
TD 03 – Le temps de calcul est une fonction de la taille des entrées

Exercice 1.*Machines de Turing déterministes*

Donner une machine de Turing déterministe (peut importe le nombre de rubans) et une fonction (de la taille de l'entrée) bornant son temps d'exécution, pour les problèmes de décision suivants.

Palindrome

1. *entrée* : un mot $x \in \{a, b, k, y\}^*$
question : x est-il un palindrome ?

Un palindrome est un mot qui se lit identiquement de la gauche vers la droite et de la droite vers la gauche, tel que le mot *kayak*.

Nous souhaitons maintenant avoir une machine de Turing équivalente à votre réponse à la question 1, en réduisant l'alphabet à $\{0, 1\}$.

2. Donner une machine qui simule la machine de votre réponse en 1, fonctionnant sur l'alphabet $\{0, 1, B\}$.

Exercice 2.*SAT et Verif-SAT*

1. Donner un algorithme pour le problème suivant.

SAT

entrée : une formule propositionnelle ϕ
question : ϕ est-elle satisfaisable ?

2. Borner le temps d'exécution de l'algorithme proposé en 1.
3. Donner un algorithme pour le problème suivant.

Verif-SAT

entrée : une formule propositionnelle ϕ et une valuation $v : X \rightarrow \{\top, \perp\}$ avec X l'ensemble des variables de ϕ
question : v satisfait-elle ϕ ?

4. Borner le temps d'exécution de l'algorithme proposé en 3.
5. Donner un algorithme non-déterministe pour le problème **SAT**.
6. Borner le temps d'exécution de l'algorithme proposé en 5.

Exercice 3.*Codages « raisonnables »*

1. Donner un algorithme pour transformer un graphe orienté donné par sa matrice d'adjacence

$(m_{i,j})_{i \in V, j \in V}$ telle que $m_{i,j} = 1$ ssi il existe un arc de i vers j ,

en listes d'adjacence

$(\ell_i)_{i \in V}$ avec $\ell_i \subseteq V$ telles que $j \in \ell_i$ ssi il existe un arc de i vers j .

2. Donner un algorithme pour faire le contraire.
3. Borner le temps d'exécution de chacun de ces deux algorithmes.

Exercice 4.

Opérations sur les langages

1. Montrer que la classe des langages reconnus par les machines de Turing est close par union, intersection et complémentaire. C'est-à-dire que si L_1 et L_2 sont des langages reconnus par des machines de Turing, alors $L_1 \cup L_2$, $L_1 \cap L_2$ et $\Sigma^* \setminus L_1$ (et $\Sigma^* \setminus L_2$) le sont aussi, avec Σ l'alphabet sur lequel le problème est défini.
2. Fournir une borne supérieure aux temps de calcul des langages $L_1 \cup L_2$, $L_1 \cap L_2$ et $\Sigma^* \setminus L_1$ en fonction des temps de calcul de L_1 et L_2 .