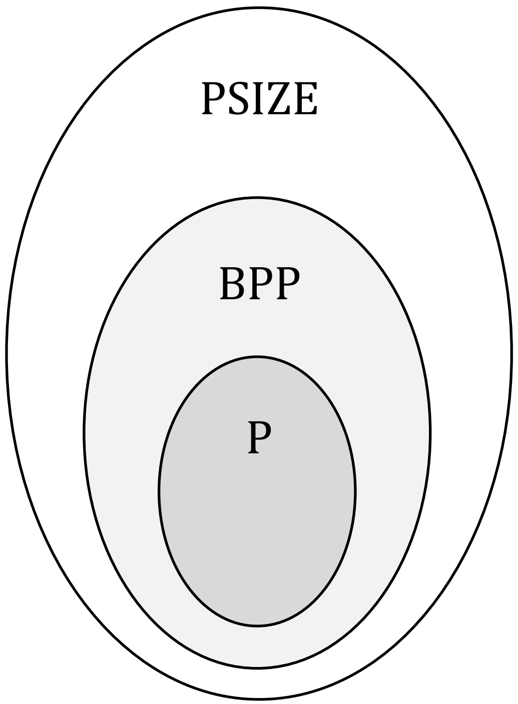
CMSC 28100

Introduction to Complexity Theory

Spring 2024
Instructor: William Hoza

Adleman's theorem



- Last class, we showed that $\mathrm{P} \subseteq \mathrm{PSIZE}$
- We got started proving a stronger theorem:

Adleman's Theorem: $\mathrm{BPP} \subseteq \mathrm{PSIZE}$

- Adleman's theorem is tantalizingly similar to the statement "$\mathrm{P} = \mathrm{BPP}$"

Proof of Adleman's theorem ($\mathrm{BPP} \subseteq \mathrm{PSIZE}$)

- **Proof:** Let $L \in \mathrm{BPP}$ where $L \subseteq \Sigma^*$
- Amplification lemma $\Rightarrow$ There exists a polynomial-time randomized Turing machine M such that for every $n \in \mathbb{N}$ and every $w \in \Sigma^n$:
  - If $w \in L$, then $\Pr[M \text{ accepts } w] > 1 - 1/|\Sigma|^n$
  - If $w \notin L$, then $\Pr[M \text{ rejects } w] < 1 - 1/|\Sigma|^n$
- Let $T(n)$ be the time complexity of M

Random bits: Good for all inputs simultaneously

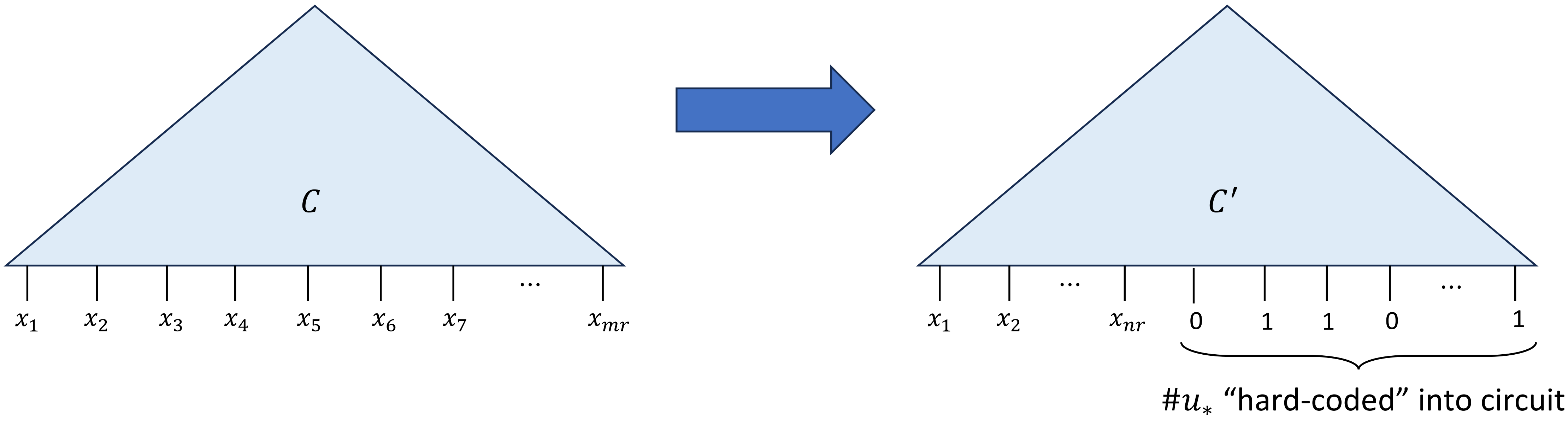
Claim: For every n, there exists $u_* \in \{0,1\}^{T(n)}$ that is "good" for all $w \in \Sigma^n$

