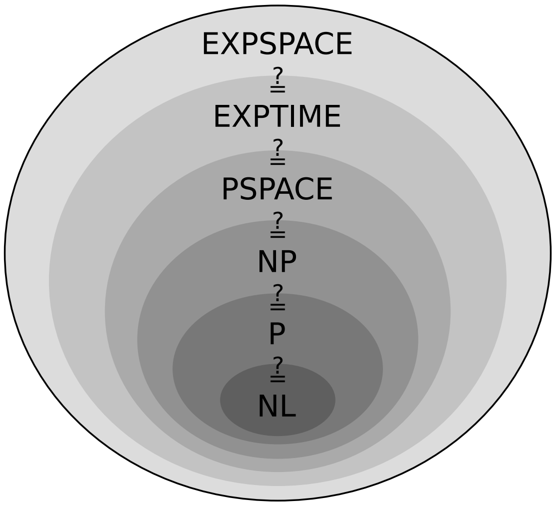
P (complexité)

La classe **P** est une classe très importante de la théorie de la complexité, un domaine de l'informatique théorique et des mathématiques. Par définition, un problème de décision est dans **P** s'il peut être décidé par une machine de Turing déterministe en temps polynomial par rapport à la taille de l'entrée. On qualifie alors le problème de polynomial (en temps).

Les problèmes dans **P** sont considérés comme “faisables” (*feasible* en anglais), faciles à résoudre (dans le sens où on peut le faire relativement rapidement). C'est une thèse émise par le scientifique américain Alan Cobham[1],[2]. La classe **P** est incluse dans la **classe NP**, conduisant à l'un des grands problèmes ouvert de la théorie de la complexité, à savoir : **P est-il égal à NP ?**



Représentations des inclusions des classes usuelles

1 Définitions

1.1 Définition classique

On note **TIME**(t(n)), la classe des problèmes de décision qui peuvent être résolus en temps de l'ordre de grandeur de t(n) sur une machine déterministe (où n est la taille de l'entrée). Alors par définition

$$\mathrm{P} = \bigcup_{c>1} \mathrm{TIME}(n^c).$$

1.2 Problèmes complets

Un problème de décision *A* est dit **P**-*complet*, ou *complet pour la classe* **P** s'il est dans la classe **P** et si tout problème de la classe **P** peut être réduit à *A* par une fonction qui demande un espace logarithmique, donc une réduction dans la classe **L**.

2 Relations avec les autres classes

On a l'inclusion évidente **P** ⊆ **NP,** mais l'une des grandes questions ouvertes de l'informatique théorique est le problème de savoir si P=NP ou pas.

On peut placer **P** dans la hiérarchie des langages, classés selon l'espace requis : elle contient **NL** (la classe des problèmes pouvant être résolus sur une machine non-déterministe en espace logarithmique) et est contenu par **PSPACE** (la classe des problèmes pouvant être résolus en espace polynomial). Les inclusions sont les suivantes (on ne sait pas si elles sont strictes) :

L ⊆ **NL** ⊆**NC** ⊆ **P** ⊆ **NP** ⊆ **PSPACE** ⊆ **EXPTIME** ⊆ **NEXPTIME** ⊆ **EXPSPACE**

P ≠ **EXPTIME**

Par ailleurs, **P** est le premier niveau de la hiérarchie polynomiale. Et **P** est incluse dans les classes polynomiales sur des machines plus puissantes (quantiques ou utilisant du hasard par exemple), comme **ZPP**, **BPP** et **BQP**.

3 Problèmes de P

Les problèmes dans **P** peuvent être des problèmes très simples (notamment ceux qui demandent un temps constant de calculs), mais aussi des problèmes plus complexes comme des restrictions du problème SAT et même des problèmes a priori très difficiles et dont la preuve de l'appartenance à **P**, qui se fait généralement par la découverte d'un algorithme polynomial, a demandé des années de recherche : c'est le cas notamment du test de primalité. Voici quelques exemples de problèmes polynomiaux.

