Année 2007–2008 — Session 2

Parcours :	MCSB71–74	**Date :**	17 juin 2008, 8h30
Code UE :	INF461	**Durée :**	3h00
Épreuve de :	Modèles de calcul - M. Zeitoun		

Notes et documents de cours et TD autorisés. Autres documents interdits.
La notation tiendra compte de la clarté de la rédaction et du soin apporté dans les justifications.

Exercice 1 Répondre par vrai ou faux aux questions suivantes, sans donner une justification. Barème : $+0,5$ pour une réponse juste, $-0,5$ pour une réponse fausse, 0 pour une absence de réponse.

1. Si un langage L est récursivement énumérable, l'ensemble des mots de L de longueur paire est aussi récursivement énumérable.
2. Tout langage dénombrable est décidable.
3. Tout langage dont le complémentaire est algébrique est décidable.
4. On peut décider si, étant donnée une machine de Turing sur un alphabet Σ, elle accepte tous les mots de Σ^*.
5. Étant donné un entier n en binaire, on peut calculer la représentation binaire de l'entier 2^n en temps polynomial.
6. Tout langage sur un alphabet à une lettre est décidable.
7. Toute application de $\mathbb{N}$ dans $\mathbb{N}$ nulle sauf sur un nombre fini d'entiers est primitive récursive.
8. Il existe une fonction primitive récursive bijective de $\mathbb{N}^3$ dans $\mathbb{N}$.

Exercice 2 Dans cet exercice les entiers sont codés en binaire. On pourra utiliser les machines usuelles de manipulation d'entiers en binaire (comparaisons, opérateurs arithmétiques,...) sans en refaire les constructions. On note $L \subseteq \{0,1\}^*$ l'ensemble des représentations binaires des entiers qui ne sont pas une puissance de 3.

1. Décrire de façon informelle, mais précisément, une machine de Turing <u>déterministe</u> qui décide le langage L en temps polynomial. Justifier précisément la complexité.
2. On considère la machine suivante pour résoudre le même problème : elle génère de façon non-déterministe un entier k en binaire, différent de 1 et de taille inférieure ou égale à celle de l'entrée n. Si k est divisible par 3, elle répond KO. Sinon, elle teste si k divise n : si oui elle répond OK, sinon elle répond KO.
   - Cette machine est-elle correcte pour tester que n n'est pas une puissance de 3 ? Justifier.
   - Sa complexité est-elle polynomiale ?

Exercice 3 Soient $f : \mathbb{N} \to \mathbb{N}$ et $g : \mathbb{N}^2 \to \{0,1\}$ primitives récursives telles que pour tout $n \in \mathbb{N}$, il existe $k \leqslant f(n)$ tel que $g(k,n) = 1$. Montrer que la fonction $h : \mathbb{N} \to \mathbb{N}$ définie par

$$h(n) = \min\{k \in \mathbb{N} \mid k \leqslant f(n) \text{ et } g(k,n) = 1\}$$

est primitive récursive. Est-ce toujours vrai si on remplace min par max ? Justifier.

Exercice 4 On considère le problème TRANSITION-UTILE suivant :

Donnée Une machine de Turing déterministe M et une transition (p, a, b, q, d) de M.

Question La transition (p, a, b, q, d) est-elle utilisée au cours d'un calcul de M sur au moins une entrée ?

On veut montrer que ce problème est indécidable.

1. Expliquer pourquoi le théorème de Rice ne s'applique pas directement pour montrer que ce problème est indécidable.
2. Donner un algorithme qui, à partir d'une machine de Turing M_1, construit une machine de Turing M et une instruction (p, a, b, q, d) de M avec la propriété suivante :

$$\varepsilon \in \mathscr{L}(M_1) \Longleftrightarrow M \text{ utilise la transition } (p, a, b, q, d) \text{ sur au moins une entrée.}$$

   Indication : construire M de telle sorte qu'elle utilise la transition soit sur toutes les entrées, soit sur aucune.
3. Les résultats de la question précédente indiquent-ils que TRANSITION-UTILE se réduit au problème du mot vide, ou l'inverse ? Justifier très précisément.
4. Conclure que le problème TRANSITION-UTILE est indécidable, en justifiant.
5. Un compilateur C peut-il détecter, pour chacune des instructions du programme qu'il prend en entrée, si cette instruction sera ou non exécutée ?

Exercice 5 On considère les λ-termes $C = \lambda x.\lambda y.\lambda f.fxy$, $T = \lambda g.g(\lambda x.\lambda y.x)$.

1. Réduire le λ-terme $T(Cab)$, où a et b sont des termes réduits. Chaque pas de réduction devra être correctement justifié, avec en particulier le rédex mis en évidence.
2. Si Cab représente le couple (a, b), comment interpréter le λ-terme T ?
3. Définir un λ-terme permettant de calculer la seconde composante d'un couple.