- I.e., if we initialize M with w on tape 1 and u_* on tape 2, then it will correctly accept/reject depending on whether $w \in L$
- We proved the claim last class
- Now let's use the claim to prove Adleman's theorem

Constructing the circuit

- Let $R = \{w\#u : M \text{ accepts } w \text{ when tape 2 is initialized with } u\}$
- Then $R \in \mathrm{P} \subseteq \mathrm{PSIZE}$
- Let $n \in \mathbb{N}$. Our job is to construct a circuit that computes L_n
- Let C be an optimal circuit that computes R_m where $m = n + 1 + T(n)$
- If u is good for w, then $C(\langle w\#u\rangle) = L_n(\langle w\rangle)$

Constructing the circuit



- C' computes L_n, and it has size

$$O(m^k) = O\left(\left(n + 1 + T(n)\right)^k\right) = \mathrm{poly}(n)$$

Adleman's theorem and P vs. BPP

- Adleman's theorem ($\mathrm{BPP} \subseteq \mathrm{PSIZE}$) does not resolve the question of whether $\mathrm{P} = \mathrm{BPP}$
- However, after seeing Adleman's theorem, I hope that the conjecture "$\mathrm{P} = \mathrm{BPP}$" is starting to look plausible
- The conjecture can be supported by more compelling evidence, but it's beyond the scope of this course

BPP and the Extended Church-Turing Thesis

- Let L be a language

Extended Church-Turing Thesis:

It is physically possible to build a device that decides L in polynomial time if and only if $L \in \mathrm{P}$.

- Assuming $\mathrm{P} = \mathrm{BPP}$, the extended Church-Turing thesis survives the challenge posed by randomized computation!

BPP and the Extended Church-Turing Thesis

- Just in case, the thesis is sometimes revised to allow randomization:

Extended Church-Turing Thesis, version 2:

Let L be a language. It is physically possible to build a device that decides L in polynomial time if and only if $L \in \mathrm{BPP}$.

- This version is immune to the challenge posed by randomization
- However, there is a bigger threat: Quantum Computation

Quantum computing

- Properly studying quantum computing is beyond the scope of this course
- We will circle back to it later to discuss some key facts
- For now, let's stick with P as our model of efficient computation
- Because of quantum computing, P should probably not be considered the ultimate model of efficient computation, but it is still a valuable model

Which problems

can be solved

through computation?

CLASSICAL

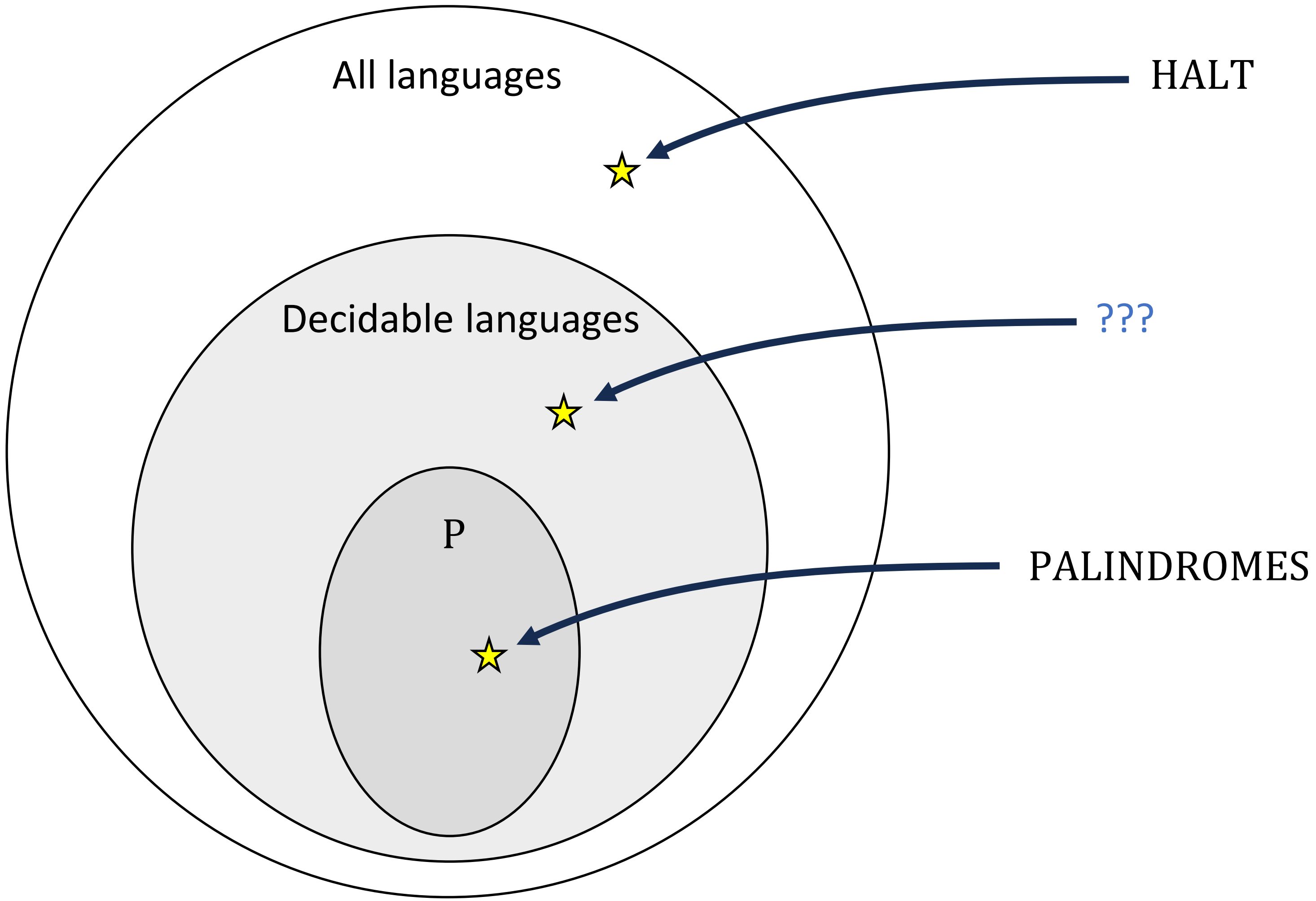
Which languages are in $\mathbf{P}$?

