Calculabilité et complexité

Contrôle continu

Les exercices sont indépendants. Le barème est donné à titre indicatif.

Exercice 1 : une variante des machines de Turing (*10 points*)

Définition. On considère un alphabet Σ contenant les symboles $\triangleright$ et $\sqcup$, mais ni le symbole $\rightarrow$ ni le symbole $\leftarrow$. On définit une *T-machine* par le quintuplet $(\Sigma, K, \delta, s, H)$ où K, s et H sont respectivement un ensemble d'états, un état de départ et un ensemble d'états d'arrêt comme pour une machine de Turing standard. $\delta : K \times \Sigma \rightarrow K \times (\Sigma \setminus \{\triangleright\}) \times \{\rightarrow, \leftarrow\}$ est une fonction de transition qui à un couple (q, α) associe un triplet (p, β, f) où p est un nouvel état, β une lettre de $\Sigma \setminus \{\triangleright\}$ et f une flèche indiquant un déplacement soit à droite ($\rightarrow$) soit à gauche ($\leftarrow$) avec les signification habituelle des machines de Turing. Autrement dit, une T-machine est une variante des machines de Turing telles que nous les avons vues en cours pour lesquelles, à chaque transition, on écrit une lettre *et* on déplace la tête de lecture/écriture.

1- Donnez pour une T-machine les définitions correspondant à configuration, pas de calcul et calcul. Par analogie avec les machines de Turing, on appelle langage T-décidable un langage reconnu par une T-machine : donnez une définition précise de cette notion. Même chose pour la notion de fonction T-calculable.

2- Étant donnée une T-machine M, montrez de manière constructive qu'il existe une machine de Turing M' équivalente à M dans un sens que vous préciserez. Vous montrerez précisément comment on construit la machine M' à partir de la machine M. Déduisez-en que les langages T-décidables sont récursifs.

3-Montrez que tout langage récursif est T-décidable.

On appelle RT-machine une T-machine où le seul déplacement possible de la tête de lecture/écriture est vers la droite, c'est-à-dire que $\delta : K \times \Sigma \rightarrow K \times (\Sigma \setminus \{\triangleright\}) \times \{\rightarrow\}$.

4- On dit qu'un langage est RT-décidable si et seulement si il existe une RT-machine qui le décide. En essayant de la ranger dans la hiérarchie des langages, que peut-on dire de la classe des langages RT-décidables ? Expliquez précisément votre réponse.

5- On dit que $f : \Sigma^* \rightarrow \Sigma^*$ est une fonction RT-calculable si il existe une RT-machine qui la calcule. Montrez que si L est un langage récursif et f est une fonction RT-calculable alors le langage $f(L)$ est récursif.

Exercice 2 : indécidabilité (*10 points*)

1- Montrez que le problème suivant est indécidable : “Déterminer, étant donnés en entrée une machine de Turing M, si le langage semi-décidé par M est fini.”

2- On dit qu'une machine de Turing M utilise k cases du ruban d'entrée avec le le mot w en entrée si il existe une configuration $(q,\, u\underline{a}v)$de M telle que $(s,\, \triangleright\underline{\sqcup}w) \vdash_M^* (q,\, u\underline{a}v)$ et $|uav| \geq k$.

a) Montrer que le problème suivant est décidable : Étant donnés une machine de Turing M, une entrée w et un nombre k, est-ce que M utilise k cases du ruban d'entrée avec l'entrée w ?
b) Montrer que le problème suivant est indécidable : Étant données une machine de Turing M et un mot w, est-ce qu'il existe un entier naturel k tel que M n'utilise pas k cases du ruban avec l'entrée w ?

3- Un homomorphisme (de monoïde) de Σ^* dans Σ^* est une application telle que pour tous mots u et v de Σ^* on ait $f(uv) = f(u)f(v)$. Montrez que tout homorphisme est défini par les valeurs qu'il prend sur les lettres de Σ et donnez un exemple (non trivial) d'homomorphisme.
Montrez qu'un homomorphisme est une fonction récursive et implantez votre exemple avec une machine de Turing.
Soit f un homomorphisme de Σ^* dans Σ^*. Montrez que si L est un langage récursif, il en est de même de $f(L)$.