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 Time Hierarchy 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 $\langle M \rangle$:

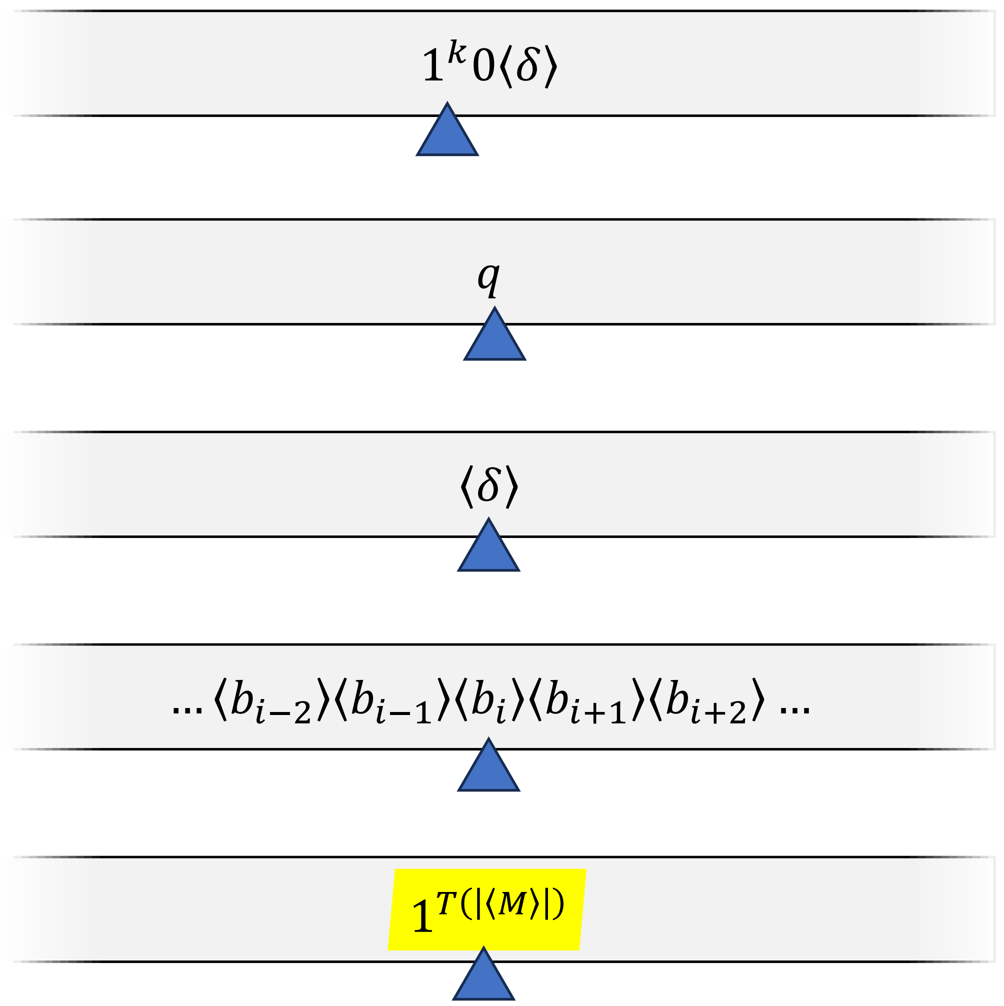
1. Simulate M on $\langle M \rangle$ for $T(|\langle M \rangle|)$ steps
2. If it rejects within that time, accept
3. Otherwise, reject

- 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

Time Hierarchy Theorem

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

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

- We showed $Y \in \mathrm{TIME}(T^4)$
- We still need to show $Y \notin \mathrm{TIME}\big(o(T)\big)$

Proof that $Y \notin \mathrm{TIME}\big(o(T)\big)$

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

- 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 $T(n)$ steps by diagonalization
- Therefore, $T'(n) > T(n)$ for infinitely many values of n, hence T' is not $o(T)$

Robustness of P, revisited

- Let $Y \subseteq \{0,1\}^*$. If $Y \notin \mathrm{P}$, then Y cannot be decided by…
  - A poly-time one-tape Turing machine
  - A poly-time multi-tape Turing machine
- **OBJECTION:** “Practical computers are very different from Turing machines!”
- **RESPONSE:** The “word RAM” model

Word RAM model (RAM = Random Access Machine)

