# CMSC 28100

# Introduction to Complexity Theory

Autumn 2025
Instructor: William Hoza

Which problems

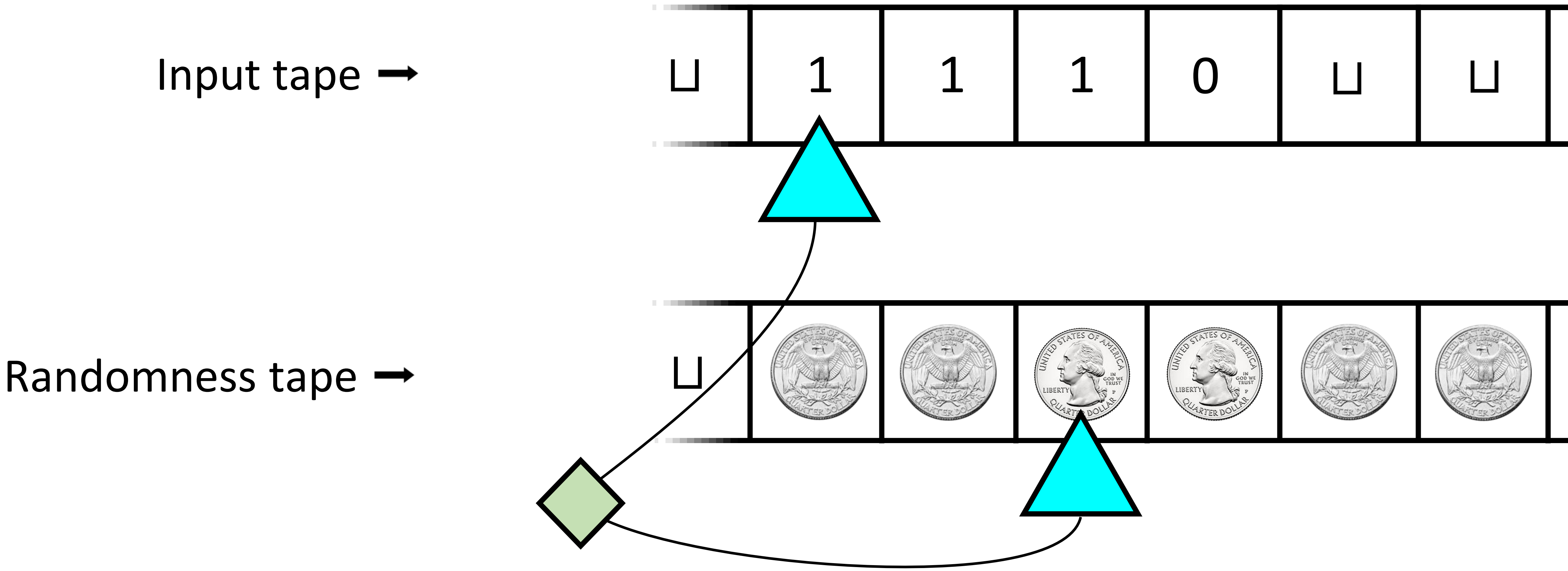
can be solved

through computation?

# Randomized computation

- Researchers often use randomness to answer questions
  - Random sampling for opinion polls
  - Randomized controlled trials in science/medicine
- What if we incorporate this ability into our computational model?

# Randomized Turing machines



# Randomized Turing machines

- Let $T : \mathbb{N} \to \mathbb{N}$ be a function (time bound)
- **Definition:** A randomized time-$T$ Turing machine is a two-tape Turing machine $M$ such that for every $n \in \mathbb{N}$, every $w \in \{0,1\}^n$, and every $x \in \{0,1\}^{T(n)}$, if we initialize $M$ with $w$ on tape 1 and $x$ on tape 2, then it halts within $T(n)$ steps
- Interpretation: $w$ is the input and $x$ is the coin tosses
- (Giving $M$ more than $T(n)$ random bits would be pointless)

# Acceptance probability

- Let $M$ be a randomized Turing machine and let $w \in \{0, 1\}^*$
- To run $M$ on $w$, we select $x \in \{0, 1\}^{T(n)}$ uniformly at random and initialize $M$ with $w$ on tape 1 and $x$ on tape 2

$$\Pr[M \text{ accepts } w] = \frac{|\{x : M \text{ accepts } w \text{ when tape 2 is initialized with } x\}|}{2^{T(n)}}$$

