CS310 : Automata Theory 2019

Lecture 38: Efficiency in computation
Classifying problems by their complexity

Instructor: S. Akshay

IITB, India

09-04-2019

Recap

Turing machines and computability

1. Turing machines
    (i) Definition & variants
    (ii) Decidable and Turing recognizable languages
    (iii) Church-Turing Hypothesis

Recap

Turing machines and computability

1. Turing machines
   (i) Definition & variants
   (ii) Decidable and Turing recognizable languages
   (iii) Church-Turing Hypothesis
2. Undecidability
   (i) A proof technique by diagonalization
   (ii) Via reductions
   (iii) Rice's theorem

Recap

Turing machines and computability

1. Turing machines
   (i) Definition & variants
   (ii) Decidable and Turing recognizable languages
   (iii) Church-Turing Hypothesis
2. Undecidability
   (i) A proof technique by diagonalization
   (ii) Via reductions
   (iii) Rice's theorem
3. Applications: showing (un)decidability of other problems
   (i) A string matching problem: Post's Correspondance Problem
   (ii) A problem for compilers: Unambiguity of Context-free languages
   (iii) Between TM and PDA: Linear Bounded Automata

Recap

Turing machines and computability

1. Turing machines
   (i) Definition & variants
   (ii) Decidable and Turing recognizable languages
   (iii) Church-Turing Hypothesis
2. Undecidability
   (i) A proof technique by diagonalization
   (ii) Via reductions
   (iii) Rice's theorem
3. Applications: showing (un)decidability of other problems
   (i) A string matching problem: Post's Correspondance Problem
   (ii) A problem for compilers: Unambiguity of Context-free languages
   (iii) Between TM and PDA: Linear Bounded Automata
4. Efficiency in computation: run-time complexity.
   (i) Running time complexity
   (ii) Polynomial and exponential time complexity

The class P

So, k-tape to 1-tape involves a polynomial blow-up, while non-det to det requires an exponential blow-up.

Definition

P is the class of languages that are decidable in polynomial time on a deterministic single-tape Turing machine, i.e.,

$$P = \bigcup_k TIME(n^k)$$

The class P

So, k-tape to 1-tape involves a polynomial blow-up, while non-det to det requires an exponential blow-up.

Definition

P is the class of languages that are decidable in polynomial time on a deterministic single-tape Turing machine, i.e.,

$$P = \bigcup_k TIME(n^k)$$

Examples:

- Given a graph G, is there a path from s to t?

The class P

Definition

P is the class of languages that are decidable in polynomial time on a deterministic single-tape Turing machine, i.e.,

$$P = \bigcup_k TIME(n^k)$$

Examples:

- Given a graph G, is there a path from s to t?
- Are two given numbers relatively prime?

Examples of problems in P

PATH: Given directed graph $G = (V, E)$ and nodes s, t, is there a path between s and t

Examples of problems in P

PATH: Given directed graph $G = (V, E)$ and nodes s, t, is there a path between s and t

Brute force algo?

Examples of problems in P

PATH: Given directed graph $G = (V, E)$ and nodes s, t, is there a path between s and t

- Mark s
- Repeat until no additional nodes are marked:
- scan all edges of G and if (a, b) is an edge with a marked and b unmarked, then mark b,
- if t is marked, accept, else reject.

Examples of problems in P

PATH: Given directed graph $G = (V, E)$ and nodes s, t, is there a path between s and t

- Mark s
- Repeat until no additional nodes are marked: at most $|V|$ times
- scan all edges of G and if (a, b) is an edge with a marked and b unmarked, then mark b,
- if t is marked, accept, else reject.

Examples of problems in P

PATH: Given directed graph $G = (V, E)$ and nodes s, t, is there a path between s and t

RELPRIME: Given $x, y \in \mathbb{N}$, is $gcd(x, y) = 1$

Examples of problems in P

PATH: Given directed graph $G = (V, E)$ and nodes s, t, is there a path between s and t

RELPRIME: Given $x, y \in \mathbb{N}$, is $gcd(x, y) = 1$

Euclid's algo!

- repeat till $y = 0$;
- assign $x := x \mod y$
- exchange x and y;
- At end if result is $x = 1$ accept, else reject.

Examples of problems in P

PATH: Given directed graph $G = (V, E)$ and nodes s, t, is there a path between s and t

RELPRIME: Given $x, y \in \mathbb{N}$, is $gcd(x, y) = 1$

Euclid's algo!

