

Partiel de calculabilité et complexité

M1 informatique – 2h

Le 23 novembre 2009

La qualité de la rédaction et de la présentation sera prise en compte dans l'évaluation. Le barème est donné à titre indicatif.

Exercice 1 – Vrai ou faux ?

(6 points)

Les affirmations suivantes sont-elles vraies ou fausses ? Ne justifiez pas vos réponses. On comptera +0,5 point par bonne réponse, −0,5 point par mauvaise réponse, et 0 pour une absence de réponse. Si la note totale est négative, on donnera 0 à l'exercice.

1. Si le problème A se réduit au problème B et si A est décidable, alors B est décidable.
2. Si $A \subseteq \mathbb{N}$ est récursivement énumérable alors $B = \{n \bmod 13 : n \in A\}$ est décidable. (*Rappel : on note $n \bmod 13$ l'entier compris entre 0 et 12 qui est le reste de la division euclidienne de n par 13.*)
3. Tous les langages sur un alphabet à une lettre sont décidables.
4. Toute fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ telle que $f(n) = 1$ si $n \geq 2009$ est calculable.
5. Il existe des problèmes décidables qui se réduisent au problème de l'arrêt.
6. Tout langage décidable se réduit au langage $\{0, 1\}$.
7. Il existe un langage reconnu par machine de Turing non déterministe mais pas par machine de Turing déterministe.
8. Si A et B sont des langages indécidables, alors $A \cup B$ est indécidable.
9. La classe **NP** contient tous les problèmes qu'on ne peut pas résoudre en temps polynomial.
10. Le langage des mots sur l'alphabet $\{a, b\}$ dont la longueur est un cube est dans **P**.
11. Étant donné un entier n écrit en binaire, on peut calculer la représentation binaire de l'entier 2^n en temps polynomial.
12. Le théorème d'accélération linéaire permet de simuler une machine de Turing déterministe fonctionnant en temps 10^n par une machine de Turing déterministe fonctionnant en temps 2^n .

Exercice 2 (3 points)

Donner explicitement (c'est-à-dire donner ses transitions) une machine de Turing déterministe à un ruban pour décider le langage $\{a^n b^n c^n : n \in \mathbb{N}\}$. Vous expliquerez d'abord intuitivement ce que fait la machine et vous calculerez sa complexité en temps.

Exercice 3 (3 points)

Soit L le langage $\{k : k \text{ n'est pas un nombre premier}\}$, où k est un entier donné en binaire. Montrer que $L \in \text{NP}$.

Exercice 4 (4 points)

Soit A le problème suivant :

Entrée – le code d'une machine de Turing déterministe M .

Problème – est-ce que M contient un état qui n'est jamais atteint (à partir d'aucune entrée) ?

1. Justifier que le théorème de Rice ne permet pas de conclure quant à l'indécidabilité de A .
2. Montrer que A est indécidable (vous pourrez donner une réduction du problème de l'arrêt).

Exercice 5 (4 points)

Soit M_1, M_2, \dots une énumération calculable des machines de Turing déterministes. On définit $m_i \in \{0, 1\}$ par

$$m_i = \begin{cases} 1 & \text{si } M_i \text{ s'arrête à partir du ruban vide} \\ 0 & \text{sinon.} \end{cases}$$

Enfin, $\Omega \in [0, 1]$ désigne le réel dont la représentation binaire est $0, m_1 m_2 m_3 \dots$, c'est-à-dire

$$\Omega = \sum_{i \geq 1} m_i 2^{-i}.$$

1. Soit $b \in]0, 1[$ un nombre réel compris entre 0 et 1. On note b_i le i -ème chiffre de b dans sa représentation binaire, c'est-à-dire

$$b = \sum_{i \geq 1} b_i 2^{-i}.$$

Soit $L_b = \{i : b_i = 1\}$. Montrer que si b est rationnel ($b \in \mathbb{Q}$), alors L_b est décidable.

2. Montrer que Ω n'est pas rationnel.