Review for CMSC 452 Final: P and NP

Turing Machines Def

We will **not** define *Turing Machine* until we need to (after midterm).

Here is all you need to know:

Turing Machines Def

We will **not** define *Turing Machine* until we need to (after midterm).

Here is all you need to know:

1. Everything computable is computable by a Turing machine.

Turing Machines Def

We will **not** define *Turing Machine* until we need to (after midterm).

Here is all you need to know:

1. Everything computable is computable by a Turing machine.
2. Turing machines compute with discrete steps so one can talk about how many steps a computation takes.

Turing Machines Def

We will **not** define *Turing Machine* until we need to (after midterm).

Here is all you need to know:

1. Everything computable is computable by a Turing machine.
2. Turing machines compute with discrete steps so one can talk about how many steps a computation takes.
3. There are many different models of computation. They are all equivalent to Turing machines. And better- they are all equivalent within poly time.

Polynomial Time and Other Classes

Def

Polynomial Time and Other Classes

Def

1. $\mathrm{P} = \mathrm{DTIME}(n^{O(1)})$.

Polynomial Time and Other Classes

Def

1. $\mathrm{P} = \mathrm{DTIME}(n^{O(1)})$.
2. $\mathrm{EXP} = \mathrm{DTIME}(2^{n^{O(1)}})$.

Polynomial Time and Other Classes

Def

1. $\mathrm{P} = \mathrm{DTIME}(n^{O(1)})$.
2. $\mathrm{EXP} = \mathrm{DTIME}(2^{n^{O(1)}})$.
3. PF is the set of a **functions** computable in poly time.

Polynomial Time and Other Classes

Def

1. $\mathrm{P} = \mathrm{DTIME}(n^{O(1)})$.
2. $\mathrm{EXP} = \mathrm{DTIME}(2^{n^{O(1)}})$.
3. PF is the set of a **functions** computable in poly time.

These definitions are model independent.

$\mathbf{3SAT}, \mathbf{HAM}, \mathbf{EUL}, \mathbf{CLIQ}, \mathbf{3COL}$ All Walk into a Bar

We rewrite $\mathrm{3SAT}$, HAM, EUL.

$\mathrm{3SAT}, \mathrm{HAM}, \mathrm{EUL}, \mathrm{CLIQ}, \mathrm{3COL}$ All Walk into a Bar

We rewrite 3SAT, HAM, EUL.

$$\mathrm{3SAT} = \{\phi : (\exists \vec{b})[\phi(\vec{b}) = T]\}$$

$\mathbf{3SAT, HAM, EUL, CLIQ, 3COL}$ All Walk into a Bar

We rewrite 3SAT, HAM, EUL.

$$\mathrm{3SAT} = \{\phi : (\exists \vec{b})[\phi(\vec{b}) = T]\}$$

$$\mathrm{HAM} = \{G : (\exists v_1, \ldots, v_n)[v_1, \ldots, v_n \text{ is a Ham Cycle}]\}.$$

$\mathbf{3SAT, HAM, EUL, CLIQ, 3COL}$ All Walk into a Bar

We rewrite $\mathrm{3SAT}$, HAM, EUL.

$$\mathrm{3SAT} = \{\phi : (\exists \vec{b})[\phi(\vec{b}) = T]\}$$

$$\mathrm{HAM} = \{G : (\exists v_1, \ldots, v_n)[v_1, \ldots, v_n \text{ is a Ham Cycle}]\}.$$

$$\mathrm{EUL} = \{G : (\exists v_1, \ldots, v_n)[v_1, \ldots, v_n \text{ is an Eul Cycle}]\}.$$

$3\mathrm{SAT}, \mathrm{HAM}, \mathrm{EUL}, \mathrm{CLIQ}, 3\mathrm{COL}$ All Walk into a Bar

We rewrite 3SAT, HAM, EUL.

$$3\mathrm{SAT} = \{\phi : (\exists \vec{b})[\phi(\vec{b}) = T]\}$$

$$\mathrm{HAM} = \{G : (\exists v_1, \ldots, v_n)[v_1, \ldots, v_n \text{ is a Ham Cycle}]\}.$$

$$\mathrm{EUL} = \{G : (\exists v_1, \ldots, v_n)[v_1, \ldots, v_n \text{ is an Eul Cycle}]\}.$$

$$\mathrm{CLIQ} = \{(G, k) : (\exists v_1, \ldots, v_k)[v_1, \ldots, v_k \text{ are a Clique}]\}.$$

$\mathbf{3SAT}, \mathbf{HAM}, \mathbf{EUL}, \mathbf{CLIQ}, \mathbf{3COL}$ All Walk into a Bar

We rewrite $\mathrm{3SAT}$, HAM, EUL.

$$\mathrm{3SAT} = \{\phi : (\exists \vec{b})[\phi(\vec{b}) = T]\}$$

$$\mathrm{HAM} = \{G : (\exists v_1, \ldots, v_n)[v_1, \ldots, v_n \text{ is a Ham Cycle}]\}.$$

$$\mathrm{EUL} = \{G : (\exists v_1, \ldots, v_n)[v_1, \ldots, v_n \text{ is an Eul Cycle}]\}.$$

$$\mathrm{CLIQ} = \{(G, k) : (\exists v_1, \ldots, v_k)[v_1, \ldots, v_k \text{ are a Clique}]\}.$$

For the above sets: If x is a member then there is a short verifiable witness of this.

