# Chapter 3 : Complexity classes

## ENSIIE - Computational complexity theory

Dimitri Watel (dimitri.watel@ensiie.fr)

2022

# Complexity of a computation of a Turing machine

**Definition**

- The time complexity of a computation of a Turing machine is the number of iterations performed by the machine before it stops.
- The space complexity of a computation of a Turing machine is the number of cells the machine write on before it ends.

Usually, the complexity depends on the size of $x$.

# Complexity of a computation of a Turing machine

**Definition**

Let $\mathcal{M}$ be a **deterministic** machine and $x$ be a word, we define $t(\mathcal{M}, x)$ as the time complexity of the computation of $\mathcal{M}$ when the input is $x$. Similarly, $s(\mathcal{M}, x)$ is its space complexity.

**Definition**

Let $\mathcal{M}$ be a **deterministic** machine, $x$ be a word and $C$ be a sequence of choices, we define $t(\mathcal{M}, x, C)$ as the time complexity of the computation of $\mathcal{M}$ when the input is $x$ and when it follows the choices of $C$. Similarly, $s(\mathcal{M}, x, C)$ is its space complexity.

**Definition**

Let $\mathcal{M}$ be a **deterministic** machine, the worst-case (time) complexity of $\mathcal{M}$ is a function $f$ such that

$$f(n) = \max_{x \in \{0,1,B\}^n} (t(\mathcal{M}, x))$$

**Definition**

Let $\mathcal{M}$ be a **deterministic** machine, the worst-case (time) complexity of $\mathcal{M}$ is a function $f$ such that

$$f(n) = \max_{x \in \{0,1,B\}^n} \min_{\text{Choices } C} (t(\mathcal{M}, x, C))$$

We similarly define the worst-case space complexity.

# Asymptotical worst-case time complexity

## Definition

Let $\mathcal{M}$ be a machine and let $f$ be its worst case complexity, we say $\mathcal{A}$

- has a worst-case complexity bounded by $g$ asymptotically if $f(n) = O(g(n))$ when $n \to +\infty$.
- has a worst-case complexity not dominated by $g$ asymptotically if $f(n) = \Omega(g(n))$ when $n \to +\infty$.
- has a worst-case complexity bounded both above and below by $g$ asymptotically if $f(n) = \Theta(g(n))$ when $n \to +\infty$.

We know say, through misuse of langage, that the complexity of $\mathcal{M}$ is $O(g(n))$ if the **wost case time complexity** of $\mathcal{M}$ is asymptotically bounded by $g$.

# Decision problem

## (Formal) definition

A decistion problem $\Pi$ is a set $\mathcal{L}$ of *instances* or *inputs* and a subset $\mathcal{L}_Y \subset \mathcal{L}$ of *positive instances*. We define $\mathcal{L} \backslash \mathcal{L}_Y = \mathcal{L}_N$ the *negative instances*.

## Solve a problem with a deterministic Turing machine

A **deterministic** Turing machine $\mathcal{M}$ solves $\Pi$ if

- when $x \in \mathcal{L}_Y$, $\mathcal{M}$ accepts $x$;
- when $x \in \mathcal{L}_N$ or $x \notin \mathcal{L}$, $\mathcal{M}$ rejects $x$.

# Complexity of a decision problem

**Definition: DTIME**

Let $\Pi$ be a decision problem, then $\Pi$ has a (worst-case) complexity $O(f(n))$ if there exists a **deterministic** Turing machine with complexity $O(f(n))$ solving $\Pi$. We write $\Pi \in \text{DTIME}(f(n))$.

# The complexity class P

## Definition of the class P

A decision problem $\Pi$ is polynomial, or belongs to P, if its complexity is polynomial. In other words, there is a **deterministic** Turing machine that solves $\Pi$ in polynomial time.

$$\mathsf{P} = \bigcup_{c \in \mathbb{N}} \mathsf{DTIME}(n^c)$$

# The complexity class EXPTIME

**Definition of the class EXPTIME**

A decision problem $\Pi$ is exponential, or belongs to EXPTIME, if its complexity is exponential. In other words, there is a **deterministic** Turing machine that solves $\Pi$ in exponential time:

$$\text{EXPTIME} = \bigcup_{c \in \mathbb{N}} \text{DTIME}(2^{n^c})$$

$$\text{P} \subsetneq \text{EXPTIME}$$

# Space complexity of a decision problem

**Definition: DSPACE**

Let $\Pi$ be a decision problem, then $\Pi$ has a (worst-case) space complexity $O(f(n))$ if there exists a **deterministic** Turing machine with space complexity $O(f(n))$ solving $\Pi$. We write $\Pi \in \text{DSPACE}(f(n))$.