Which languages are not in $\mathbf{P}$?

Intractability vs. undecidability

- How can we prove that certain languages are outside P?
- Certainly $\mathrm{HALT} \notin \mathrm{P}$
- Is every decidable language in P?
  - This would mean that every algorithm can be modified to make it run in polynomial time!

Intractability vs. undecidability



The Time Hierarc

To prove this theorem, what should we prove?

A: The function T^3 grows faster than functions that are $o(T)$

B: Every Turing machine has time complexity $\Omega(T^3)$

C: There exists $L \in \mathrm{TIME}\big(o(T)\big)$ such that $L \notin \mathrm{TIME}(T^3)$

D: There exists $L \in \mathrm{TIME}(T^3)$ such that $L \notin \mathrm{TIME}\big(o(T)\big)$

Respond at PollEv.com/whoza or text "whoza" to 22333

- Let $T: \mathbb{N} \to \mathbb{N}$ be a "reasonabl

Theorem: $\mathrm{TIME}\big(o(T)\big) \neq \mathrm{TIME}(T^3)$

- "$\mathrm{TIME}\big(o(T)\big)$" means the class of languages decidable in time $o(T)$
- Note: $\mathrm{TIME}\big(o(T)\big) \subseteq \mathrm{TIME}(T) \subseteq \mathrm{TIME}(T^3)$
- Theorem interpretation: Given a little more time, we can solve more problems

Proof of the Time Hierarchy Theorem

Let $L = \{\langle M\rangle : M \text{ rejects } \langle M\rangle \text{ within } T(|\langle M\rangle|) \text{ steps}\}$

- **Claim 1:** Let M be any Turing machine with time complexity $T_M(n) = o\big(T(n)\big)$. Then M does not decide L.
- **Proof:** Let M' be a modified version of M, constructed by adding dummy states to artificially inflate $|\langle M'\rangle|$ until $T_M(|\langle M'\rangle|) \leq T(|\langle M'\rangle|)$
- If M accepts $\langle M'\rangle$, then $\langle M'\rangle \notin L$, and if M rejects M', then $\langle M'\rangle \in L$!

Proof of the Time Hierarchy Theorem

Let $L = \{\langle M \rangle : M \text{ rejects } \langle M \rangle \text{ within } T(|\langle M \rangle|) \text{ steps}\}$

- **Claim 2:** $L \in \mathrm{TIME}(T^3)$

Subtle point: How do we know when we're done?

- **Proof:** Given $\langle M \rangle$, we simulate M on $\langle M \rangle$ for $T(|\langle M \rangle|)$ steps and check whether it rejects
- Exercise: Verify that we can simulate a single step of M using $O(T^2)$ steps
- Total time complexity: $O(T \cdot T^2) = O(T^3)$