CMSC 28100

Introduction to Complexity Theory

Autumn 2025

Instructor: William Hoza

Deciding a language in time T

- Let $Y \subseteq \{0, 1\}^*$ and let $T: \mathbb{N} \to [0, \infty)$ be a function
- **Definition:** We say that Y can be decided in time T if there exists a one-tape Turing machine M such that
  - M decides Y, and
  - For every $n \in \mathbb{N}$ and every $w \in \{0, 1\}^n$, the running time of M on w is at most $T(n)$

The complexity class P

- **Definition:** For any function $T : \mathbb{N} \to [0, \infty)$, we define

$$\mathrm{TIME}(T) = \{Y \subseteq \{0,1\}^* : Y \text{ can be decided in time } O(T)\}$$

- **Definition:**

$$\mathrm{P} = \{Y \subseteq \{0,1\}^* : Y \text{ can be decided in time poly}(n)\}$$

$$= \bigcup_{k=1}^{\infty} \mathrm{TIME}\left(n^k\right)$$

- “Polynomial time”

The knapsack problem

- $\text{KNAPSACK} = \{\langle w_1, \dots, w_k, v_1, \dots, v_k, W, V\rangle : \text{there exists } S \subseteq \{1, 2, \dots, k\}$ $\text{such that } \Sigma_{i \in S}\, w_i \le W \text{ and } \Sigma_{i \in S}\, v_i \ge V\}$

Conjecture: $\text{KNAPSACK} \notin \text{P}$

The knapsack problem

- UNARY-VAL-KNAPSACK $= \{\langle w_1, \dots, w_k, 1^{v_1}, \dots, 1^{v_k}, W, 1^V \rangle$: there exists $S \subseteq \{1, 2, \dots, k\}$ such that $\Sigma_{i \in S}\, w_i \leq W$ and $\Sigma_{i \in S}\, v_i \geq V\}$

Theorem: UNARY-VAL-KNAPSACK $\in$ P

- **Proof technique:** "Dynamic programming"

Theorem: UNARY-VAL-KNAPSACK $\in$ P

- **Proof sketch:** We are given $\langle w_1, \ldots, w_k, 1^{v_1}, \ldots, 1^{v_k}, W, 1^V \rangle$
- Let $S_{j,v} \subseteq \{0, 1, \ldots, j\}$ minimize $\sum_{i \in S_{j,v}} w_i$ subject to $\sum_{i \in S_{j,v}} v_i \geq v$
  - Dummy item: $w_0 = v_0 = \infty$
- For $j = 1$ to k, for $v = 1$ to V:
  - Compute $S_{j,v} =$ whichever is less heavy: $S_{j-1,v}$ or $\{j\} \cup S_{j-1,v-v_j}$
- If $\sum_{i \in S_{k,V}} w_i \leq W$, then accept, otherwise reject

Exercise: Rigorously analyze time complexity

Note on standards of rigor

- Going forward, when we analyze specific algorithms, we will often assert that they run in polynomial time without a rigorous proof
  - In each case, one can rigorously prove the time bound by describing a TM implementation and reasoning about the motions of the heads…
  - But this is tedious
  - Note: We still prove correctness whenever it is nontrivial, just not efficiency
- You should follow this convention on exercise 13 and beyond