- (This model will not be on homework exercises or exams)
- A word RAM program consists of a list of instructions
- Available instructions include:
  - $R_i \leftarrow 0$ or $R_i \leftarrow 1$ or $R_i \leftarrow R_j$
  - $R_i \leftarrow R_j$ op R_k where op $\in$ { `+, -, *, /, %, ==, <, >, &&, ||, &, |, ^, <<, >>` }
  - IF R_i GOTO k
  - ACCEPT or REJECT

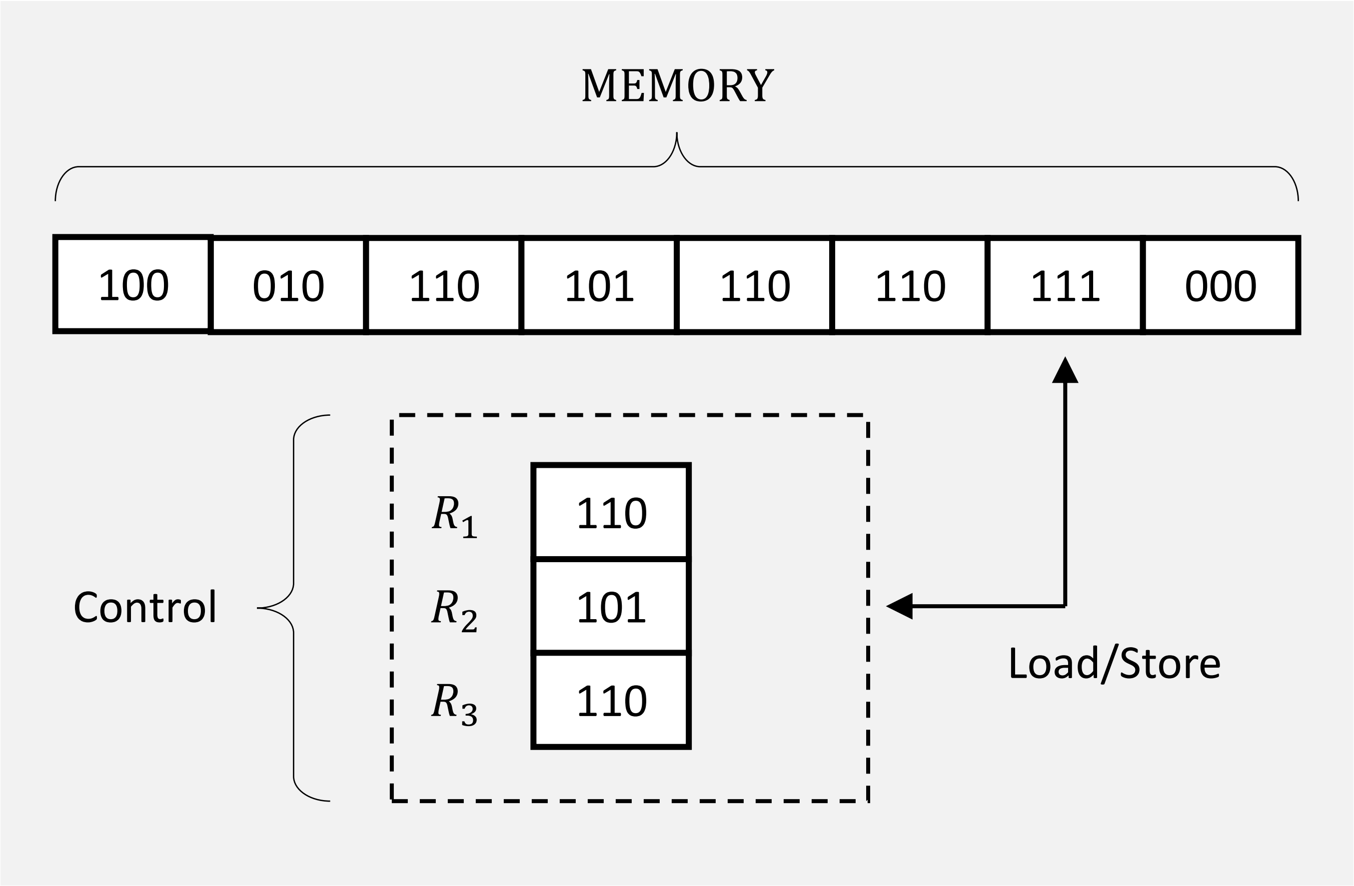
R_i is a "global variable" of type `unsigned int`

(The details are not completely standardized. This is just one reasonable version of the model)

Word RAM model

- Each R_i holds a k-bit "word" representing a number in $\{0, 1, \ldots, 2^k - 1\}$
- k is called the "word size"
- In practice, maybe $k = 64$
- In theory, we think of k as "large enough" and growing with n
- Operations on words take $O(1)$ time, unlike TM model!

