Université Paris Cité
LIPADE

Algorithmic Complexity

Deterministic Time

Jean-Guy Mailly (jean-guy.mailly@u-paris.fr)

2022

Outline

Deterministic Time Complexity Classes

Polynomial vs Exponential Time

Decidability vs Complexity

- Decidability of a problem means that there is an algorithm which solves the problem in a finite time

Decidability vs Complexity

- Decidability of a problem means that there is an algorithm which solves the problem in a finite time
- What exactly means « finite time »? Is it always « reasonable »?

Decidability vs Complexity

- Decidability of a problem means that there is an algorithm which solves the problem in a finite time
- What exactly means « finite time »? Is it always « reasonable »?
  - Age of the univers $\simeq 4.7 \times 10^{17}$ seconds

Decidability vs Complexity

- Decidability of a problem means that there is an algorithm which solves the problem in a finite time
- What exactly means « finite time »? Is it always « reasonable »?
  - Age of the univers $\simeq 4.7 \times 10^{17}$ seconds
  - Frequency of modern CPUs $\simeq 4GHz = 4 \times 10^{9}$ operations/second

Decidability vs Complexity

- Decidability of a problem means that there is an algorithm which solves the problem in a finite time
- What exactly means « finite time »? Is it always « reasonable »?
  - Age of the univers $\simeq 4.7 \times 10^{17}$ seconds
  - Frequency of modern CPUs $\simeq 4GHz = 4 \times 10^9$ operations/second
  - If a problem is decidable but requires 10^n steps to be solved, is it « easy » to solve this problem?

Decidability vs Complexity

- Decidability of a problem means that there is an algorithm which solves the problem in a finite time
- What exactly means « finite time »? Is it always « reasonable »?
  - Age of the univers $\simeq 4.7 \times 10^{17}$ seconds
  - Frequency of modern CPUs $\simeq 4GHz = 4 \times 10^9$ operations/second
  - If a problem is decidable but requires 10^n steps to be solved, is it « easy » to solve this problem?
- Complexity theory aims at classifying decidable problems to determine which ones are « easy » to solve

Back to Turing Machines

Intuition

The size of the input must be taken into account when evaluating complexity: it takes more time to sort a list of 1000000 elements than a list of 3 elements.

Back to Turing Machines

Intuition

The size of the input must be taken into account when evaluating complexity: it takes more time to sort a list of 1000000 elements than a list of 3 elements.

Evaluating Time with Deterministic Turing Machines [Hartmanis and Stearns 1965]

Given a function $f : \mathbb{N} \mapsto \mathbb{N}$, DTIME($f(n)$) is the set of all languages decided by a DTM $\mathcal{M}$ in less than $g(n)$ steps, with $g(n) \in \mathcal{O}(f(n))$

Basic Results on DTIME

Proposition

- if $\forall n \in \mathbb{N}, f(n) \leq g(n)$, then $\text{DTIME}(f(n)) \subseteq \text{DTIME}(g(n))$
- $\forall f(n) \geq n$, $\text{DTIME}(f(n))$ is closed for finite union, finite intersection and complement
  - if $\mathcal{L}_1, \ldots, \mathcal{L}_m \in \text{DTIME}(f(n))$, then $\mathcal{L}_1 \cup \cdots \cup \mathcal{L}_m \in \text{DTIME}(f(n))$
  - if $\mathcal{L}_1, \ldots, \mathcal{L}_m \in \text{DTIME}(f(n))$, then $\mathcal{L}_1 \cap \cdots \cap \mathcal{L}_m \in \text{DTIME}(f(n))$
  - if $\mathcal{L} \in \text{DTIME}(f(n))$, then $\bar{\mathcal{L}} \in \text{DTIME}(f(n))$