# Randomized polynomial time, attempt #1

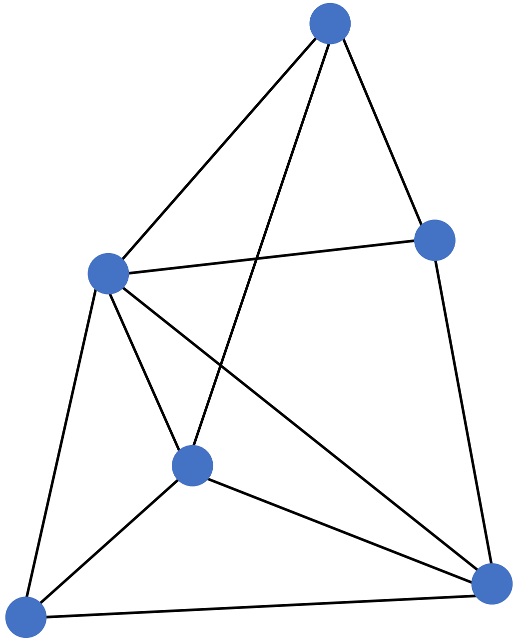
- Let $Y \subseteq \{0,1\}^*$
- **Definition:** $Y \in \mathrm{NP}$ if there exists a randomized polynomial-time Turing machine $M$ such that for every $w \in \{0,1\}^*$:
  - If $w \in Y$, then $\Pr[M \text{ accepts } w] \neq 0$
  - If $w \notin Y$, then $\Pr[M \text{ accepts } w] = 0$

"Nondeterministic Turing machine"

- "Nondeterministic Polynomial-time"

# Example: $\text{CLIQUE}$

- $\text{CLIQUE} = \{\langle G, k\rangle : G \text{ has a } k\text{-clique}\}$
- **Claim:** $\text{CLIQUE} \in \text{NP}$
- **Proof:**
  1. Pick a random subset of the vertices
  2. Check if it is a $k$-clique
  3. If yes, accept; if no, reject.



# How to interpret $\mathsf{NP}$

- Can we conclude that $\mathrm{CLIQUE}$ is tractable?
- No!
- Even if $G$ has a $k$-clique, $\Pr[\mathrm{accept}]$ might be extremely small
- When the algorithm rejects, it gives us practically no information

# How to interpret NP

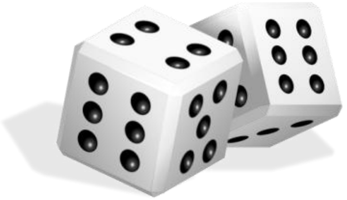
- NP is not a good model of tractability
- NP is an extremely useful conceptual tool…
- More on this later

Which problems

can be solved

through computation?

# Error probability



- Let $M$ be a randomized time-$T$ Turing machine for some $T: \mathbb{N} \to \mathbb{N}$
- Let $Y \subseteq \{0,1\}^*$ and let $\delta \in [0,1]$
- We say $M$ decides $Y$ with error probability $\delta$ if for every $w \in \{0,1\}^*$:
  - If $w \in Y$, then $\Pr[M \text{ accepts } w] \geq 1 - \delta$
  - If $w \notin Y$, then $\Pr[M \text{ accepts } w] \leq \delta$

# The complexity class $\mathrm{BPP}$

- **Definition:** $\mathrm{BPP}$ is the set of languages $Y \subseteq \{0, 1\}^*$ such that there exists a randomized polynomial-time Turing machine that decides $Y$ with error probability $1/3$
- "<u>B</u>ounded-error <u>P</u>robabilistic <u>P</u>olynomial-time"

# Amplification lemma

- Suppose a language $Y \subseteq \{0,1\}^*$ can be decided by a time-$T$ Turing machine $M_0$ with error probability $1/3$
- Let $k \in \mathbb{N}$ be any constant

**Amplification Lemma:** There exists a randomized time-$T'$ Turing machine that decides $Y$ with error probability $1/2^{n^k}$, where $T'(n) = O\left(T(n) \cdot n^k\right)$.

