are poly(n).

2. Prover sends Π (supposed to be assignment to vars of I_x)

3. Verifier: chooses random constraint C_i, reads from Π the $\leq k$ bits of Π in C_i; accept iff constraint C_i is satisfied

   # random bits $= \log m = O(\log n)$ since m is poly(n)

Proof (2) → (1) : Let $L \in NP$ and assume there is a ptime reduction Π from $x \to I_x$ (I_x instance of Max kCSP) satisfying:

$x \in L \Rightarrow OPT(I_x) = 1$; $x \notin L \Rightarrow OPT(I_x) \leq \varepsilon$

then $L \in PCP[\log n, 1]$

PCP Protocol

1. Verifier on x, $|x| = n$: computes $I_x = \overbrace{C_1 \wedge \dots \wedge C_m}^{\text{kCSP constraints over } n' \text{ vars}}$

   since reduction ptime, m, n' are poly(n).

2. Prover sends Π (supposed to be assignment to vars of I_x)

3. Verifier: chooses random constraint C_i, reads from Π the $\leq k$ bits of Π in C_i; accept iff constraint C_i is satisfied

   # random bits = $\log m = O(\log n)$ since m is poly(n)

Correctness

For every Π: $Pr[V^{\Pi}(x) \text{ accepts}]$ = fraction of constraints of C satisfied by Π

$x \in L \Rightarrow OPT[I_x] = 1$ so $\exists \Pi$ that is accepted with prob 1

$x \notin L \Rightarrow OPT[I_x] \leq \varepsilon$ so <u>every</u> Π accepted with prob $\leq \varepsilon$

This shows PCP [log n, 1] is equivalent to

max-kCSP is NP-hard to approximate with soundness ε.

Translating hardness of approximation for max kSAT to hardness of approx for other NP-optimization problems.

Example $\overset{\max}{k}$SAT : $C = C_1 \wedge \cdots \wedge C_m$

each C_i contains $\leq k$ literals

UB:

① Randomized alg : Pick random $\alpha \in \{0,1\}^n$

$$\mathop{\mathbb{E}}_{\alpha}[\#\text{ clauses of } C \text{ satisfied by } \alpha] \geq \left(1 - \tfrac{1}{2^k}\right) m$$

← For $k=3$ this is $\frac{7}{8} m$

Example $\max$ KSAT : $C = C_1 \wedge \cdots \wedge C_m$

each C_i contains $\leq k$ literals

UB:

① Randomized alg : Pick random $\alpha \in \{0,1\}^n$

$\mathbb{E}_{\alpha}$ [# clauses of C satisfied by α]

$\geq \left(1 - \frac{1}{2^k}\right) m$ ← For $k=3$ this is $\frac{7}{8} m$

② Can derandomize by method of "conditional expectations" to get a deterministic polytime alg. A s.t.

for every KSAT Formula C over $x_1 \cdots x_n$

$A(C)$ outputs an assignment satisfying $\geq \frac{7}{8}$ fraction of clauses

$\therefore$ A is a $\frac{7}{8}$-approx alg for max KSAT

We saw Max3SAT has a ptime $\frac{1}{8}$ approx alg. (max KSAT : $1-\frac{1}{2^k}$ approx)

Can we do better?

Hardness of Approx of Max KSAT from PCP [$\log n$, 1]:

Let I be a max KCSP instance with m constraints.

construct equivalent KCNF formula F over some vars:

Convert each KCSP constraint C_i into CNF

$C_i \rightarrow$ at most 2^k clauses, each of width $\leq k$

Claim # clauses of F falsified by α = # constraints of I violated by α

$\therefore$ If $OPT(I)=1 \Rightarrow OPT(F)=1$

If $OPT(I) \leq \varepsilon \Rightarrow OPT(F) \leq \frac{(1-\varepsilon)m}{2^k m} = \frac{(1-\varepsilon)}{2^k}$

for $k=3$ this is $1-\frac{1-\varepsilon}{8} > \frac{7}{8}$

UB for Max 3SAT : $7/8$

Better Hardness of Approx for MaxKSAT :

For Max3SAT, the generic PCP theorem implies hardness of approx ε, for some $\varepsilon < 7/8$.

Similarly we get hardness of approx for MaxKSAT for some ε_k but not $\varepsilon_k = 1 - 1/2^k$

Note We can't simply amplify error in PCP theorem to get $\varepsilon = 7/8$ for max3SAT since amplification increases q (queries) $\Rightarrow$ therefore increases k (clause size)

Instead: Need a better PCP theorem that directly implies $1 - 1/2^k$ hardness of approx for maxKSAT, matching UB.

