Foundations of Computing

Lab 11 – $\mathcal{P}$, $\mathcal{NP}$, $co-\mathcal{NP}$

April 17, 2024

Outline

Boolean Formulas

- Boolean formula: A Boolean formula of size n is a logic equation with n letters (e.g., $x_1, \overline{x_1}$)

$$(x_1 \vee x_2 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_1}) \wedge (x_2 \vee \overline{x_3})$$

Boolean Formulas

- Boolean formula: A Boolean formula of size n is a logic equation with n letters (e.g., $x_1, \overline{x_1}$)

$$(x_1 \vee x_2 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_1}) \wedge (x_2 \vee \overline{x_3})$$

- An assignment is some assignment of 0 and 1 to the variables:

$$x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 1$$

Boolean Formulas

- Boolean formula: A Boolean formula of size n is a logic equation with n letters (e.g., $x_1, \overline{x_1}$)

$$(x_1 \vee x_2 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_1}) \wedge (x_2 \vee \overline{x_3})$$

- An assignment is some assignment of 0 and 1 to the variables:

$$x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 1$$

- An assignment satsfies a formula ϕ if ϕ evaluates to 1 with the variables fixed according to the assignment

Boolean Formulas

- Boolean formula: A Boolean formula of size n is a logic equation with n letters (e.g., $x_1, \overline{x_1}$)

$$(x_1 \vee x_2 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_1}) \wedge (x_2 \vee \overline{x_3})$$

- An assignment is some assignment of 0 and 1 to the variables:

$$x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 1$$

- An assignment satsfies a formula ϕ if ϕ evaluates to 1 with the variables fixed according to the assignment
- A formula may have 0, 1, or more satisfying assigments

Boolean Formulas

- Boolean formula: A Boolean formula of size n is a logic equation with n letters (e.g., $x_1, \overline{x_1}$)

$$(x_1 \vee x_2 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_1}) \wedge (x_2 \vee \overline{x_3})$$

- An assignment is some assignment of 0 and 1 to the variables:

$$x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 1$$

- An assignment satsfies a formula ϕ if ϕ evaluates to 1 with the variables fixed according to the assignment
- A formula may have 0, 1, or more satisfying assigments
- If ϕ has 0 satisfying assignments, it is unsatisfiable

Boolean Formulas

- Boolean formula: A Boolean formula of size n is a logic equation with n letters (e.g., $x_1, \overline{x_1}$)

$$(x_1 \vee x_2 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_1}) \wedge (x_2 \vee \overline{x_3})$$

- An assignment is some assignment of 0 and 1 to the variables:

$$x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 1$$

- An assignment satsfies a formula ϕ if ϕ evaluates to 1 with the variables fixed according to the assignment
- A formula may have 0, 1, or more satisfying assigments
- If ϕ has 0 satisfying assignments, it is unsatisfiable
- Formula ϕ is in CNF if it is an AND-of-ORs formula

$$(x_1 \vee x_2) \wedge (x_3 \vee x_4)$$

Why Boolean Formulas

- Boolean formulas give us many interesting problems to study
- Formulas are easy to reason about, and you've seend them before.
- SAT, 3-SAT are $\mathcal{NP}$-complete

Outline

The Class $\mathcal{P}$

Informal: $\mathcal{P}$ is the class of problems we can solve in poly-time

The Class $\mathcal{P}$

Informal: $\mathcal{P}$ is the class of problems we can solve in poly-time

Formal: A language $L \in \mathcal{P}$ if there exists a poly-time DTM M such that M decides the language L

- $M(x) = 1 \iff x \in L$
- M halts on ALL x and gives the correct answer
- M halts in time $poly(|x|)$

The Class $\mathcal{P}$

Informal: $\mathcal{P}$ is the class of problems we can solve in poly-time

Formal: A language $L \in \mathcal{P}$ if there exists a poly-time DTM M such that M decides the language L

