Last time:
- more nonuniformity: Boolean circuits
- \( \text{P/poly} \) corr. to poly-size chts
- Karp-Lipton theorem \( \Rightarrow \) AB 6.4, Ca: 4.2, Pp. 17.3
  \[ \text{NP has poly-size chts} \Rightarrow \text{PH collapses} \]

Today:
- Finish Karp-Lipton
- Baker-Gill-Solovay thm: \( \Rightarrow \) AB 3.4, Ca: 12.1
  - \( \exists \) oracle \( A \) s.t. \( P^A = NP^A \)
  - \( \exists \) oracle \( B \) s.t. \( P^B \neq NP^B \)
- Start next visit: relations between resources:
  - padding (Papad. 20.1), hierarchy thms (Sipser 9.1, Papad. 21.7.2)

Questions?

Recall Karp-Lipton:

**Karp-Lipton Thm:**

If \( \text{NP} \subseteq \text{P/poly} \) then \( \text{PH} = \Sigma_2^P \).

If SAT has poly-size chts then \( \text{PH} = \Sigma_2^P \).

Recall: \( \Sigma_2 \) + Collapse thm say:

- If \( \Pi_2^P \subseteq \Sigma_2^P \) then \( \text{PH} = \Sigma_2^P \).

So fix \( L \in \Pi_2^P \); plan is to use \( \) to show \( L \in \Sigma_2^P \), some dot poly time \( D \).
\[ \iff x \in \mathcal{L} \iff \forall y \exists z \left[ D(x,y,z) = 1 \right] \]

Let's define
\[ \mathcal{L}' = \{ (x,y) : \exists z \left[ D(x,y,z) = 1 \right] \} \]
\( L' \in \text{NP}. \) So \( L' \leq_p \text{SAT}. \)

So given \((x,y),\) there's an efficient mapping

\[ (x,y) \mapsto \emptyset_{x,y} \quad \text{s.t.} \]

\[ \exists z \left[ D(x,y,z) = 1 \right] \iff \emptyset_{x,y} \in \text{SAT}. \]

So \[ x \in \mathcal{L} \iff \forall y \left[ \emptyset_{x,y} \in \text{SAT} \right]. \]

Continuing...

SAT has poly-size cuts, so
there's a cut fam. \( \{ C_n \}_{n \geq 1} \) s.t.
for any \( \emptyset \in \text{SAT} \)
\[ C_{|\emptyset|} (\emptyset) = 1. \]
\[ \emptyset_{x,y} \in \text{SAT} \iff C_{|\emptyset_{x,y}|} (\emptyset_{x,y}) = 1. \]

Now we use "search-to-decision" reduction:

Claim: fix \( \{ C_n \}_{n \geq 1} \) is a fam. of poly-size cuts

deciding SAT.

Then there's a fam. \( \{ C_n \}_{n \geq 1} \)
that outputs a sat. asst. when one exists (or outputs 1
if no sat. asst.)

(Idea: like HW: use \( C_n \) to decide if SAT or not,
then use \( C_m \) for various \( m \), to find s.t.)
Key obs: "∀" quant. can "guess" the $C_x$, ok.
Taking stock:

- $\exists x \in L$. Then $\exists^* C', \forall y (\phi_{x \cdot y}(C'(\phi_{x \cdot y})) = 1)$. \\

- $\exists x \notin L$. Then $\exists^* y (\phi_{x \cdot y} \notin \text{SAT})$: so for that $y$, no string $z$ causes $\phi_{x \cdot y}(z) = 1$, so no $C'$ can have $\phi_{x \cdot y}(C'(\phi_{x \cdot y})) = 1$. \\
  $D(x, C')$ \\

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

This is a $\Sigma_2$ statement: given $x, C', y$: in poly time can constr. $\phi_{x \cdot y}$; run $C'$ on $\phi_{x \cdot y}$ to get some string $z$; eval $\phi_{x \cdot y}$ on $z$ + acc. iff it sat. $\phi_{x \cdot y}$.

So $L \in \Sigma_2$.

Baker - Gill - Solovay Thm

Cx thy is frustrating...

$C_x = C_e$. 
Maybe for some \( L \), can prove \( L_c^c = L_c^c \)?

= of classes "up to an oracle" can be misleading.

Thus: There are decidable langs \( A, B \) s.t.

1. \( P^A = NP^A \), but
2. \( P^B \neq NP^B \).

1. IOU. (A so powerful that having nondet. matter)

2. Given any lang. \( B \), let \( U_B \) be the unary lang

\[ U_B = \{ 1^n : \exists x \in B \text{ with } |x| = n \} \]

For any \( B \), \( U_B \in NP^B \). Given input \( 1^n \)

NTM guesses an \( |x| = n \), calls \( B \)-oracle on \( x \),

acc iff \( B \)-oracle says \( Y \).

