Last time: • $NP^{SAT} \subseteq \Sigma_2^P$ (finishing $NP^{SAT} = \Sigma_2^P$)

• Circuits, P/poly, nonuniformity

Reading: P+p. 4.3, 11.4, AB 6.4, Ca: 4.1, 4.2

• P/poly = poly-size ckts

(Special case: if $L \in P$, then L has poly-size ckts)

motivates field of ckt cxity!

Today:

Reading: AB 6.4, Ca: 4.2 Papad. 17.13

end of unit
• Karp-Lipton thm: $NP \subseteq P/poly \Rightarrow PH$ collapses to Σ_2^P.
• Baker-Gill-Solovay: $P^A = NP^A$, $P^B \neq NP^B$ some oracles A, B.

Reading: AB 3.4, Ca: 12.1

• start Next unit: Hierarchy thms, relationships among resources
  • "padding arguments" Papad. 20.1, Sipser 9.1, Papad. 7.1, 7.2
    • condit. results

Questions?

Admin: PS1 today
Next week: videos.

Motiv: what's conn. betw. P/poly + NP, PSPACE, etc?

Know: $P \subseteq P/poly$; but even

$P/1 \not\subseteq NP, PSPACE, EXP$, b/c

$P/1$ contains undec L's:

$L = \{1^n : M_n \text{ halts on empty string}\}$ undec.

the 1 bit of advice encodes L.

Think: P/poly "not much bigger" than P...

Evidence?

Karp-Lipton Thm: If $NP \subseteq P/poly$, then $PH = \Sigma_2^p$.

Equiv. to: If SAT has poly(n)-size ckts, .

Idea: "exist. quant. for free"; leverage to get .

Pf: By collapse thm, to show $PH = \Sigma_2^p$, enough to show $\Pi_2^p \subseteq \Sigma_2^p$; we'll do this.

So let $L \in \Pi_2^p$; i.e., $x \in L \Leftrightarrow \forall^p y \, \exists^p z \, [\underbrace{D}_{\text{det poly time } D}(x,y,z)=1]$.

Consider $L' := \{(x,y) : \exists^p z \, [D(x,y,z)=1\}$.

$L' \in NP$, so $L' \leq_p SAT$.

So there's a poly-time reduc. s.t. given any (x,y), reduc. outputs ϕ_{xy} (a formula) s.t.

$\exists^p z \, [D(x,y,z)=1] \Leftrightarrow \phi_{x,y} \in SAT$.

So

$x \in L \Leftrightarrow \forall^p y \, [\phi_{x,y} \in SAT]$.

Since (assump) SAT has poly-size ckts, $\exists$ ckt fam. $(C_n)_{n \geq 1}$ s.t. $\phi_{x,y} \in SAT$ iff

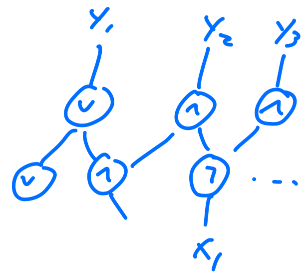
$$C_{|\phi_{x,y}|}(\phi_{x,y}) = 1.$$

Now we use "search-to-dec" reduction for ckts:

"Suppose $\exists$ poly-size ckt fam. $(C_n)_{n \geq 1}$ that decides SAT( ).

Then there's a $\overset{\text{poly-size}}{\text{ckt}}$ family $(C'_n)_{n\geq 1}$ that outputs a sat asst (when one exists).*

C' has multiple output bits: no prob.



So there is a $\overset{\text{poly-size}}{\text{ckt}}$ family $(C'_n)_{n\geq 1}$ s.t.

$\phi_{x,y} \in SAT$ iff $C'_{|\phi_{x,y}|}(\phi_{x,y})$ outputs a sat asst for $\phi_{x,y}$.

Key idea: "$\exists$" quant. can "guess" this C'.

Endgame: Consider any x.

- Sps $x \in L$. Then $\exists^p C' \forall^p y\, [\phi_{x,y}(C'_{|\phi_{x,y}|}(\phi_{x,y}))=1]$.

- Sps $x \notin L$. Means $\exists^p y\, [\phi_{x,y} \notin SAT]$.