Outline

Deterministic Time Complexity Classes

Polynomial vs Exponential Time

Polynomial vs Exponential Time

Definition

- The complexity class P is the set of languages decided in polynomial time, i.e

$$\mathrm{P} = \bigcup_{k \in \mathbb{N}} \mathrm{DTIME}(n^k)$$

- The complexity class EXP is the set of languages decided in exponential time, i.e

$$\mathrm{EXP} = \bigcup_{k \in \mathbb{N}} \mathrm{DTIME}(2^{n^k})$$

Polynomial vs Exponential Time

Definition

- The complexity class P is the set of languages decided in polynomial time, i.e

$$\mathrm{P} = \bigcup_{k \in \mathbb{N}} \mathrm{DTIME}(n^k)$$

- The complexity class EXP is the set of languages decided in exponential time, i.e

$$\mathrm{EXP} = \bigcup_{k \in \mathbb{N}} \mathrm{DTIME}(2^{n^k})$$

Theorem

$$\mathrm{P} \subset \mathrm{EXP}$$

Examples of Polynomial Problems

Integer Multiplication

- Input: $a, b \in \mathbb{N}$, in binary form, and $k \in \mathbb{N}$
- Problem: Is the k^{th} bit of $a \times b$ equal to 1?

Matrices Multiplication

- Input: A, B two matrices $m \times m$, a pair (i, j) with $1 \leq i, j \leq m$ and $k \in \mathbb{N}$
- Problem: Is the k^{th} bit of $AB_{(i,j)}$ equal 1?

Access

- Input: a digraph G and two vertices v_1, v_2
- Problem: Is there a path from v_1 to v_2 in G?

Examples of Exponential Problems

Bin Packing

- Input: $w_1, \ldots, w_n \in \mathbb{N}$ (weights of n objects), $m, w \in \mathbb{N}$ (number of boxes, maximal weight in boxes)
- Problem: Can we put all the objects in the boxes without exceeding the maximal weight?

k-Halting

- Input: a Turing Machine $\mathcal{M}$, $k \in \mathbb{N}$
- Problem: Does $\mathcal{M}(\epsilon)$ halt in less than k steps?

Easy vs Hard Problems?

P

When $\mathcal{P} \in$ P, $\mathcal{P}$ is considered as « easy ». We say that $\mathcal{P}$ is tractable.

Easy vs Hard Problems?

P

When $\mathcal{P} \in$ P, $\mathcal{P}$ is considered as « easy ». We say that $\mathcal{P}$ is tractable.

EXP

If $\mathcal{P} \notin$ P, $\mathcal{P}$ is considered as « hard ». The question is « how hard is it? » Can we separate problems from EXP in more fine-grained complexity classes?

Easy vs Hard Problems?

P

When $\mathcal{P} \in$ P, $\mathcal{P}$ is considered as « easy ». We say that $\mathcal{P}$ is tractable.

EXP

If $\mathcal{P} \notin$ P, $\mathcal{P}$ is considered as « hard ». The question is « how hard is it? » Can we separate problems from EXP in more fine-grained complexity classes?

- Not with DTM

Easy vs Hard Problems?

P

When $\mathcal{P} \in$ P, $\mathcal{P}$ is considered as « easy ». We say that $\mathcal{P}$ is tractable.

EXP

If $\mathcal{P} \notin$ P, $\mathcal{P}$ is considered as « hard ». The question is « how hard is it? » Can we separate problems from EXP in more fine-grained complexity classes?

- Not with DTM
- Can be done with NDTM: next chapter

Easy vs Easier Problems?

P

If $\mathcal{P} \in$ P, $\mathcal{P}$ is considered as « easy ». Can we distinguish between several levels of hardness in P problems?

Easy vs Easier Problems?

P

If $\mathcal{P} \in$ P, $\mathcal{P}$ is considered as « easy ». Can we distinguish between several levels of hardness in P problems?

- P-complete problems: the hardest problems in P
- Some classes are included in P (for instance L, NL, ...) representing problems which are easier than "general" P problems

[Turing 1936] A. M. Turing, *On computable numbers, with an application to the Entscheidungsproblem*. Proceedings of the London Mathematical Society, 1936.

[Hartmanis and Stearns 1965] J. Hartmanis and R. E. Stearns, *On the computational com- plexity of algorithms*. Transactions of the American Mathematical Society, 117, 285-306, 1965.