3.1 Problèmes remarquables

- Le test de primalité AKS est un algorithme qui montre que le problème de savoir si un entier est premier ou non peut être résolu par un algorithme polynomial.

- Programmation linéaire : Les algorithmes de points intérieurs permettent de classer la programmation linéaire dans la classe **P**. Ce problème est même **P**-complet.

3.2 Problèmes P-complet

Les problèmes suivant sont P-complets.

- Horn-SAT : Le problème SAT *général* est NP-complet, mais en le restreignant à des clauses de Horn il entre dans **P**.
- Problème de la valeur d'un circuit (*circuit value*) : ce problème, en anglais *circuit value problem* ou CVP (en), consiste à déterminer si, pour un circuit booléen donné, le résultat de la fonction réalisée sur une entrée donnée correspond à une valeur donnée.
- Le problème qui consiste à savoir si une grammaire algébrique est vide ou pas[3].

3.3 Autres problèmes

- Connexité dans un graphe : Un exemple de problème polynomial est celui de la connexité dans un graphe. Étant donné un graphe à n sommets (on considère que la taille de la donnée, donc du graphe, est son nombre de sommets), il s'agit de savoir si toutes les paires de sommets sont reliées par un chemin. Un algorithme de parcours en profondeur construit un arbre couvrant du graphe à partir d'un sommet. Si cet arbre contient tous les sommets du graphe, alors le graphe est connexe. Le temps nécessaire pour construire cet arbre est au plus $c.n^2$ (où c est une constante et n le nombre de sommets du graphe), donc le problème est dans la classe **P**.

4 Propriétés supplémentaires

On parle dans certains cas, d'algorithmes en temps *fortement ou faiblement polynomial*. Cette distinction existe pour les problèmes dont l'entrée contient des entiers. On parle de temps faiblement polynomial si les entiers doivent être donnés en écriture unaire (c'est-à-dire que le nombre n compte pour une taille n) pour avoir un temps polynomial, et on parle de temps fortement polynomial si même l'écriture compacte (n a taille $\log(n)$) donne une complexité polynomiale.

5 Bibliographie

(en) Sanjeev Arora et Boaz Barak, Computational Complexity : A modern Approach, Cambridge University Press, 2009 (ISBN 0-521-42426-7), chap. 1 (« The computational model —and why it doesn't matter »)

6 Lien externe

(en) La classe P sur le *Complexity Zoo*

7 Notes et références

[1] (en) Sanjeev Arora et Boaz Barak, Computational Complexity : A modern Approach, Cambridge University Press, 2009 (ISBN 0-521-42426-7), chap. 1 dans les notes historiques

[2] Alan Cobham, « The intrinsic computational difficulty of functions », *Proceedings of the 1964 International Congress for Logic, Methodology, and Philosophy of Science*, North Holland, 1965

[3] Neil D. Jones et William T. Laaser, « Complete Problems for Deterministic Polynomial Time », *Theor. Comput. Sci.*, vol. 3, nº 1, 1976, p. 105-117

- Portail de l'informatique théorique

8 Sources, contributeurs et licences du texte et de l'image

8.1 Texte

- **P (complexité)** *Source :* https://fr.wikipedia.org/wiki/P_(complexit%C3%A9)?oldid=118765303 *Contributeurs :* PIerre.Lescanne, Ange Gabriel, Syrak, ManiacParisien, Roll-Morton, Addbot, YassineMR et Anonyme : 5

8.2 Images

- **Fichier:Complexity_subsets_pspace.svg** *Source :* https://upload.wikimedia.org/wikipedia/commons/6/6e/Complexity_subsets_pspace.svg *Licence :* Public domain *Contributeurs :* Own work by uploader, intended to replace bitmap image illustrating same thing *Artiste d'origine :* Hand drawn in Inkscape Qef
- **Fichier:Max-cut.svg** *Source :* https://upload.wikimedia.org/wikipedia/commons/c/cf/Max-cut.svg *Licence :* CC BY-SA 3.0 *Contributeurs :* Travail personnel *Artiste d'origine :* Miym

8.3 Licence du contenu