- As $n \to \infty$, the error probability goes to $0$ extremely rapidly!

# Proof of the amplification lemma (1 slide)

- For simplicity, assume th… 1
  - For $w \notin Y$, we merely ass…

If $M_0$ uses $R(n)$ many random bits, then how many random bits does the new TM use?

A: $R(n) + n^k$

B: $R(n) \cdot n^k$

C: $R(n)^k$

D: Not enough information

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

Given $w \in \{0,1\}^n$:

1) For $i = 1$ to $n^k$:
   a) Simulate $M_0$ on $w$ using fresh random bits. If it rejects, reject.
2) Accept.

Time complexity: $O(T(n) \cdot n^k)$

- If $w \in Y$, then $\Pr[M \text{ accepts } w] = 1$
- If $w \notin Y$, then $\Pr[M \text{ accepts } w] \leq (1/2)^{n^k} = 1/2^{n^k}$

# $\mathrm{BPP}$ as a model of tractability

- Because of the amplification lemma, languages in $\mathrm{BPP}$ should be considered "tractable"
- A mistake that occurs with probability $1/2^{100}$ can be safely ignored
- (Even if you use a deterministic algorithm, can you really be 100% certain that the computation was carried out correctly?)
- Next: An interesting example of a language in $\mathrm{BPP}$

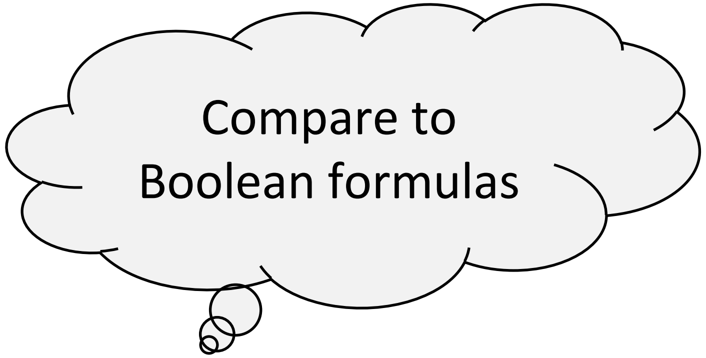
# Example: High school algebra

- "Expand and simplify: $\underbrace{(x+1)\cdot(x-1)}_{}$"

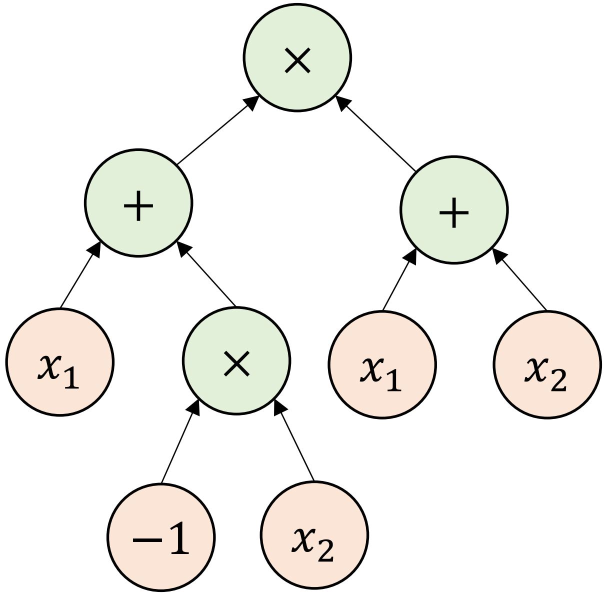
  This type of expression is called an arithmetic formula

- How difficult is this type of exercise?