Was shown in subsequent work!

Where things stand NOW

For most NP-hard optimization problems, improved PCP thm gave optimal hardness of approximation matching best upper bounds:

Problem	Polytime Approx alg	Hardness of approx
MaxkSAT	$1 - \frac{1}{2^k}$	$1 - \frac{1}{2^k}$
Max 3XOR	$\frac{1}{2}$	$\frac{1}{2} + \varepsilon \quad \forall \varepsilon < 1$
chromatic Number	n colors (trivial)	$n^{1-\varepsilon} \quad \forall \varepsilon < 1$

Some Problems required a stronger form of PCP known as "UGC"
unique games conjecture

Problem	Polytime Approx Alg	Hardness of approx from PCP	Hardness of approx under UGC
Max-Cut	.878	.94	Nearly .878
Min Vertex cover	$2-o(1)$ greedy alg	$\sqrt{2}$	$2-\varepsilon$

Some Problems required a stronger form of PCP known as "UGC"
unique games conjecture

Problem	Polytime Approx Alg	Hardness of approx from PCP	Hardness of approx under UGC
Max-Cut	.878	.94	Nearly .878
Min Vertex cover	$2-o(1)$ greedy alg	$\sqrt{2}$	$2-\varepsilon$

* This month OpenAI released proof of UGC
(proof still not fully verified)

Last Topic: Circuit Complexity and Comm. Complexity

(1) Boolean Circuits over $\vee, \wedge, \neg$

(2) $L \in P \Rightarrow \exists k > 0$ $\forall n$ suff large

$\exists$ circuit C_n of size $\leq O(n^k)$ s.t.

$\forall x \in \{0,1\}^n$ $C_n(x) = 1 \iff x \in L$

$\Big\}$ $L \in P/poly$

(3) (Converse of 2 is false): $\exists L \in P/poly$ but $L \notin P$

But can get equivolence by adding "p-time uniform":

$L \in P \iff L$ is decided by a P-uniform family of polysize circuits

(4) Restricted circuit classes:

$$AC^0 \subseteq AC^0[2] \subseteq NC^1 \subseteq NC \subseteq P/poly$$

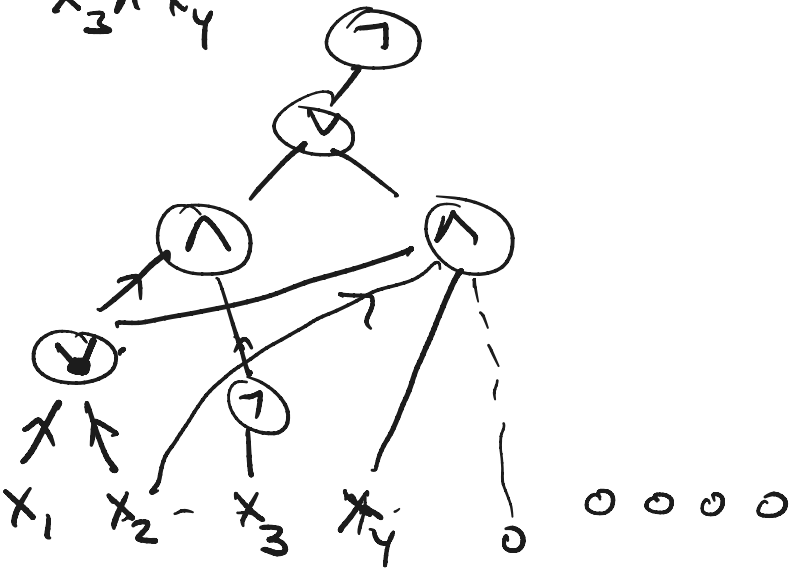
(1) Boolean Circuits, P/poly

Boolean circuit over $\vee, \wedge, \neg$

$x_1 \vee x_2$ $\quad$ $x_3 \wedge x_4$

Circuit over $x_1 \cdots x_n$:

$n=4$



Size of C : ① # of $\neg, \vee, \wedge$ gates in C

② total # of edges in C

A circuit family recognizes a language $L \subseteq \{0,1\}^*$

$\mathcal{C} =$

$\{C_0$: 1 or 0

C_1: one input x_1

C_2: over $x_1 x_2$

C_3: over $x_1 x_2 x_3$

⋮

$\}$

A language L is recognized by a circuit family $\mathcal{C} = \{C_0, C_1, \ldots\}$ if

