CMPT307: Complexity Classes: $\mathcal{P}$ and $\mathcal{NP}$

Week 13-1

Xian Qiu

Simon Fraser University

xianq@sfu.ca

Strings and Languages

- an alphabet Σ is a finite set of symbols

$$\{0,1\}, \quad \{\mathsf{T},\mathsf{F}\}, \quad \{a,b,\ldots,z\}, \quad \mathbb{N}$$

- a string x is a finite sequence of symbols from some alphabet

$$10011010010, \quad \text{asdfghjkl}, \quad \epsilon \text{ (empty string)}$$

- a language L over an alphabet Σ is a set of strings made up of symbols from Σ
- $\Sigma^* =$ the language of all strings over Σ

$$\Sigma = \{0,1\}, \quad \Sigma^* = \{\epsilon, 0, 1, 01, 10, 11, 000, \ldots\}$$

Decision Problems

Hamiltonian-Cycle

- given: undirected graph $G = (V, E)$
- question: does G contain a Hamiltonian cycle?

we can characterize a decision problem by its yes-instances

$$L = \left\{ G \;\middle|\; \begin{array}{l} G \text{ is an undirected graph} \\ G \text{ contains a Hamiltonian cycle} \end{array} \right\}$$

- binary encoding $\Sigma = \{0, 1\}$
- a decision problem Q can be viewed as a language L over Σ such that $L = \{x \in \Sigma^* \mid Q(x) = 1\}$
- Hamiltonian Cycle: given $x \in \Sigma^*$, $x \in L$?

Decision Problems

TSP

- given: undirected graph $G = (V, E)$ and distance function $c : E \to \mathbb{Z}^+$
- question: does there exist a Hamiltonian cycle in G of total length $\leq k$?

PRIME

given a natural number n, is n a prime number?

PATH

- given: undirected graph $G = (V, E)$ and $s, t \in V$ and an integer $k > 0$
- question: does there exist an s-t path with no more than k edges?

Class $\mathcal{P}$

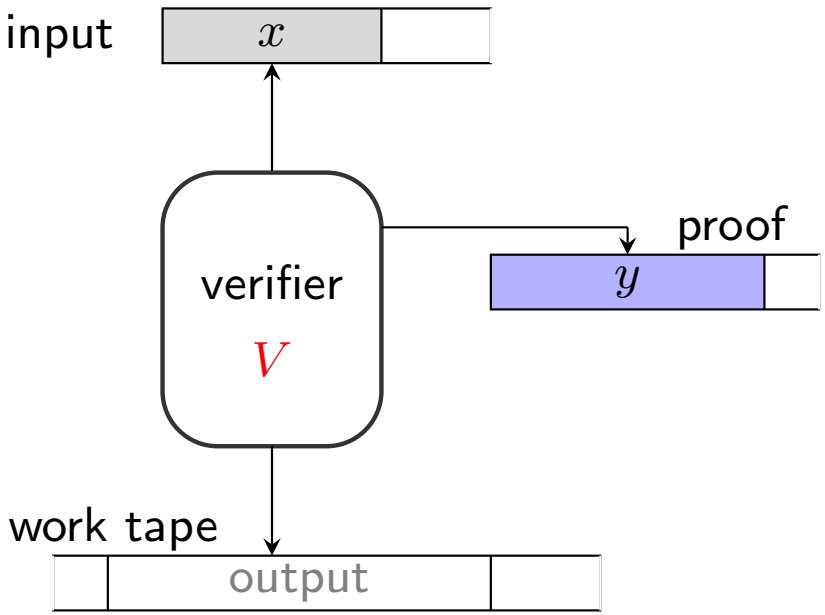
- given language L and $x \in \Sigma^* = \{0,1\}^*$, is $x \in L$?
- algorithm $\mathbb{A}$ accepts a string $x \in \Sigma^*$ if the output $\mathbb{A}(x) = 1$

$$\mathcal{P} = \left\{ L \;\middle|\; \begin{array}{l} \mathbb{A} \text{ is a polynomial time algorithm} \\ x \in L \Leftrightarrow \mathbb{A} \text{ accepts } x \end{array} \right\}$$

- $\mathcal{P}$ is the set of decision problems which can be solved in polynomial time
- $\textsc{Path} \in \mathcal{P}$
- what about $\textsc{Tsp}$? poly-time algorithm not known, but a proof can be verified in poly-time

A Proof System

- statement: $x \in L$ or $x \notin L$
- prover: writes down a proof y
- verifier: checks the statement/proof, accepting or rejecting



Class $\mathcal{NP}$

- verifier $\mathbb{V}$: polynomial time algorithm
- prover $\mathbb{P}$: arbitrarily powerful

Class $\mathcal{NP}$ (non-deterministic polynomial-time)

a language $L \in \mathcal{NP}$ if and only if

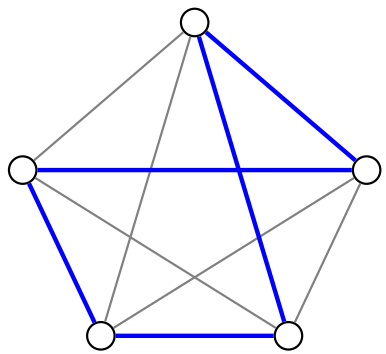
- $\forall x \in L$, $\mathbb{P}$ can write a proof y of length poly($|x|$) that $\mathbb{V}$ accepts
- $\forall x \notin L$, no matter what poly($|x|$)-length proof $\mathbb{P}$ writes, $\mathbb{V}$ rejects