# The complexity class PSPACE

**Definition of the class PSPACE**

A decision problem $\Pi$ belongs to PSPACE, if its space complexity is polynomial. In other words, there is a **deterministic** Turing machine that solves $\Pi$ in polynomial space.

$$\text{PSPACE} = \bigcup_{c \in \mathbb{N}} DSPACE(n^c)$$

$$\text{P} \subset \text{PSPACE} \subset \text{EXPTIME}$$

# The complexity class EXPSPACE

## Definition of the class EXPSPACE

A decision problem Π belongs to EXPSPACE, if its space complexity is exponential. In other words, there is a **deterministic** Turing machine that solves Π in exponential space:

$$\text{EXPSPACE} = \bigcup_{c \in \mathbb{N}} \text{DSPACE}(2^{n^c})$$

$$\text{PSPACE} \subsetneq \text{EXPSPACE}$$

$$\text{P} \subset \text{PSPACE} \subset \text{EXPTIME} \subset \text{EXPSPACE}$$

# Decision problem

## (Formal) definition

A decistion problem $\Pi$ is a set $\mathcal{L}$ of *instances* or *inputs* and a subset $\mathcal{L}_Y \subset \mathcal{L}$ of *positive instances*. We define $\mathcal{L} \backslash \mathcal{L}_Y = \mathcal{L}_N$ the *negative instances*.

## Solve a problem with a non-deterministic Turing machine

A **non-deterministic** Turing machine $\mathcal{M}$ solves $\Pi$ if

- when $x \in \mathcal{L}_Y$, $\mathcal{M}$ accepts $x$;
- when $x \in \mathcal{L}_N$ or $x \notin \mathcal{L}$, $\mathcal{M}$ strongly rejects $x$.

# Non-deterministic complexity of a decision problem

**Definition: NTIME**

Let $\Pi$ be a decision problem, then $\Pi \in \text{NTIME}(f(n))$ if there exists a **non-deterministic** Turing machine with complexity $O(f(n))$ solving $\Pi$.

# The complexity class NP

### Definition of the class NP (non-deterministic Polynomial)

A decision problem $\Pi$ belongs to NP, if there is a **non-deterministic** Turing machine that solves $\Pi$ in polynomial time.

$$\mathrm{NP} = \bigcup_{c \in \mathbb{N}} \mathrm{NTIME}(n^c)$$

$$\mathrm{P} \subset \mathrm{NP} \subset \mathrm{PSPACE}$$

# The complexity class NEXPTIME

## Definition of the class NEXPTIME

A decision problem Π belongs to NEXPTIME if there is a **non-deterministic** Turing machine that solves Π in exponential time:

$$\text{NEXPTIME} = \bigcup_{c \in \mathbb{N}} \text{NTIME}(2^{n^c})$$

$$\text{EXPTIME} \subset \text{NEXPTIME} \subset \text{EXPSPACE}$$

$$\text{NP} \subsetneq \text{NEXPTIME}$$

# Non-deterministic space complexity of a decision problem

**Definition: NSP**

Let $\Pi$ be a decision problem, then $\Pi \in \text{NSPACE}(f(n))$ if there exists a **non-deterministic** Turing machine with space complexity $O(f(n))$ solving $\Pi$.

# The complexity class NPSPACE

## Definition of the class NPSPACE

A decision problem Π belongs to PSPACE if there is a **non-deterministic** Turing machine that solves Π in polynomial space.

$$\text{NPSPACE} = \bigcup_{c \in \mathbb{N}} \text{NSPACE}(n^c)$$

!!!

$$\text{PSPACE} = \text{NPSPACE}$$

# The complexity class NEXPSPACE

## Definition of the class NEXPSPACE

A decision problem $\Pi$ belongs to NEXPSPACE if there is a **non-deterministic** Turing machine that solves $\Pi$ in exponential space:

$$\text{NEXPSPACE} = \bigcup_{c \in \mathbb{N}} \text{NSPACE}(2^{n^c})$$

!!!

$$\text{EXPSPACE} = \text{NEXPSPACE}$$

# The complexity class Co-NP

## Definition of the class Co-NP

A decision problem $\Pi = (\mathcal{L}, \mathcal{L}_Y, \mathcal{L}_N)$ belongs to Co-NP if $\Pi^c = (\mathcal{L}, \mathcal{L}_N, \mathcal{L}_Y)$ belongs to NP.

## Other definition of the class Co-NP

A decision problem $\Pi$ belongs to Co-NP if there exists a **non deterministic** machine $\mathcal{M}$ that strongly accepts the words of $\mathcal{L}_Y$ and rejects the others.

$$\text{P} \subset \text{NP} \cap \text{Co-NP}$$

$$\text{Co-NP} \subset \text{PSPACE}$$

## So many others

- Polynomial hierarchy : $\Sigma_2, \Pi_2, \Sigma_k, \Pi_k, PH$
- Probabilistic classes : $BPP, ZPP, RP$
- Quantum classes : $BQP, EQP$
- Non decidable classes : $RE, ALL$

https://complexityzoo.uwaterloo.ca/Complexity_Zoo