- $M(x) = 1 \iff x \in L$
- M halts on ALL x and gives the correct answer
- M halts in time $poly(|x|)$

$\mathcal{P}$ captures the class of problems we can solve efficiently

The Class $\mathcal{P}$

Informal: $\mathcal{P}$ is the class of problems we can solve in poly-time

Formal: A language $L \in \mathcal{P}$ if there exists a poly-time DTM M such that M decides the language L

- $M(x) = 1 \iff x \in L$
- M halts on ALL x and gives the correct answer
- M halts in time $poly(|x|)$

$\mathcal{P}$ captures the class of problems we can solve efficiently

To show that a problem is in $\mathcal{P}$ you give a poly-time algorithm for it.

The Class $\mathcal{P}$

Informal: $\mathcal{P}$ is the class of problems we can solve in poly-time

Formal: A language $L \in \mathcal{P}$ if there exists a poly-time DTM M such that M decides the language L

- $M(x) = 1 \iff x \in L$
- M halts on ALL x and gives the correct answer
- M halts in time $poly(|x|)$

$\mathcal{P}$ captures the class of problems we can solve efficiently

To show that a problem is in $\mathcal{P}$ you give a poly-time algorithm for it.

Exercise:
Show that 2-COLORING $\in \mathcal{P}$
2-Coloring is the problem given a graph G can you color it with 2 colors such that no edge has the same color on both ends.

The Class $\mathcal{NP}$

Informal: $\mathcal{NP}$ is the class of problems that have solutions that can be verified in poly-time.

The Class $\mathcal{NP}$

Informal: $\mathcal{NP}$ is the class of problems that have solutions that can be verified in poly-time.

Formal: A language $L \in \mathcal{NP}$ if there exists a poly-time verifier V such that

- $\forall x \in L, \exists w$ s.t. $V(x, w) = 1$
- $\forall x \notin L, \forall w, V(x, w) = 0$

The Class $\mathcal{NP}$

Informal: $\mathcal{NP}$ is the class of problems that have solutions that can be verified in poly-time.

Formal: A language $L \in \mathcal{NP}$ if there exists a poly-time verifier V such that

- $\forall x \in L, \exists w$ s.t. $V(x, w) = 1$
- $\forall x \notin L, \forall w, V(x, w) = 0$

Example Problems: SAT, 3-SAT, 3-Coloring, Vertex Cover, etc.

The Class co-$\mathcal{NP}$

Informal: co-$\mathcal{NP}$ is the class of problems where we can verify that $x \notin L$ in poly-time.

The Class co-$\mathcal{NP}$

Informal: co-$\mathcal{NP}$ is the class of problems where we can verify that $x \notin L$ in poly-time.

- co-$\mathcal{NP}$ is thus the complement of $\mathcal{NP}$

The Class co-$\mathcal{NP}$

Informal: co-$\mathcal{NP}$ is the class of problems where we can verify that $x \notin L$ in poly-time.

- co-$\mathcal{NP}$ is thus the complement of $\mathcal{NP}$

Formal: A language $L \in$co-$\mathcal{NP}$ if there exists a poly-time verifier V s.t.

- $\forall x \in L, \forall w, V(x, w) = 0$
- $\forall x \notin L, \exists w$ s.t. $V(x, w) = 1$

The Class co-$\mathcal{NP}$

Informal: co-$\mathcal{NP}$ is the class of problems where we can verify that $x \notin L$ in poly-time.

- co-$\mathcal{NP}$ is thus the complement of $\mathcal{NP}$

Formal: A language $L \in$co-$\mathcal{NP}$ if there exists a poly-time verifier V s.t.

- $\forall x \in L, \forall w, V(x, w) = 0$
- $\forall x \notin L, \exists w$ s.t. $V(x, w) = 1$

Exercise:
Give some examples of languages in co-$\mathcal{NP}$, and justify why they are in co-$\mathcal{NP}$.