- repeat till $y = 0$;how many times is this done?
- assign $x := x \mod y$
- exchange x and y;
- At end if result is $x = 1$ accept, else reject.

The class EXP

Definition

EXP is the class of languages that are decidable in exponential time on a deterministic single-tape Turing machine, i.e.,

$$EXP = \bigcup_k TIME(2^{n^k})$$

The class EXP

Definition

EXP is the class of languages that are decidable in exponential time on a deterministic single-tape Turing machine, i.e.,

$$EXP = \bigcup_k TIME(2^{n^k})$$

Examples

The class EXP

Definition

EXP is the class of languages that are decidable in exponential time on a deterministic single-tape Turing machine, i.e.,

$$EXP = \bigcup_k TIME(2^{n^k})$$

Examples

- All P time problems! i.e., $P \subseteq EXP$.

The class EXP

Definition

EXP is the class of languages that are decidable in exponential time on a deterministic single-tape Turing machine, i.e.,

$$EXP = \bigcup_k TIME(2^{n^k})$$

Examples

- All P time problems! i.e., $P \subseteq EXP$.
- HAMILTONIAN-PATH: G, s, t: is there a path from s to t that goes through each node of G exactly once?

The class EXP

Definition

EXP is the class of languages that are decidable in exponential time on a deterministic single-tape Turing machine, i.e.,

$$EXP = \bigcup_k TIME(2^{n^k})$$

Examples

- All P time problems! i.e., $P \subseteq EXP$.
- HAMILTONIAN-PATH: G, s, t: is there a path from s to t that goes through each node of G exactly once?
- (Generalized) CHESS

The class EXP

Definition

EXP is the class of languages that are decidable in exponential time on a deterministic single-tape Turing machine, i.e.,

$$EXP = \bigcup_k TIME(2^{n^k})$$

Examples

- All P time problems! i.e., $P \subseteq EXP$.
- HAMILTONIAN-PATH: G, s, t: is there a path from s to t that goes through each node of G exactly once?
- (Generalized) CHESS
- COMPOSITIES: is a number composite?

The class NP

$NTIME(t(n))$ is set of all languages decidable by a $O(t(n))$ 1-tape non-det Turing machine

The class NP

$NTIME(t(n))$ is set of all languages decidable by a $O(t(n))$ 1-tape non-det Turing machine

Definition

NP is the class of languages that are decidable in polynomial time on a non-deterministic single-tape Turing machine, i.e.,

$$NP = \bigcup_k NTIME(n^k)$$

The class NP

$NTIME(t(n))$ is set of all languages decidable by a $O(t(n))$ 1-tape non-det Turing machine

Definition

NP is the class of languages that are decidable in polynomial time on a non-deterministic single-tape Turing machine, i.e.,

$$NP = \bigcup_k NTIME(n^k)$$

Verifier

for language A is an algorithm V s.t. $w \in A$ iff V accepts $\langle w, c \rangle$ for some witness or proof string c.

The class NP

NTIME$(t(n))$ is set of all languages decidable by a $O(t(n))$ 1-tape non-det Turing machine

Definition

NP is the class of languages that are decidable in polynomial time on a non-deterministic single-tape Turing machine, i.e.,

$$NP = \bigcup_k NTIME(n^k)$$

Verifier

for language A is an algorithm V s.t. $w \in A$ iff V accepts $\langle w, c \rangle$ for some witness or proof string c. Examples: HAMPATH, Composities

The class NP

$NTIME(t(n))$ is set of all languages decidable by a $O(t(n))$ 1-tape non-det Turing machine

Definition

NP is the class of languages that are decidable in polynomial time on a non-deterministic single-tape Turing machine, i.e.,

$$NP = \bigcup_k NTIME(n^k)$$

Verifier

for language A is an algorithm V s.t. $w \in A$ iff V accepts $\langle w, c \rangle$ for some witness or proof string c. Examples: HAMPATH, Composities

Theorem

A language is in NP iff it has a poly-time verifier

Non-determinism vs Poly-time verifiers

Theorem

A language is in NP iff it has a poly-time verifier

Non-determinism vs Poly-time verifiers

Theorem

A language is in NP iff it has a poly-time verifier

- $\implies$

Non-determinism vs Poly-time verifiers

Theorem

A language is in NP iff it has a poly-time verifier

- $\implies$ Suppose N is the NTM that decides $L \in NP$, we define verifier V:
- On input $\langle w, c \rangle$, where w, c are strings,

Non-determinism vs Poly-time verifiers

Theorem

A language is in NP iff it has a poly-time verifier

- $\implies$ Suppose N is the NTM that decides $L \in NP$, we define verifier V:
- On input $\langle w, c \rangle$, where w, c are strings,
  1. Simulate N on w, with c as a description of the non-det choice.
  2. If this branch of N's computation accepts, accept, else reject.