NP

Def A is in NP if there exists a set $B \in \mathrm{P}$ and a polynomial p such that

$$A = \{x : (\exists y)[|y| = p(|x|) \wedge (x, y) \in B]\}.$$

NP

Def A is in NP if there exists a set $B \in \mathrm{P}$ and a polynomial p such that

$$A = \{x : (\exists y)[|y| = p(|x|) \wedge (x, y) \in B]\}.$$

Intuition. Let $A \in \mathrm{NP}$.

NP

Def A is in NP if there exists a set $B \in \mathrm{P}$ and a polynomial p such that

$$A = \{x : (\exists y)[|y| = p(|x|) \wedge (x, y) \in B]\}.$$

Intuition. Let $A \in \mathrm{NP}$.

- If $x \in A$ then there is a SHORT (poly in $|x|$) proof of this fact, namely y, such that x can be VERIFIED in poly time.

NP

Def A is in NP if there exists a set $B \in \mathrm{P}$ and a polynomial p such that

$$A = \{x : (\exists y)[|y| = p(|x|) \wedge (x, y) \in B]\}.$$

Intuition. Let $A \in \mathrm{NP}$.

- If $x \in A$ then there is a SHORT (poly in $|x|$) proof of this fact, namely y, such that x can be VERIFIED in poly time.
- So if I wanted to convince you that $x \in A$, I could give you y. You can verify $(x, y) \in B$ easily and be convinced.

NP

Def A is in NP if there exists a set $B \in \mathrm{P}$ and a polynomial p such that

$$A = \{x : (\exists y)[|y| = p(|x|) \wedge (x, y) \in B]\}.$$

Intuition. Let $A \in \mathrm{NP}$.

- If $x \in A$ then there is a SHORT (poly in $|x|$) proof of this fact, namely y, such that x can be VERIFIED in poly time.
- So if I wanted to convince you that $x \in A$, I could give you y. You can verify $(x, y) \in B$ easily and be convinced.
- If $x \notin A$ then there is NO proof that $x \in A$.

Note $\mathrm{3SAT}$, HAM, EUL, CLIQ are all in NP.

Reductions and Cook-Levin

Def Let X, Y be sets. $X \leq Y$ means there is a poly-time function f:

$$x \in X \text{ iff } f(x) \in Y.$$

Reductions and Cook-Levin

Def Let X, Y be sets. $X \leq Y$ means there is a poly-time function f:

$$x \in X \text{ iff } f(x) \in Y.$$

Def A set Y is **NP-complete (NPC)** if the following hold:

- $Y \in \mathrm{NP}$
- If $X \in \mathrm{NP}$ then $X \leq Y$.

Reductions and Cook-Levin

Def Let X, Y be sets. $X \leq Y$ means there is a poly-time function f:

$$x \in X \text{ iff } f(x) \in Y.$$

Def A set Y is **NP-complete (NPC)** if the following hold:

- $Y \in \mathrm{NP}$
- If $X \in \mathrm{NP}$ then $X \leq Y$.

Cook-Levin Theorem $\mathrm{3SAT}$ is NP-complete.

Reductions and Cook-Levin

Def Let X, Y be sets. $X \leq Y$ means there is a poly-time function f:

$$x \in X \text{ iff } f(x) \in Y.$$

Def A set Y is **NP-complete (NPC)** if the following hold:

- $Y \in \mathrm{NP}$
- If $X \in \mathrm{NP}$ then $X \leq Y$.

Cook-Levin Theorem $\mathrm{3SAT}$ is NP-complete.

Since then thousands of problems have been shown NP-complete.

SAT, HAM, CLIQ, 3COL Walk into a Bar

SAT, HAM, CLIQ, 3COL Walk into a Bar

1. SAT is NP-complete by Cook-Levin Theorem.

$\mathrm{SAT}, \mathrm{HAM}, \mathrm{CLIQ}, 3\mathrm{COL}$ Walk into a Bar

1. SAT is NP-complete by Cook-Levin Theorem.
2. IS is NP-complete. We proved this by showing $3\mathrm{SAT} \le \mathrm{IS}$.