- a certificate is a proof y such that $\mathbb{V}$ accepts x
- $\mathcal{NP}$ is a class of decision problems which admit short and checkable certificate
- $\mathcal{P} \subseteq \mathcal{NP}$

Examples

Hamiltonian-Cycle $\in \mathcal{NP}$

- $\mathbb{P}$ writes down a proof y (of polynomial size in $|G|$)
- $\mathbb{V}$ checks y and returns "yes" if y is a cycle and y is Hamiltonian
- a certificate is a Hamiltotian cycle



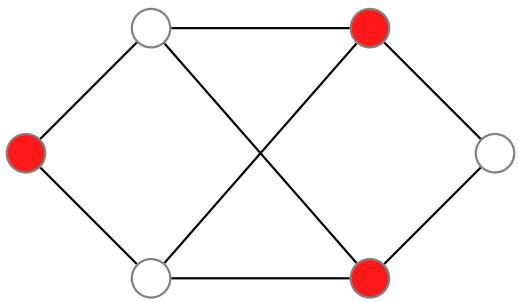
- short: Hamiltotian cycle has size $O(n)$
- checkable: $\mathbb{V}$ can verify a Hamiltonian cycle in $O(n)$

Examples

VERTEX-COVER $\in \mathcal{NP}$

- input: undirected graph G and integer $k > 0$
- question: does exist a vertex cover of G with size $\leq k$?

- $C \subseteq V$ is a vertex cover if C "covers" E



- a certificate is a vertex set C s.t. $|C| \leq k$ and C covers E
- $\mathbb{V}$ can check whether C covers E in poly-time

Examples

PRIME $\in \mathcal{NP}$?

- ▷ yes, but a certificate is non-trivial Pratt, SIAM J. on Computing, 1975

CO-PRIME: is $n \in \mathbb{N}$ not a prime number?

- ▷ CO-PRIME consists of all no-instances of PRIME
- ▷ the language of CO-PRIME is the complement of the language of PRIME
- ▷ a short, checkable certificate is a number $d \in \mathbb{N}$ s.t. $\frac{n}{d} = 0$
- ▷ CO-PRIME $\in \mathcal{NP}$

Class co-$\mathcal{NP}$

let $\bar{L}$ be the complement of L

$$\text{co-}\mathcal{NP} = \{L \mid \bar{L} \in \mathcal{NP}\}$$

CO-HAMILTONIAN-CYCLE

does G not contain a Hamiltonian cycle?

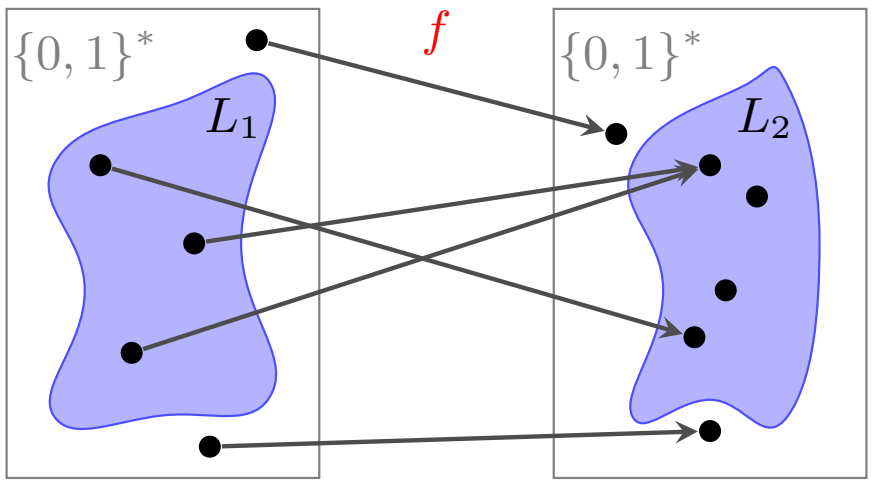
- the language of CO-HAMILTONIAN-CYCLE is the set of graphs which do not have a Hamiltonian cycle
- CO-HAMILTONIAN-CYCLE $\in \mathcal{NP}$? unknown
- PRIME $\in$ co-$\mathcal{NP}$

Polynomial Time Reduction

$L_1, L_2 =$ languages

L_1 is polynomially reducible to L_2 if

- $\exists$ poly-time computable function $f : \{0,1\}^* \to \{0,1\}^*$
- $x \in L_1 \Leftrightarrow f(x) \in L_2$



denoted by $L_1 \leq_{\mathsf{P}} L_2$

NP-completeness

assume $L_1 \leq_{\mathsf{P}} L_2$ and consider decision problems

$$P : x_1 \in L_1? \quad Q : x_2 \in L_2?$$

- P is polynomially reducible to Q
- Q is as hard as P

a language $L \subseteq \{0,1\}^*$ is NP-complete if

1. $L \in \mathcal{NP}$ and
2. $L' \leq_{\mathsf{P}} L$ for every $L' \in \mathcal{NP}$

- if L is NP-complete, then the associated decision problem is the hardest among those in $\mathcal{NP}$
- L is NP-hard if 2 is satisfied (1 not necessarily)

NP-completeness

"I can't find an efficient algorithm, but neither can all these famous people." Garey & Johnson (1979): Computers and Intractability