Décidabilité et classes de complexité

Quentin Fortier

November 28, 2024

Décidabilité

Définition

Un algorithme est un programme fini (suite d'instructions) qui prend une entrée finie.
On suppose que cet algorithme s'exécute sur un ordinateur avec une quantité illimité de mémoire.
Sur une entrée, un algorithme peut :

- renvoyer un résultat en temps fini
- ne pas renvoyer de résultat, soit parce qu'il ne termine pas (boucle infinie...), soit parce qu'il plante (dépassement de tableau...)

Décidabilité

Définition

Un problème de décision est un couple (I, P) tel que :

- I est l'ensemble des instances du problème
- P est l'ensemble des instances positives du problème

On peut aussi définir un problème sous forme d'une question binaire sur une instance.

Décidabilité

Exemples de problèmes de décision :

CONNEXE

- **Instance** : un graphe G
- **Question** : G est-il connexe ?

SAT

- **Instance** : une formule logique φ en forme normale conjonctive
- **Question** : φ est-elle satisfiable ?

APPARTIENT

- **Instance** : un mot w et un automate A
- **Question** : $w \in L(A)$?

Décidabilité

Définition

Un problème de décision (I, P) est dit décidable s'il existe un algorithme qui :

- prend une instance $i \in I$ du problème en entrée
- renvoie `true` si i est une instance positive ($i \in P$)
- renvoie `false` si i est une instance négative ($i \notin P$)

Sinon, le problème est dit indécidable.

Décidabilité

Remarques :

- Pour montrer qu'un problème est décidable, il suffit d'exhiber un algorithme qui le décide. Montrer qu'il est indécidable est a priori plus difficile : il faut montrer qu'aucun algorithme ne peut le résoudre.
- Quand on s'intéresse à la décidabilité d'un problème, seule l'existence d'un algorithme compte : sa complexité n'a aucune importance. Il n'est donc pas non plus nécessaire de préciser les structures de données utilisées, tant que celles-ci sont calculables.
- Si l'ensemble des instances positives est de cardinal fini, le problème est trivialement décidable : il suffit d'énumérer toutes les instances positives et tester si l'une d'entre elles est égale à l'entrée.

Décidabilité

Exercice

Montrer que le problème SAT est décidable.

Décidabilité

Définition

Une fonction $f : E \to F$ est calculable s'il existe un algorithme A qui, pour tout élément $x \in E$, termine en temps fini et renvoie $f(x)$.

Décidabilité

ARRET

- **Instance** : le code source d'un programme f et un argument x
- **Question** : f termine-t-il sur l'entrée x ?

Décidabilité

ARRET

- **Instance** : le code source d'un programme f et un argument x
- **Question** : f termine-t-il sur l'entrée x ?

En OCaml, cela revient à écrire une fonction `termine : string -> string -> bool` telle que si `f` contient le code source d'une fonction f et une chaîne de caractères `x` correspondant à un argument x, `termine f x` renvoie `true` si $f(x)$ termine, `false` sinon.

Décidabilité

ARRET

- **Instance** : le code source d'un programme f et un argument x
- **Question** : f termine-t-il sur l'entrée x ?

En OCaml, cela revient à écrire une fonction `termine : string -> string -> bool` telle que si `f` contient le code source d'une fonction f et une chaîne de caractères `x` correspondant à un argument x, `termine f x` renvoie `true` si $f(x)$ termine, `false` sinon.

Théorème

Le problème de l'arrêt est indécidable.

Décidabilité

Définition

On dit qu'un problème de décision $A = (I_A, P_A)$ se réduit à un problème de décision $B = (I_B, P_B)$, noté $A \leq B$, s'il existe une fonction calculable $f : I_A \longrightarrow I_B$ telle que : $\forall i \in I_A, i \in P_A \Leftrightarrow f(i) \in P_B$.

Autrement dit : A se réduit à B si un algorithme pour résoudre B permet de résoudre A.

Exercice

Montrer que CONNEXE $\leq$ ACCESSIBLE.

ACCESSIBLE

- **Instance** : un graphe $G = (S, A)$ et deux sommets $s, t \in S$
- **Question** : existe-t-il un chemin de s à t dans G ?

Décidabilité

Théorème

Soient A et B deux problèmes de décision tels que $A \leq B$. Alors :

- Si B est décidable, alors A est décidable.
- Si A est indécidable, alors B est indécidable.

Décidabilité

Exercice

Montrer que le problème ARRET-VIDE est indécidable.

ARRET-VIDE

- **Instance** : le code source d'un programme f
- **Question** : f termine-t-il sur l'entrée vide ?