$\forall n \geq 0$ and $\forall \alpha \in \{0,1\}^n$

$C_n(\alpha) = 1$ iff $\alpha \in L$

Fact for every n, and every $f : \{0,1\}^n \to \{0,1\}$

there is a circuit C_f that computes f

DNF: Big disjunction of conjunctions

$$x_1 \bar{x}_2 x_3 \vee \bar{x}_1 \bar{x}_2 \vee x_1 x_2 x_3$$

$x_1 \wedge \bar{x}_2 \wedge x_3$

x	$f(x)$
000	0
001	1
010	1
011	0
100	1
101	1
110	1
111	0

write each 1 input as a term:

$\bar{x}_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 x_2 \bar{x}_3$

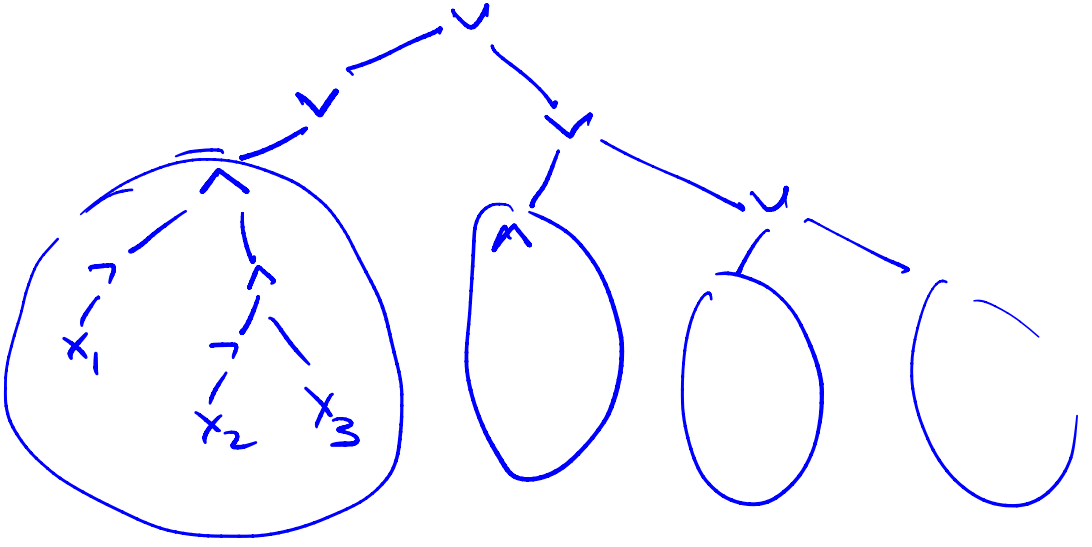
$$x_1 \bar{x}_2 x_3 \vee \bar{x}_1 \bar{x}_2 \vee x_1 x_2 x_3$$

$x_1 \wedge \bar{x}_2 \wedge x_3$

x	f(x)
000	0
001	1
010	1
011	1
100	0
101	1
110	1
111	0

write each 1 input as a term:

$\bar{x}_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 x_2 \bar{x}_3$



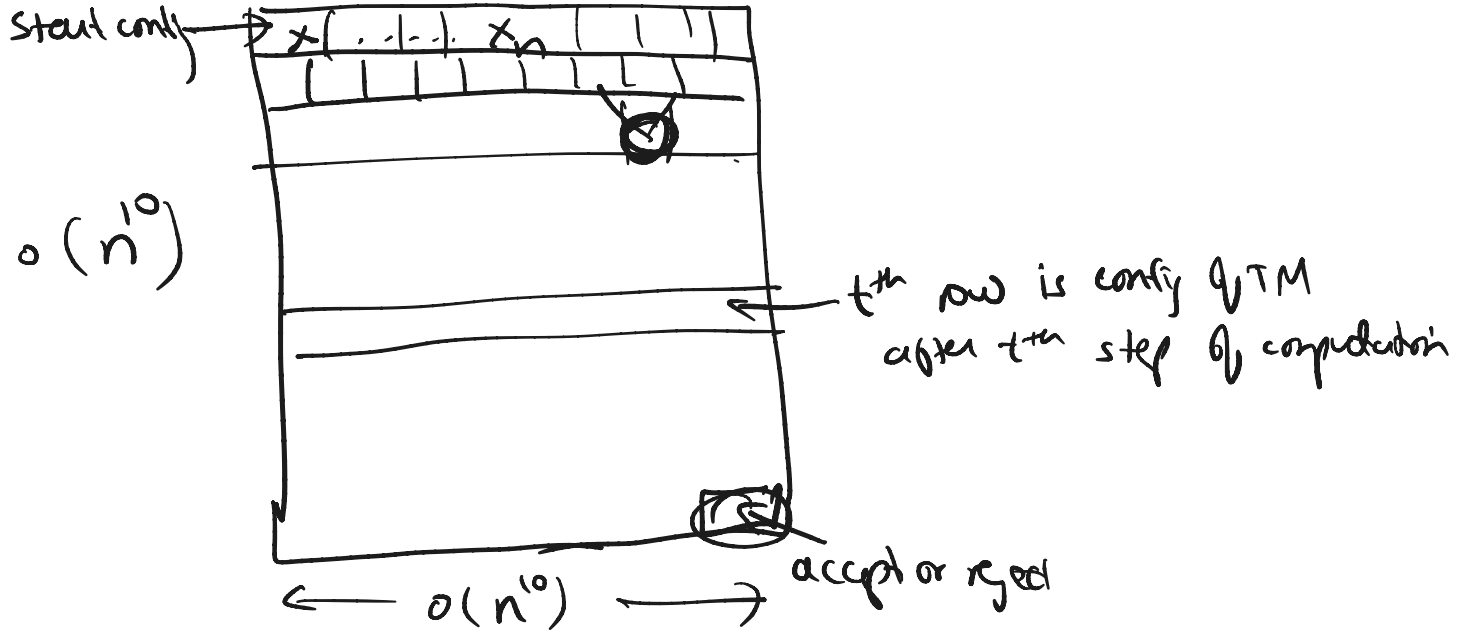
Defn $L \in$ "P/poly" If there is a circuit family $C = \{C_0, C_1, \ldots\}$ that recognizes L

and $\exists c$ s.t. $\forall n$ sufficiently large $|C_n| \leq n^c$

} Circuits have polynomial size in n

(2) $L \in P \Rightarrow L \in P/poly$ Theorem

"Tableau" of a time n^{10} TM recognizing L on all inputs of length n:



(3) Not every $L \in P/poly$ is in P
(But <u>every</u> L has exponential-size circuits)

<u>Intuition</u>: no since for circuits, we can have an entirely different circuit/algorithm for each n.

Consider $L = \{ 1^n \mid n$ in binary ~~is~~ describes/encodes a TM M s.t. M ~~does~~ halts on input $x=0 \}$

This L is undecidable (def. not in P)

$C_n(1^n) = \begin{cases} 1 & \text{if } 1^n \in L \\ 0 & \text{ow} \end{cases}$

$C_n(\text{any other string}) = 0$

To summarize:

every $L \in P$ is in $P/poly$
but converse is not true.

One way to try to prove $P \neq NP$ is to
show any circuit family $C = \{C_0, C_1, \ldots\}$
of polysize cannot recognize 3SAT.

(3) P-time uniform P/poly

We can restrict circuit families to be "polytime" uniform and then this is equiv. to P.

A P/poly circuit family $C = \{C_0, C_1, \ldots\}$ is p-time uniform if there is a Ptime TM M that on input 1^n (n in unary) ~~that~~ ~~or~~ s.t. $M(1^n)$ outputs C_n

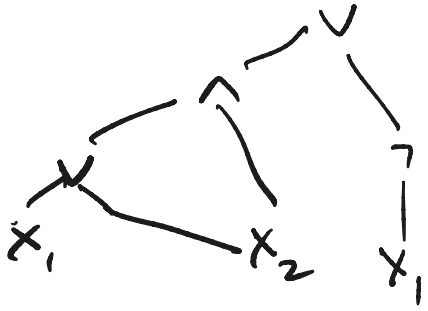
(4) Circuit Classes and Known Lower Bounds

$$AC^0 \subseteq AC^0[2] \subseteq NC_1 \subseteq NC \subseteq P/poly$$

NC_1: polysize formulas $\equiv NC_1$ of depth $O(\log n)$

NC: polysize ckts with depth $(\log n)^{O(1)}$

$P/poly$: polysize circuit family



It is a circuit where fanout of every non leaf node is 1