Remains to cook up a lang. \( B \) s.t. \( U_B \notin P^B \)

Idea: any det poly-time machine for \( U_B \) must,
given \( I^n \) as input, figure out whether any \( |x| = n \) is in \( B \).
But \( 2^n \) poss, \( + \) in poly(\( n \)) time can't make enough calls
to \( B \)-oracle.
We'll construct a B that makes this idea rigorous.

Let $M_i = i^{th}$ TM (descr. is binary rep. of i)

We construct B in stages:

- Initially empty
- Each stage adds some strings to B
- "" det. x ∈ B vs x ∉ B for finitely many strings
- $i^{th}$ stage ensures that $M_i^B$ doesn't decide $U_B$ correctly even if it runs for $2^{7/10}$ steps.

Here's construct of B:

Start of stage i: only finite # strings b.s.d B-Fate det. so far.

Choose an x ∈ $\Sigma_i^*$ st. x longer than all of y.

Run $M_i^B$ on $1^n$ for $2^{7/10}$ steps or till it halts. Suppose

If $M_i^B$ asks B-oracle abt a string whose B-Fate det. in earlier stage, answer consistently.

If $M_i^B$ asks B-oracle abt new string (different from earlier)

declare that string not in B.

Once $M_i^B$ stops, point is to ensure $M_i^B$'s answer on $1^n$ is wrong. Know $2^{7/10}$ str. in $\Sigma^n$ had fate det. all were det. not to be in B.

So if $M_i^B$ acc. $1^n$, we say all remaining str. in $\Sigma^n$ are ∉ B (so $M_i^B$ wrong on $1^n$)

if $M_i^B$ rej. $1^n$ we pick on x ∈ $\Sigma^n$ whose
fate wasn’t dec. & put it in B.

So $M_i^B$ is wrong.

Any poly $p(n) < 2^n / 10$ for suff large n.
Every TM occurs only many times in enum.

Fix any polytime TM $M_i$; it occurs in enum as $M_i$ for arb. large i (large enough s.t. $p(n) < 2^n / 10$ for $n > i$).

So in constr of $B$, $M_i^B$ run to complete.

+ it’s wrong on $1^n$. So $M_i^B$ doesn’t dec. $U_B$.

Unit #3: Relations among Resources, Hierarchy

Thus

Very hard to prove uncond. statements " ... but easy to prove conditional statements

(Padding)

Suppose $\text{NTIME}(t^2) \subseteq \text{TIME}(t^3)$.

Would this imply $\text{NTIME}(t) \subseteq \text{TIME}(t^{1.5})$?

But:

Thus if $\text{NTIME}(t^2) \subseteq \text{TIME}(t^3)$, the...
\[ \text{NTIME}(n^{10}) \subseteq \text{TIME}(n^{15}). \]

"Containment translates upward, inequality/分离 down." \[ \rightarrow \text{time} \]

\textbf{Pf:} Let \( L \subseteq \text{NTIME}(n^{10}) \). Let \( N \) be NTM for \( L \).

Let \( L_2 := \{ x \#^{1x^5 \cdot 1x^5} : x \in L \} \) (new dummy symbol)

This \( L_2 \subseteq \text{NTIME}(n^{2}) \): given input, check its of the form \( x\#^{1x^5 \cdot 1x^5} \) (very fast) + then run \( N \) on the \( x \)-part length \( n^{15} \), so \( N \) takes time \( n^{2} \).

So by assumption, \( \exists \) \text{det M} \text{ time } n^{3} \text{ for } L_2.

Here's on \( n^{15} \) time: \text{det M} \text{ for } L_1:

given \( 1x^5 \), write down \( x\#^{1x^5 \cdot 1x^5} \)

run \( M \) on it: \( n^{15} \) time.

\[ \text{Hierarchy Thus} \]

\[ \text{Things we don't know:} \]

- \( \text{is } \text{TIME}(n) = \text{TIME}(n^{10})? \quad \text{NO} \)
- \( \text{is every decidable } L \text{ in } \text{TIME}(2^n)? \quad \text{NO} \)

\[ \text{TIME}(n^3)? \]
Recall: • a f: \mathbb{N} \rightarrow \mathbb{N} is \underline{computable} if some TM computes it. \quad n \rightarrow f(n)
• a lang L is \underline{decidable} if some TM decides it (always halts either acc. or rej. every x).

Thm: Given any computable f(n), there is a \underline{decidable} lang. L \in \text{TIME}(f(n)).

Pf: The idea: Diagonalize to construct a TM that decides a lang. L but for each f(n)-time-bounded TM, disagrees with M_i on some input.

We'll have L \subseteq \{0,1\}^*.

x_1, x_2, x_3, ... \ x_i; ... \ \text{enum. of binary strings}

i; \text{in binary}

Can view x_i as a descip. of M_i; i'th TM.
We'll use an $\text{uni. TM}$ to view an input w = x_i as both an input & a machine... (next time)