Word RAM model



- There is also a large memory (an array of words)
- Instructions:
  - $R_i \leftarrow \text{MEMORY}[R_j]$
  - $\text{MEMORY}[R_i] \leftarrow R_j$
- Instantly access any desired location in memory, unlike the TM model!

Word RAM model

- Let the input be $w \in \{0, 1\}^n$ and let the word size be $k \geq \log(n+1)$
- MEMORY has 2^k cells
- Initially, $\mathrm{MEMORY}[0] = n$ and $\mathrm{MEMORY}[i] = w_i$ for $1 \leq i \leq n$

Example:

$w = 100110$

$k = 3$

110	001	000	000	001	001	000	000

MEMORY

Word RAM model

- Let $Y \subseteq \{0,1\}^*$, let P be a word RAM program, and let $T\colon \mathbb{N} \to \mathbb{N}$
- We say that P decides Y within time T if whenever we run P on an input $w \in \{0,1\}^*$ using a word size $k \geq \log(|w|+1)$:
  - P halts within $T(|w|)$ steps
  - If P halts within 2^k steps and $w \in Y$, then P accepts
  - If P halts within 2^k steps and $w \notin Y$, then P rejects

Computation is not reliable after 2^k steps

Word RAM model

- Word RAM Time Complexity $\approx$ Time Complexity "In Practice"
- Some version of the word RAM model is typically assumed (implicitly or explicitly) in algorithms courses and the computing industry

Robustness of P

- Let $Y \subseteq \{0,1\}^*$

Theorem: There is a word RAM program that decides Y in time $\mathrm{poly}(n)$ if and only if there is a Turing machine that decides Y in time $\mathrm{poly}(n)$.

- Proof omitted

Which problems

can be solved

through computation?

Is $\mathbf{P}$ a good model of tractability?

Boolean logic

- We have studied several rival models of computation
  - Turing machine, multi-tape Turing machine, word RAM, …
- Next: Computation based on networks of logic gates
  - Closely related to practical electronics
  - Extremely important in theory, too!

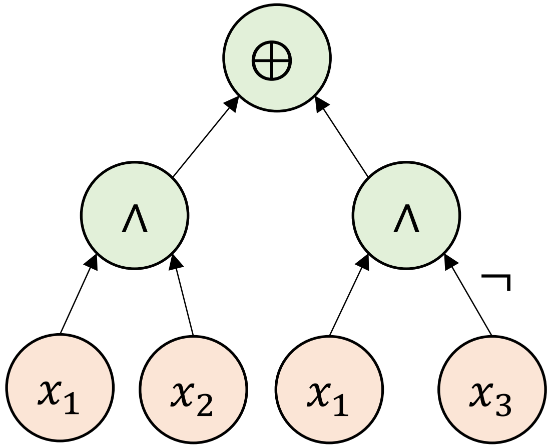
Binary logical operations

- AND: $a \wedge b$
- OR: $a \vee b$
- XOR: $a \oplus b$
- Equality: $a == b$

- AND/OR combined with negations:
  - $\bar{a} \vee b$, $a \vee \bar{b}$, $\bar{a} \wedge \bar{b}$, etc.
- Notation: $\bar{a}$ denotes the negation of a
  - Pronounced “NOT a”
  - Also written $\neg a$

Boolean formulas

- **Definition:** An n-variate Boolean formula is a rooted binary tree
  - Each internal node is labeled with a binary logical operation
  - Each leaf is labeled with 0, 1, or a variable among $x_1, \dots, x_n$
- It computes $f: \{0,1\}^n \to \{0,1\}$
- E.g., $f(x_1, x_2, x_3) = (x_1 \wedge x_2) \oplus (x_1 \wedge \bar{x}_3)$



Boolean circuits

- A Boolean circuit is like a Boolean formula, except that we permit vertices to have multiple outgoing wires