Classes de complexité

Définition

La taille $|x|$ d'une instance x d'un problème est le nombre de bits nécessaires pour la coder.

Classes de complexité

Définition

La taille $|x|$ d'une instance x d'un problème est le nombre de bits nécessaires pour la coder.

Remarques :

- Un entier n est codé en base 2, donc sa taille est $\log_2(n)$. On pourrait aussi le coder en unaire ce qui donnerait une taille n, mais ce n'est pas « raisonnable ».
- On s'intéresse seulement à l'ordre de grandeur de la taille (O(...)).

Classes de complexité

Définition

La taille $|x|$ d'une instance x d'un problème est le nombre de bits nécessaires pour la coder.

Remarques :

- Un entier n est codé en base 2, donc sa taille est $\log_2(n)$. On pourrait aussi le coder en unaire ce qui donnerait une taille n, mais ce n'est pas « raisonnable ».
- On s'intéresse seulement à l'ordre de grandeur de la taille (O(...)).

Exemples :

- La taille d'un entier n est $\log_2(n)$.
- Un ensemble de p entiers dans $[\![1, n]\!]$ a une taille de $p \log_2(n)$.
- Un graphe à n sommets représenté par une matrice d'adjacence a une taille de n^2.

Classes de complexité : P

Définition

La classe P est l'ensemble des problèmes de décision qui admettent un algorithme de complexité polynomiale en la taille de l'entrée (c'est-à-dire $O(n^k)$ pour une constante k, où n est la taille de l'entrée).

Classes de complexité : P

Définition

La classe P est l'ensemble des problèmes de décision qui admettent un algorithme de complexité polynomiale en la taille de l'entrée (c'est-à-dire $O(n^k)$ pour une constante k, où n est la taille de l'entrée).

Définition (HP)

La classe EXP est l'ensemble des problèmes de décision qui admettent un algorithme de complexité exponentielle en la taille de l'instance (c'est-à-dire $O(2^{n^k})$ pour une constante k, où n est la taille de l'entrée).

Remarque : P $\subset$ EXP.

Classes de complexité : P

Exemple :

PREMIER

- **Instance** : un entier n
- **Question** : n est-il premier ?

On peut énumérer les entiers de 2 à $\sqrt{n}$ pour tester si n est divisible par l'un d'entre eux, en complexité $O(\sqrt{n})$. Ceci est polynomial en n mais exponentielle en la taille $\log_2(n)$ de n (car $\sqrt{n} = 2^{\frac{\log_2(n)}{2}}$), donc cela montre PREMIER $\in EXP$.

Classes de complexité : P

Soit $G = (S, A)$ un graphe et $G' = (S', A')$ un sous-graphe de G (c'est-à-dire $S' \subset S$ et $A' \subset A$).
On dit que G' est une clique de G si G' est complet, c'est-à-dire que pour tout couple de sommets $u, v \in G'$, il existe une arête entre u et v.

Exercice

1. Soit k un entier fixé. Montrer k-CLIQUE $\in$ P.
2. Montrer que CLIQUE $\in$ EXP.

k-CLIQUE

- **Instance** : un graphe G
- **Question** : G contient-il une clique de taille k ?

CLIQUE

- **Instance** : un graphe G et un entier k
- **Question** : G contient-il une clique de taille k ?

Classes de complexité : NP

Contrairement à la classe P qui est l'ensemble des problèmes que l'on peut résoudre en complexité polynomiale, la classe NP est l'ensemble des problèmes pour lesquels on peut vérifier une solution en complexité polynomiale.

Définition

Un problème de décision A appartient à la classe NP (*Nondeterministic Polynomial*) si, pour toute instance positive x de A, il existe un certificat c de taille polynomiale en $|x|$ qui permette de vérifier en temps polynomial en $|x| + |c|$ est une instance positive de A.

Un certificat peut-être un entier, un ensemble de valeurs...

Attention : NP ne veut pas dire « non polynomial ».

Classes de complexité : NP

Exercice

Montrer que les problèmes suivants appartiennent à NP :

CLIQUE

- **Instance** : un graphe G et un entier k
- **Question** : G contient-il une clique de taille k ?

SAT

- **Instance** : une formule logique φ en forme normale conjonctive
- **Question** : φ est-elle satisfiable ?

FACTORISATION

- **Instance** : une formule logique φ en forme normale conjonctive
- **Question** : φ est-elle satisfiable ?

Classes de complexité : NP

Théorème

$P \subset NP$.

Remarque : La question « $P = NP$? » est un des problèmes ouverts les plus célèbres en informatique.