

AGENDA

1. Reductions + NP-completeness
2. DETOUR: Circuits + Non Uniform computation
3. Back to NP-completeness (Cook-Levin)
4. Randomized computation
 - BPP $\subseteq$ P/poly
 - Example: PIT $\leftarrow$ didn't get to this.

Disclaimer: I assume you've seen Cook-Levin before.

REDUCTIONS + NP-COMPLETENESS

Recall:

$$NP = \bigcup_{c \geq 1} NTIME(n^c) = \left\{ L : \begin{array}{l} \exists \text{ polytime DTM } M \text{ and poly } p \text{ s.t. } \forall n, \forall x \in \Sigma^n \\ x \in L \iff \exists y \in \Sigma^{p(n)}, M(x, y) = 1 \end{array} \right\}$$

y is called the "certificate" or "witness"

We want to compare hardness of things: reductions are our tool for that.

DEF L' is **POLYNOMIAL-TIME REDUCIBLE** to L ($L' \leq_p L$) if there is a polynomial-time computable $f: \Sigma^* \rightarrow \Sigma^*$ s.t.

$$x \in L' \iff f(x) \in L$$

$L' \leq_p L$ means " L is at least as hard as L' ".
since "Run alg for L on $f(x)$ " is an algorithm for L' .

DEF L is **NP-HARD** if $L' \leq_p L \forall L' \in NP$
 L is **NP-COMPLETE** if L is NP-hard and $L \in NP$.

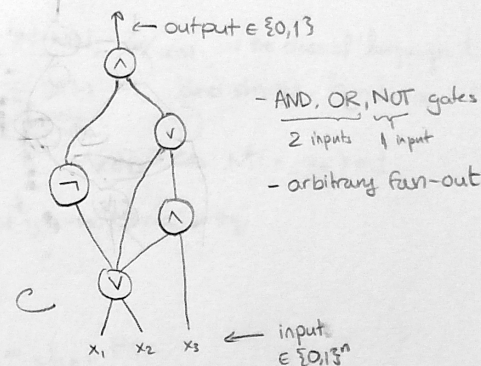
- to prove $P = NP$, just find a polytime alg for an NP-complete problem
- to indicate a problem is hard, show it's NP-hard.
- Notice that $\leq_p$ is transitive, so to show something is NP-hard, it suffices to reduce an NP-complete problem to it.

QUESTION Are there any $\uparrow$ NP-complete languages?
non-convicted

Convicted: $\{ \langle M, x, 1^t, 1^m \rangle : \exists y \in \Sigma^m, M(x, y) \text{ accepts in } t \text{ steps} \}$.

DETOUR: CIRCUITS and NON-UNIFORMITY.

DEF (by example) A **circuit** looks like this:



The **SIZE** of C is $|C| = \# \text{ gates}$.

We want to talk about asymptotics, but unlike TMs, circuits have a fixed size.
So we talk about **FAMILIES** $\{C_n\}_{n \in \mathbb{N}}$ of circuits.

DEF For $s: \mathbb{N} \rightarrow \mathbb{N}$, a circuit family $\{C_n\}_{n \in \mathbb{N}}$ has **SIZE** $s(n)$ if $\forall n \in \mathbb{N}, |C_n| \leq s(n)$.
The class $SIZE(s(n))$ is the class of languages $L \subseteq \{0,1\}^*$ s.t. $\exists \{C_n\}$ so that $\forall n, \forall x \in \{0,1\}^n$,
 $x \in L \iff C_n(x) = 1$

Note: circuits are powerful! In particular, nothing is undecidable for circuits, since you can always build a lookuptable of size 2^n .
With that in mind, it's more interesting to look at smaller circuits.

DEF $P/poly = \bigcup_{c \geq 1} SIZE(n^c)$

Another way to capture the "non-uniformity" of circuits (ie, the power to hard-code) is through advice

(3)

DEF

for $a: \mathbb{N} \rightarrow \mathbb{N}$, $P/a(n)$ is the class of languages L s.t.
 $\exists$ DTM M , and strings $\{y_n\}_{n \in \mathbb{N}}$ s.t. $|y_n| \leq a(n)$, $\forall x \in \Sigma^n$
 $x \in L \iff M(x, y_n) = 1$

↑ poly time
 Say that y is the "advice" string.

ALT. DEF

$$P/poly = \bigcup_{c \geq 1} P/n^c$$

Thm. The two defs of $P/poly$ are the same.

ATTN. We will use this proof twice more.

$$(1) \bigcup_{c \geq 1} SIZE(n^c) \subseteq \bigcup_{c \geq 1} P/n^c \quad (\text{easy})$$

For $L \in SIZE(n^c)$, choose $y_n = \langle C_n \rangle$, and $M(x, \langle C_n \rangle) = C_n(x)$.
 decided by $\{C_n\}$

$$(2) \bigcup_{c \geq 1} P/n^c \subseteq SIZE(n^c)$$

Say $L \in P/n^c$ is decided by $M(x, y_n)$.

Need to show we can simulate $M(\cdot, y_n)$ by a polysized circuit.

Let Z_t = "snapshot" of M at time t
 = (state, tape, head pos.) at time t .

Notice Z_t is determined from $Z_1, \dots, Z_{t-1}, x, y_n$ by a $f(n)$ -sized circuit.

