Last time:
- co-classes (definition)
  - poly-time reductions
  - NP completeness → Readings: Sipser 7.4, 7.5; AB 2.2-2.4;
    Papad. Ch. 8, 9
  - Cook-Levin theorem, other NPC problems
    - 3CNF, CLIQUE, IND-SET, VC, SUBSET-SUM
  - state Ladner’s Theorem AB 3.3

Today:
- Proof of Ladner’s theorem
- start unit on oracles & poly-time hierarchy

Readings: Cs: 2.3-2.6, Papad. 17.2

Note: PS 0 due today, PS 1 out today
(40H/week)

Questions?

**Thm: (Ladner) If $P \neq \text{NP}$ then there’s an $L \in \text{NP}$ s.t. (i) $L \not\in P$, but also $L \not\in \text{NPC}$.

- a priori possible for $P$, $\text{NP}$, $\text{NPC}$ assuming $P \neq \text{NP}:
  - $L \not\in \text{NPC}$

Two proofs
1st: diag. over both \( \text{poly-time TMs} \) + poly-time TMs.

\[ \text{2nd proof (Impagliazzo): } \text{Given any lang in NP/P} \]

"water it down" to get \( L' \):

- strictly easier than \( L \) (not NPC), but
- hard enough s.t. \( \notin \text{P} \).

To cook up \( L' \): diagonalization "padding".

\[ \text{Setup:} \]

- Thm assumes \( \text{P} \neq \text{NP} \), so have SAT \( \notin \text{P} \).
- Recall TMs can be enumerated.\[ \text{Let } M_i = i^{th} \text{ TM in enum. (} M_i = \text{TM whose descript. is binary rep. of } i). \]
- Prelim. def. ("padding"): Given \( H: \mathbb{N} \rightarrow \mathbb{N} \), define \( SAT_H \) to be all length-\( n \) satisfiable forms padded with \( n^{H(n)} \) many \( \text{Z}'s \) at end. Formally,

\[ SAT_H = \bigcup_{n=1}^{n^{H(n)}} \{ \psi \in \text{SAT, } \psi(1^n) = 1 \} \]

This is "watered-down SAT".

Ex: Let \( |\psi| = k \), consider \( \psi(01) \)

\[ 2^{14k-(14k+2)} \]

\[ 2^k - k - 1 = 2^e \]
\( \text{input/extension } n = 2^L \leq \)
\( \text{poly}(n) \text{ time } = \Omega(1) \text{ time} \)

Consider \( \psi_0 \).
\[
\begin{array}{c@{}c@{}c@{}c}
L^2 & (\text{MMH}) \\
L^1 & L^{2-1} & L^2 = n
\end{array}
\]

\( n^3 \text{ time alg.} \equiv L^6 \text{ time alg.} \)

(Note: write "\( SAT_T(x) = 1 \)" means \( x \in SAT_T \).
"\( SAT_T(x) = 0 \)" means \( x \not\in SAT_T \).)

\(^\star\) We use the following fn \( H \):

\( H: \mathbb{N} \to \mathbb{N} \) is:
- \( H(n) \) is the smallest \( i < \log \log n \)
  s.t. \( \forall i \leq \log n, x \in \{0,1\}^i, \) TM \( M_i \) outputs \( SAT_H(x) \)
  within \( i \cdot |x| \) steps.
- If there's no such \( i \) then \( H(n) = \log \log n \).

Two prelim. claims:
(We're assuming \( P \neq NP \))

C1: The fn \( H \) is well defined, \( \forall H(n) \) can be computed,
given \( n \), in \( \text{poly}(n) \) time. \( \square \)

C2: \( SAT_H \in P \iff H(n) \in O(1) \) \( (\exists C \text{ s.t. } \forall n, \ \text{have } H(n) \in C) \).
In fact, if \( SAT_H \notin P \) then \( \lim_{n \to \infty} H(n) = \infty \).
Proof of Ladner's thm:

1st part: show SATₜₜ ∈ P. 

**We'll argue that** SATₜₜ ∈ P ⇒ P = NP.

Sps SATₜₜ ∈ P. By C₂, H(n) ≤ C, so SATₜₜ is SAT padded w ≤ nᶜ out of garbage; so a poly-time alg for SATₜₜ ⇒ poly-time alg for SAT, so P = NP. **1st part**

2nd part: show SATₜₜ is not NPC.

**We'll argue that** if SATₜₜ is NPC ⇒ P = NP.

Sps SATₜₜ is NPC. So ∃ nⁱ-time red. P from
SAT to SAT\_H, \( i = \text{some const.} \)

By 1st part, SAT\_H \( \in \mathcal{P} \). So by C2, \( H(n) \to \infty \).