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

Examples of languages in P

- PALINDROMES
- PARITY
- UNARY-VAL-KNAPSACK
- PRIMES

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

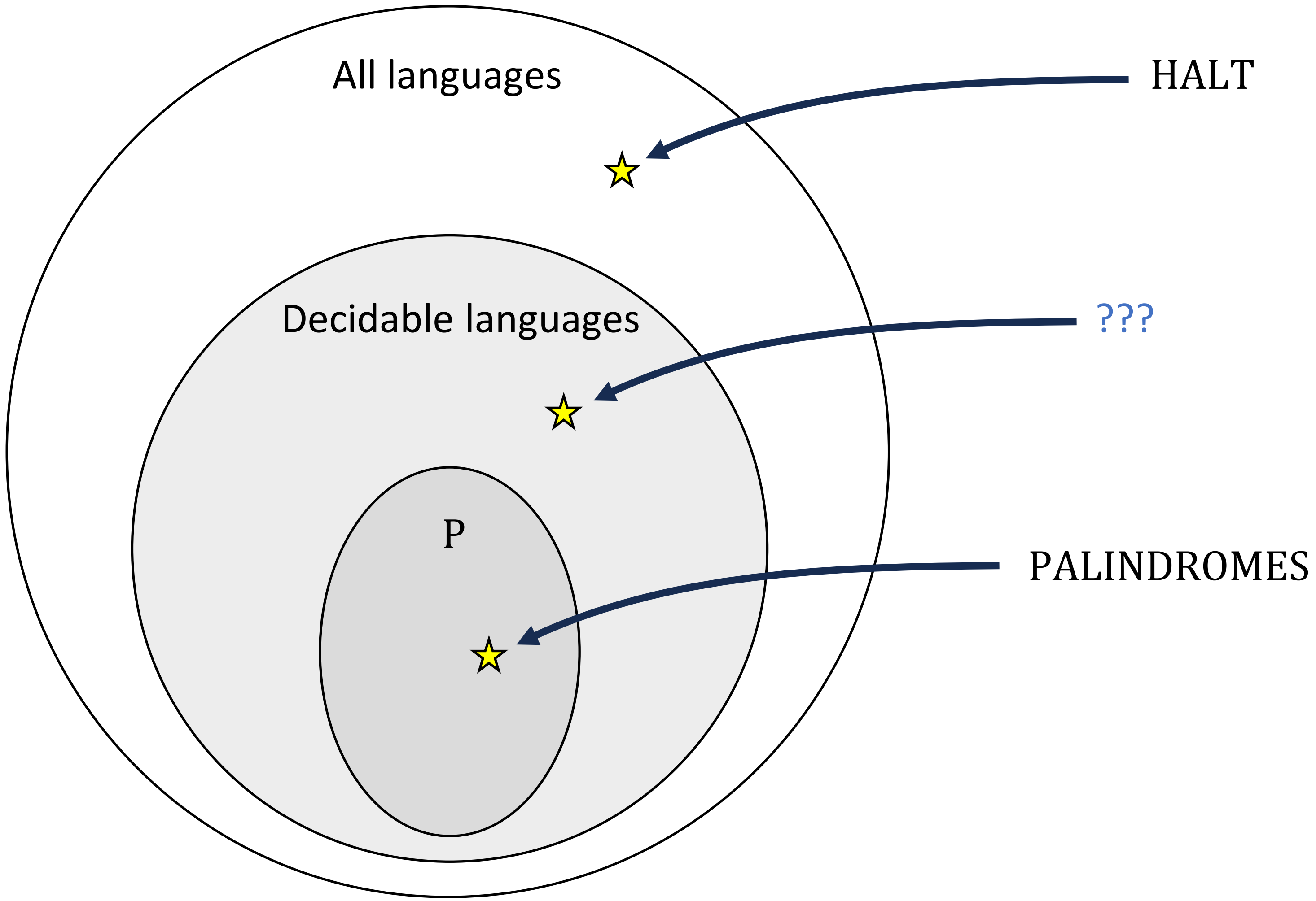
Examples of languages that are not in P

- Maybe CLIQUE ?
  - No proof…
- Maybe $\mathrm{KNAPSACK}$?
  - No proof…
- HALT

Intractability vs. undecidability

- Maybe every decidable language is in P???
    - Can every algorithm be modified to make it run in polynomial time??? 🤯

Intractability vs. undecidability



Intractability vs. undecidability

Theorem: There exists $Y \subseteq \{0,1\}^*$ such that Y is decidable, but $Y \notin \mathrm{P}$.