So the final state is also determined by a polysized circuit.

NOTE. The proof of (2) is constructive and efficient:
 given a polytime TM M , we can efficiently find a polysized circuit that simulates it.

BACK to NP-COMPLETENESS

(4)

- Are there any NP-complete problems?

Thm (Cook-Levin)

3SAT is NP-complete

Lemma

CKT-SAT = $\{\langle C \rangle : \exists x, C(x) = 1\}$
 is NP-hard.

Pf. (Sketch)

Let $L \in NP$, so $x \in L \iff M(x, y) = 1$.

By the proof we just did, there's a polysized circuit C_n so that
 $C_n(x, y) = 1 \iff M(x, y) = 1$.

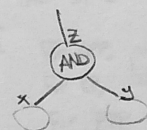
Then $x \mapsto \langle C_n(x, \cdot) \rangle$ is a polynomial time reduction from L to CKT-SAT.

So $L \leq_p CKT-SAT \forall L \in NP$.

Lemma

CKT-SAT $\leq_p$ 3SAT

Pf. Given a circuit C , turn it into a 3CNF as follows.
 (Sketch) There's a variable for each gate. For each AND gate, add the clauses



$$(\bar{z} \vee x \vee y) \wedge (\bar{z} \vee x \vee \bar{y}) \wedge (\bar{z} \vee \bar{x} \vee y) \wedge (z \vee \bar{x} \vee \bar{y})$$

which is $\iff z = x \wedge y$.

Similarly for each OR, NOT gate.

Get a formula φ s.t. $\exists x, C(x) = 1 \iff \exists x, \varphi(x) = 1$.

RANDOMIZED COMPLEXITY

(5)

DEF A RANDOMIZED TM takes an additional input r , which is iid, uniform, O(1) random variables.

The classes BPP, RP, co-RP, ZPP are defined as the class of languages L so that

$\exists$ randomized TM M and a polys $f(n)$, s.t. $\forall x \in \Sigma^n$, M runs for $\leq f(|x|)$ steps, uses $\leq f(n)$ bits of r , and

	BPP	RP	coRP	ZPP
if $x \in L$, $\mathbb{P}\{M(x,r) = 0\} \leq$	$1/3$	$1/2$	0	0
if $x \notin L$, $\mathbb{P}\{M(x,r) = 1\} \leq$	$1/3$	0	$1/2$	0
$\mathbb{P}\{M(x,r) = \perp\}$	≤ 0	0	0	$1/2$

The $1/3, 1/2$ above are arbitrary.

Lemma (Chernoff) If X_i are iid 0/1 random-variables,

$$\mathbb{P}\left\{\frac{1}{m} \sum_{i=1}^m X_i \geq \mathbb{E}X_1 + \epsilon\right\} \leq \exp(-2\epsilon^2 m)$$

Cor The $1/3, 1/2$'s above can be taken to be $1/\text{poly}(n)$.

- For BPP, run $\text{poly}(n)$ times and take majority.
- For RP, run $\text{poly}(n)$ times and accept if you see a 1.
- For ZPP, run $\text{poly}(n)$ times and output $\perp$ only if you get $\perp$ every time.

FACTS.

$$ZPP = RP \cap \text{coRP} \subseteq RP \cup \text{coRP} \subseteq BPP$$

ZPP alg $\Rightarrow$ RP alg (or coRP)
by $\perp \mapsto 0$ (or 1)

run the RP alg twice

(RP alg) $\mapsto$ ZPP alg
(coRP alg) $\mapsto$ ZPP alg
by running both and output $\perp$ on $\perp$

$$RP \subseteq NP$$

the "good" random string is the certificate.

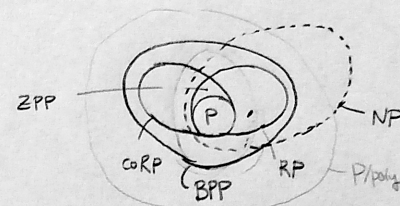
Say $L \in RP$.

$$x \in L \Rightarrow \exists r, M(x,r) = 1$$

$$x \notin L \Rightarrow \forall r, M(x,r) = 0$$

So, how big is BPP?

(6)



these separations that we drawn aren't known to be strict.

- we don't know if $BPP \subseteq NP$, or if $P \subseteq BPP$.

$$NP \subseteq BPP \Rightarrow NP = RP$$

One limit we do know is $BPP \subseteq P/\text{poly}$.

Thm $BPP \subseteq P/\text{poly}$.

Pf. Idea: Find one random seed r_n that works on all inputs of length n . That will be the advice string.

Let $L \in BPP$, $\exists M$ s.t. $\forall n \geq 1, \forall x \in \Sigma^n$,
 $\mathbb{P}\{M(x,r) \neq L(x)\} \leq \frac{1}{10 \cdot |\Sigma|^n}$

So by the union bound,

$$\mathbb{P}\left\{\forall x \in \Sigma^n, M(x,r) = L(x)\right\} \geq 1 - |\Sigma|^n \left(\frac{1}{10 \cdot |\Sigma|^n}\right) > 0.$$

So $\exists r_n$ that "works" for all x . This was our alt. def of P/poly :

$$\exists M, \{r_n\} \text{ s.t. } x \in L \iff M(x, r_n) = 1$$