Since \( P \)'s runtime is \( \leq n^i \), it maps

\[ n \text{-bit} \rightarrow (\leq n^i) \text{-bit} \]

inst. of SAT \( \rightarrow \) inst. of SAT\_H.

If fixed, \( H(n) \to \infty \); so \( n^i < \eta H(n) \) for large enough \( n \).

So these \( n^i \)-length inst. are of length \( 5x \) \( < (\eta^{\frac{1}{3}}) H(\eta^{\frac{1}{3}}) \).

This means (for large enough \( n \)) \( \psi \) maps

\[ n \text{-bit form}(\emptyset \in \text{SAT}) \rightarrow \text{a string} \]

where \[ |\psi| < \eta^{\frac{1}{3}} \]

\( \psi \in \text{SAT} \) iff \( \emptyset \in \text{SAT} \).

But this gives a \( \text{poly}(n) \)-time \( \psi \) for SAT.

(repeatedly applying \( \psi \)) So \( \mathcal{P} = \mathcal{NP} \).

Q: Are there natural NP-intermediate lang's?

A: we think so...

- Factoring
- Graph isomorphism = \( \{ (G_1, G_2) : G_1 \text{ isom. to } G_2 \} \)

New Topic: Poly-Time Hierarchy,
Oracles, + a little bit of
Circuits

Let's recall \( \text{NP} \):

new name: \( \Sigma_1^p = \text{NP} \)

A lang \( L \) is in \( \text{NP} \) if there's a TM \( D \) (det) and a poly \( p(n) \) st

\[
\forall \sum \exists \prod \left[ \exists y \in \Sigma^{p(n)} \left[ O(w, y) = \text{I} \right] \right]
\]

where \( n = |w| \).

Handy: "\( \exists \prod \)" mean "\( \exists y \in \Sigma^{p(n)} \)"

\( \forall p(n) y \) sim.

new name: \( \Pi_1^p = \text{co NP} \)

A lang \( L \) is in \( \text{coNP} \) if there's a TM \( D \) (det) and a poly \( p(n) \) st

\[
\forall \sum \exists \prod \left[ \forall y \in \prod^{p(n)} \left[ O(w, y) = \text{I} \right] \right]
\]

where \( n = |w| \).

Equiv. \( \exists \text{DTM} D', \) poly p st.

\[
\forall \sum \exists \prod \left[ O'(w, y) = \text{I} \right]
\]

(\( D' \) outputs opp. of \( D \).)

Reminder: we think

\[
\forall D = D' \Rightarrow \forall D
\]
Let's Generalize:

**Def:** Let $L$ is in $\Sigma_2^p$ if there's a poly-time det $TMD + a poly p$ s.t.

$$w \in L \iff \exists^{p(n)} y \forall^{p(n)} z \left[ O(w, y, z) = 1 \right]$$

Let $L$ is in $\Pi_2^p$ if there's a poly-time det $TMD + a poly p$ s.t.

$$w \in L \iff \forall^{p(n)} y \exists^{p(n)} z \left[ O(w, y, z) = 1 \right]$$

**Ex:** $MEF = \text{Minimum Equiv. Formula}$

$MEF = \{ \phi : \phi \text{ is a Bool Formula } \text{ s.t. there is no shorter form } \psi \text{ s.t. } \phi \equiv \psi \}$

$\overline{MEF} = \{ \phi : \phi \text{ is a Bool Formula st. there does exist a shorter form } \psi \text{ s.t. } \phi \equiv \psi \}$

$\phi \in \overline{MEF} \text{ means } \exists \psi \forall x \left[ \psi(x) = \phi(x) \text{ and } |\psi| < |\phi| \right]$. Checkable in det poly time given $\phi \& \psi$.
So \( MEF \in \Sigma_2^p \).

**Claim:** \( L \in \Sigma_2^p \iff L \in \Pi_2^p \).

**Pf:** \( x \in L \) means \( \exists \in^{p(n)} \forall \in^{p(n)} \exists D(x,y,z) = 1 \).
- \( x \notin L \) means \( \exists \in^{p(n)} \forall \in^{p(n)} \exists D(x,y,z) = 1 \).
  
  i.e.,
  \[
  \forall \in^{p(n)} \exists \in^{p(n)} \exists D'(x,y,z) = 1 
  \]
  \( (D' = \text{machine that outputs opp. of } D) \). ☐

**Generalize to any \( k \):**

\( \Sigma_k^p \): def. analogous to \( \Sigma_2^p \), now \( k \) alt. quant.,
  first one \( \exists \).

\( \Pi_k^p \): def. analogous to \( \Pi_2^p \), now \( k \) alt. quant.,
  first one \( \forall \).

**Def:** \( PH = \bigcup_{k \geq 2} \Sigma_k^p \cup \Pi_k^p \)
Next time: PH, "collapse them" oracles
basics of Boolean circuits