# Arithmetic formulas




- **Definition:** A $k$-variate arithmetic formula is a rooted binary tree
  - Each internal node is labeled with $+$ or $\times$
  - Each leaf is labeled with a constant $c \in \mathbb{Z}$ or a variable among $x_1, \ldots, x_k$
- It computes $F: \mathbb{R}^k \to \mathbb{R}$
- E.g., $F(x_1, x_2) = (x_1 - x_2) \cdot (x_1 + x_2)$
- **Warm-up:** Let's think about the case of zero variables



# Evaluating an arithmetic formula

- **Problem:** Given an arithmetic formula with zero variables, determine whether it evaluates to 0
  - Example: $(2+3)\cdot(1-2)+5=0$
  - Example: $(2+3)\cdot(2-1)+5\neq 0$
- **As a language:**

$$\text{EQUALS-ZERO} = \{\langle F\rangle : F \text{ is a } 0\text{-variate arithmetic formula and } F \equiv 0\}$$

# Evaluating an arithmetic formula

**Lemma:** $\text{EQUALS-ZERO} \in \text{P}$

- **Proof idea:** Grade-school arithmetic
- Possible concern: How big are the numbers we are working with?

# Numbers are not getting terribly big

- Let $c_1, c_2, \dots, c_d$ be the constants at the leaves of the formula $F$
- Let $M = \max(|c_1|, |c_2|, \dots, |c_d|, 2)$
- **Claim:** $|F| \leq M^d$. Proof by induction:
  - Base case: $d = 1$: trivial ✔
  - If $F = F_L \cdot F_R$, then $|F| = |F_L| \cdot |F_R| \leq M^{d_L} \cdot M^{d_R} = M^d$
  - If $F = F_L + F_R$, then $|F| \leq |F_L| + |F_R| \leq M^{d_L} + M^{d_R} \leq M^{d_L} \cdot M^{d_R} = M^d$

# Evaluating an arithmetic formula

**Lemma:** $\text{EQUALS-ZERO} \in \text{P}$

- **Proof sketch:** Evaluate the nodes one by one, starting at the leaves
- $M \leq 2^n$ and $d \leq n$, so each node outputs $y$ such that $|y| \leq M^d \leq 2^{n^2}$
- In other words, $y$ is an $O(n^2)$-bit integer
- There are $O(n)$ nodes, and we can do arithmetic in polynomial time ✔

# Identity testing

- **Problem:** Given an arithmetic formula $F$, possibly including one or more variables, determine whether $F \equiv 0$
  - Example: $(2x + 1) \cdot 3 - 6x - 3 \equiv 0$
  - Example: $(x + 1) \cdot (x + 2) + 4 \not\equiv 0$
- **As a language:**

$$\text{IDENTICALLY-ZERO} = \{\langle F \rangle : F \text{ is an arithmetic formula and } F \equiv 0\}$$

# Complexity of identity testing

- $\text{IDENTICALLY-ZERO} = \{\langle F \rangle : F \text{ is an arithmetic formula and } F \equiv 0\}$
- **High school algorithm:** Expand $F$ into monomials, then simplify by canceling like terms

**What is the time complexity of this algorithm?**

**A:** $\text{poly}(n)$

**B:** $2^{\Omega(n)}$

**C:** $O(1)$

**D:** $\infty$

Respond at PollEv.com/whoza or text “whoza” to 22333

# Identity testing example

- Given: $F = (ab + a - b - 1) \cdot (cd - ad + a - c) \cdot (b - e) + (bd + d - b - 1) \cdot (bc + ea - ab - ce) \cdot (1 - a)$
- Expand:

$$
\begin{aligned}
F \equiv\ & ab^2cd - eabcd - a^2b^2d + ea^2bd - ab^2c + eabc + a^2b^2 - ea^2b + acdb - eacd - a^2db + ea^2d - acb \\
& + eac + a^2b - ea^2 - b^2cd + ebcd + b^2da - ebda + b^2cb - ebc - b^2a + eba - cdb + ecd + dab - eda + cb \\
& - ec - ab + ea - ea^2bd + eabd + ea^2b - eab - ea^2d + ead + ea^2 - ea + a^2b^2d - ab^2d - a^2b^2 + ab^2 \\
& + a^2db - adb - a^2b + ab - b^2cda + b^2cd + bcdea - bcde + b^2ca - b^2c - bcea + bce - cdab + cdb + cab \\
& - cb + cdea - cde - cea + ce
\end{aligned}
$$

- Everything cancels out: $F \equiv 0$

# Complexity of identity testing

- Expanding $F$ takes $2^{\Omega(n)}$ time in some cases 😦
- E.g., $F = (x+y)\cdot(x+y)\cdot(x+y)\cdots(x+y)$