- **Proof**: Let $Y = \left\{\langle M \rangle : M \text{ rejects } \langle M \rangle \text{ within } 2^{|\langle M \rangle|} \text{ steps}\right\}$
- On the next three slides, we will show that Y is decidable and $Y \notin \mathrm{P}$

Proof that Y is decidable

$Y = \{\langle M \rangle : M \text{ rejects } \langle M \rangle \text{ within } 2^{|\langle M \rangle|} \text{ steps}\}$

- An algorithm that decides Y:

Given the input $\langle M \rangle$:

1. Simulate M on $\langle M \rangle$ for $2^{|\langle M \rangle|}$ steps
2. If it rejects within that time, accept
3. Otherwise, reject

Proof that $Y \notin$ P

Which of the following best describes what we've proven?

A: We showed that $T(n) > 2^n$ for a single value of n

B: We showed that $T(n) > 2^n$ for all n

C: We showed that $T(n) > 2^n$ for all sufficiently large n

D: We showed that $T(n) > 2^n$ for infinitely many n

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

- Let R be a TM that decide
- Let $T: \mathbb{N} \to \mathbb{N}$ be the time complexity of R, and let $n = |\langle R \rangle|$
- Does R accept $\langle R \rangle$? No, because that would imply $\langle R \rangle \notin Y$
- Does R reject $\langle R \rangle$ within 2^n steps? No, because that would imply $\langle R \rangle \in Y$
- Only remaining possibility: R rejects $\langle R \rangle$ after more than 2^n steps
- Therefore, $T(n) > 2^n$… but this does not imply $T(n) \neq \text{poly}(n)$ 😟

Proof that $Y \notin \mathrm{P}$

$Y = \{\langle M \rangle : M \text{ rejects } \langle M \rangle \text{ within } 2^{|\langle M \rangle|} \text{ steps}\}$

- Let R be a TM that decides Y, with time complexity $T: \mathbb{N} \to \mathbb{N}$
- Add dummy states!
- For infinitely many values of n, there exists a TM R_n such that R_n decides Y, R_n has time complexity T, and $|\langle R_n \rangle| = n$
- Each R_n must reject $\langle R_n \rangle$ after more than 2^n steps by diagonalization
- Therefore, $T(n) > 2^n$ for infinitely many values of n, hence $T(n) \neq \mathrm{poly}(n)$

The Time Hierarchy Theorem

- Using the same proof idea, we can prove a more general theorem:

Time Hierarchy Theorem: For every* function $T: \mathbb{N} \to \mathbb{N}$ such that $T(n) \geq n$, there is a language $Y \in \mathrm{TIME}(T^4)$ such that $Y \notin \mathrm{TIME}\big(o(T)\big)$.

  - *assuming T is a "reasonable" time complexity bound. We will come back to this

- "$\mathrm{TIME}\big(o(T)\big)$" means the set of languages that are decidable in time $o(T)$
- "Given more time, we can solve more problems"

Proof of the Time Hierarchy Theorem

- Let $Y = \{\langle M \rangle : M \text{ rejects } \langle M \rangle \text{ within } T(|\langle M \rangle|) \text{ steps}\}$
- On the next four slides, we will prove:
  - $Y \in \mathrm{TIME}(T^4)$
  - $Y \notin \mathrm{TIME}\big(o(T)\big)$

Proof that $Y \in \mathrm{TIME}(T^4)$

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

- An algorithm that decides Y:

Given the input $\langle M \rangle$:

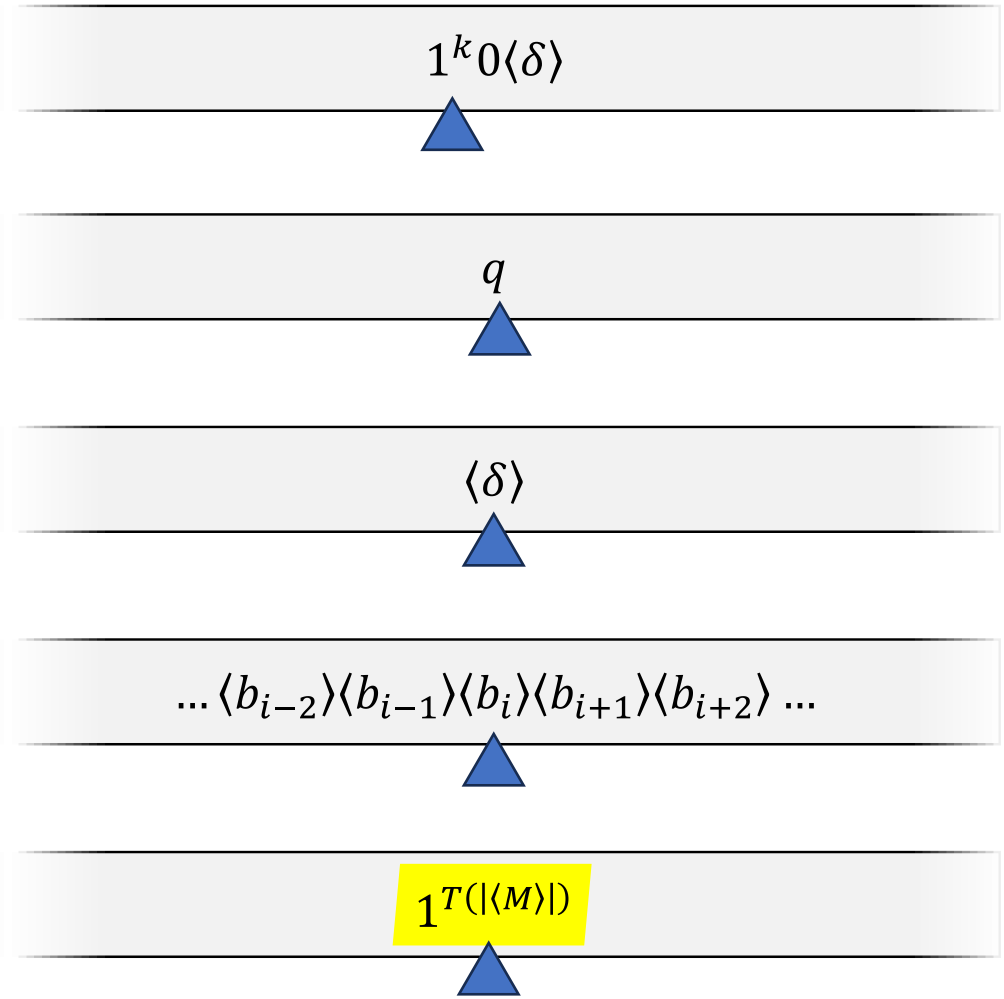
1. Simulate M on $\langle M \rangle$ for $T(|\langle M \rangle|)$ steps
2. If it rejects within that time, accept
3. Otherwise, reject

- Time complexity in the TM model?

Proof that $Y \in \mathrm{TIME}(T^4)$

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

- Let $n = |\langle M \rangle|$
- Each simulated step takes $O(n)$ actual steps
- Total time complexity of multi-tape machine: $O(T(n) \cdot n)$
- After converting to a one-tape machine: $O(T(n)^2 \cdot n^2) = O(T(n)^4)$



Time-constructible functions

- **Definition:** A function $T: \mathbb{N} \to \mathbb{N}$ is time-constructible if there exists a multi-tape Turing machine M such that
  - Given input 1^n, M halts with $1^{T(n)}$ written on tape 2
  - M has time complexity $O\big(T(n)\big)$
- Our proof that $Y \in \mathrm{TIME}(T^4)$ works assuming T is time-constructible
- All "reasonable" time complexity bounds (e.g., $5n$ or n^2 or 2^n) are time-constructible