So no z has $\phi_{x,y}(z)=1$, hence

$$\nexists^p C' \forall^p y\, [\phi_{x,y}(C'(\phi_{x,y}))=1].$$

So $x \in L$ iff $\exists^p C' \forall^p y\, [\phi_{x,y}(C'(\phi_{x,y}))=1]$.

This is Σ_2^p: given x, C', & y, can

- compute $\phi_{x,y}$

- eval. $\underbrace{C' \text{ on } \phi_{x,y},}_{b}$ then
- eval. $\phi_{x,y}$ on b

$\Bigg\}$ all in poly time.

Baker-Gill-Solovay thm:

There are oracles/langs A, B s.t.

(1) $P^A = NP^A$

(2) $P^B \neq NP^B$.

Motiv: natural to ask $P^A \stackrel{?}{=} NP^A$

Pf: (1): IOU. Idea: A oracle for "very powerful" (PSPACE-complete) lang.; so strong that nondet doesn't help.

---

(2) For any $B \subseteq \{0,1\}^*$, define unary lang.

$$U_B := \{1^n : \exists x, |x| = n, x \in B\}.$$

For any B, $U_B \in NP^B$: if input $\neq 1^n$, reject;

if input $= 1^n$ some n, guess (nondet.) $|x| = n$, feed to B, accept if B-oracle says yes ".

Need to construct a B s.t. $U_B \notin P^B$.

Idea: any det polytime machine deciding U_B must, given 1^n, figure out whether some $x \in \{0,1\}^n$ is in B. 2^n poss... feels like poly(n)-time machine can't query B often enough to be sure.

Pf: diagonalization. We'll construct a B.

M_i = i^{th} TM in enum.

Build B in stages: initially, empty lang.
Add strings in each stage.
Each stage determines "fate" ($\in B$ / $\notin B$) of finite # strings.
Stage i ensures M_i^B doesn't correctly decide U_B in $2^n/10$ time.

Start of stage i: finite # strings had fate determined.
Choose $n >$ length of all . ($n > i$).
Run M_i on 1^n ~~for~~ with B-oracle $2^n/10$ steps (or till it stops, whichever is first).
- if M_i queries B-oracle on a string with already-det. fate: B answers consistently.
- if M_i queries B-oracle on a string whose fate is not yet det: declare the string $\notin B$.

Once M_i finishes: want to ensure M_i wrong on 1^n.

At most 1/10 of all length-n strings had fate dec. in this stage; all were declared $\notin B$.

So if M_i accepted 1^n, we set all rem. length-n strings to <u>not</u> be in B;

if M_i rejected we set all the ($\geq \frac{9}{10} 2^n$) remaining not-queried string of length n & set them to be in B.

So M_i^B is either wrong on 1^n, or doesn't halt in $2^n/10$ steps.

Every poly $p(n)$ has $p(n) < 2^n/10$ for suff large n.

Every TM occurs ∞ly often in enum.

So, fix any $p(n)$-time TM: it's M_i for some i s.t. $p(n) < 2^n/10 \ \forall n \geq i$.

By construc, $M_i^B(1^n)$ terminates in $< 2^n/10$ & is wrong.

So M_i^B doesn't decide U_B.

So $U_B \notin P^B$. ■

End of unit on oracles, ckts, poly-time hierarchy....

• Relationships among Resources •

Conditional results: often easy.

Std technique for condit. results: "padding".

"Containment (equality) translates P" ("inequality (contrap: separation) translates $\not\mathsf{P}$").

SUPPOSE $\mathsf{NTIME}(n^2) \subseteq \mathsf{TIME}(n^3)$.

Then could deduce sim. results "higher up":

Thm: If $\mathsf{NTIME}(n^2) \subseteq \mathsf{TIME}(n^3)$, then

then $\mathsf{NTIME}(n^{10}) \subseteq \mathsf{TIME}(n^{15})$.

Contrap: if $\mathsf{NTIME}(n^{10}) \not\subseteq \mathsf{TIME}(n^{15})$,
then

$\mathsf{NTIME}(n^2) \not\subseteq \mathsf{TIME}(n^3)$.

Pf: easy; next time.