Non-determinism vs Poly-time verifiers

Theorem

A language is in NP iff it has a poly-time verifier

- $\implies$ Suppose N is the NTM that decides $L \in NP$, we define verifier V:
- On input $\langle w, c \rangle$, where w, c are strings,
    1. Simulate N on w, with c as a description of the non-det choice.
    2. If this branch of N's computation accepts, accept, else reject.
- $\impliedby$ Given verifier V which runs in time n^k, we construct NTM N as follows:
- On input w of length n;

Non-determinism vs Poly-time verifiers

Theorem

A language is in NP iff it has a poly-time verifier

- $\implies$ Suppose N is the NTM that decides $L \in NP$, we define verifier V:
- On input $\langle w, c \rangle$, where w, c are strings,
  1. Simulate N on w, with c as a description of the non-det choice.
  2. If this branch of N's computation accepts, accept, else reject.
- $\impliedby$ Given verifier V which runs in time n^k, we construct NTM N as follows:
- On input w of length n;
  1. Guess (i.e., non-det choice) string c of length at most n^k
  2. Run V on $\langle w, c \rangle$
  3. Accept if V accepts, else reject

Examples of problems in NP class

Exercises

- CLIQUE: Does an undir graph G contain a clique of size k?
- SUBSET-SUM: Given a set of numbers, does some set add up to exactly S?

Examples of problems in NP class

Exercises

- CLIQUE: Does an undir graph G contain a clique of size k?
- SUBSET-SUM: Given a set of numbers, does some set add up to exactly S?

CLIQUE: $\{\langle G, k\rangle \mid G$ is an undirected graph with a k-clique$\}$. Give two proofs!

The start of many many questions

What about complementation?

The start of many many questions

What about complementation?

- Is $\overline{CLIQUES}$ in NP?

The start of many many questions

What about complementation?

- Is $\overline{CLIQUES}$ in NP?
- We define Co-NP for problems whose complement is in NP.

The start of many many questions

What about complementation?

- Is $\overline{CLIQUES}$ in NP?
- We define Co-NP for problems whose complement is in NP.
- Give an example of a problem in Co-NP.

The start of many many questions

What about complementation?

- Is $\overline{CLIQUES}$ in NP?
- We define Co-NP for problems whose complement is in NP.
- Give an example of a problem in Co-NP. COMPOSITES

The start of many many questions

What about complementation?

- Is $\overline{CLIQUES}$ in NP?
- We define Co-NP for problems whose complement is in NP.
- Give an example of a problem in Co-NP. COMPOSITES
- Is NP separated from Co-NP?

The start of many many questions

What about complementation?

- Is $\overline{CLIQUES}$ in NP?
- We define Co-NP for problems whose complement is in NP.
- Give an example of a problem in Co-NP. COMPOSITES
- Is NP separated from Co-NP?

What about *NP* vs *EXPTIME*?

The start of many many questions

What about complementation?

- Is $\overline{CLIQUES}$ in NP?
- We define Co-NP for problems whose complement is in NP.
- Give an example of a problem in Co-NP. COMPOSITES
- Is NP separated from Co-NP?

What about NP vs $EXPTIME$?

- $NP \subseteq EXPTIME$, but is it strict?

The start of many many questions

What about complementation?

- Is $\overline{CLIQUES}$ in NP?
- We define Co-NP for problems whose complement is in NP.
- Give an example of a problem in Co-NP. COMPOSITES
- Is NP separated from Co-NP?

What about NP vs $EXPTIME$?

- $NP \subseteq EXPTIME$, but is it strict?

One question to rule them all: is $P = NP$?

The start of many many questions

What about complementation?

- Is $\overline{CLIQUES}$ in NP?
- We define Co-NP for problems whose complement is in NP.
- Give an example of a problem in Co-NP. COMPOSITES
- Is NP separated from Co-NP?

What about NP vs $EXPTIME$?

- $NP \subseteq EXPTIME$, but is it strict?

One question to rule them all: is $P = NP$?

If $P = NP$, then what about the earlier questions?