SAT, HAM, CLIQ, 3COL Walk into a Bar

1. SAT is NP-complete by Cook-Levin Theorem.
2. IS is NP-complete. We proved this by showing $3\mathrm{SAT} \le \mathrm{IS}$.
3. 3COL is NP-complete. We proved this.

SAT, HAM, CLIQ, 3COL Walk into a Bar

1. SAT is NP-complete by Cook-Levin Theorem.
2. IS is NP-complete. We proved this by showing $3SAT \le IS$.
3. 3COL is NP-complete. We proved this.
4. HAM is NP-complete. Just take my word for it.

Closure of P

Closure Properties of P

Assume $L_1, L_2 \in \mathrm{P}$.

Closure Properties of P

Assume $L_1, L_2 \in \mathrm{P}$.

1. $L_1 \cup L_2 \in \mathrm{P}$. EASY. Uses polys closed under addition.

Closure Properties of P

Assume $L_1, L_2 \in \mathrm{P}$.

1. $L_1 \cup L_2 \in \mathrm{P}$. EASY. Uses polys closed under addition.
2. $L_1 \cap L_2 \in \mathrm{P}$. EASY. Uses polys closed under addition.

Closure Properties of P

Assume $L_1, L_2 \in \mathrm{P}$.

1. $L_1 \cup L_2 \in \mathrm{P}$. EASY. Uses polys closed under addition.
2. $L_1 \cap L_2 \in \mathrm{P}$. EASY. Uses polys closed under addition.
3. $\overline{L_1} \in \mathrm{P}$. EASY.

Closure Properties of P

Assume $L_1, L_2 \in \mathrm{P}$.

1. $L_1 \cup L_2 \in \mathrm{P}$. EASY. Uses polys closed under addition.
2. $L_1 \cap L_2 \in \mathrm{P}$. EASY. Uses polys closed under addition.
3. $\overline{L_1} \in \mathrm{P}$. EASY.
4. $L_1 L_2 \in \mathrm{P}$. EASY. Uses $p(n)$ poly then $np(n)$ poly.

Closure Properties of P

Assume $L_1, L_2 \in \mathrm{P}$.

1. $L_1 \cup L_2 \in \mathrm{P}$. EASY. Uses polys closed under addition.
2. $L_1 \cap L_2 \in \mathrm{P}$. EASY. Uses polys closed under addition.
3. $\overline{L_1} \in \mathrm{P}$. EASY.
4. $L_1 L_2 \in \mathrm{P}$. EASY. Uses $p(n)$ poly then $np(n)$ poly.
5. $L_1^* \in \mathrm{P}$. HARDER- Used Dyna Programmming.

Closure of NP

Closure Properties of NP

Assume $L_1, L_2 \in \mathrm{NP}$.

Closure Properties of NP

Assume $L_1, L_2 \in \mathrm{NP}$.

1. $L_1 \cup L_2 \in \mathrm{NP}$. EASY. Uses polys closed under addition.

Closure Properties of NP

Assume $L_1, L_2 \in \mathrm{NP}$.

1. $L_1 \cup L_2 \in \mathrm{NP}$. EASY. Uses polys closed under addition.
2. $L_1 \cap L_2 \in \mathrm{NP}$. EASY. Uses polys closed under addition.

Closure Properties of NP

Assume $L_1, L_2 \in \mathrm{NP}$.

1. $L_1 \cup L_2 \in \mathrm{NP}$. EASY. Uses polys closed under addition.
2. $L_1 \cap L_2 \in \mathrm{NP}$. EASY. Uses polys closed under addition.
3. $\overline{L_1} \in \mathrm{NP}$. THOUGHT TO BE FALSE.

Closure Properties of NP

Assume $L_1, L_2 \in \mathrm{NP}$.

1. $L_1 \cup L_2 \in \mathrm{NP}$. EASY. Uses polys closed under addition.
2. $L_1 \cap L_2 \in \mathrm{NP}$. EASY. Uses polys closed under addition.
3. $\overline{L_1} \in \mathrm{NP}$. THOUGHT TO BE FALSE.
4. $L_1 L_2 \in \mathrm{NP}$. EASY. Uses polys closed under addition and mult.

Closure Properties of NP

Assume $L_1, L_2 \in \mathrm{NP}$.

1. $L_1 \cup L_2 \in \mathrm{NP}$. EASY. Uses polys closed under addition.
2. $L_1 \cap L_2 \in \mathrm{NP}$. EASY. Uses polys closed under addition.
3. $\overline{L_1} \in \mathrm{NP}$. THOUGHT TO BE FALSE.
4. $L_1 L_2 \in \mathrm{NP}$. EASY. Uses polys closed under addition and mult.
5. $L_1^* \in \mathrm{NP}$. EASY. Uses polys closed under addition and mult.