
TD 02 – Machines de Turing

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.

1. **Palindrome**; entrée : un mot $x \in \{a, b, k, y\}^*$; question : x est-il un palindrome¹?
2. **Parité**; entrée : un entier $x \in \{0, 1\}^*$ codé en binaire; question : x contient-il un nombre pair de lettre 1?

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

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

Exercice 2.*Langage de haut niveau*

Soit l'algorithme suivant :

```

entrée : une suite de bits x[1], x[2], ..., x[n]
i <- n
tant que ( x[i] == 1 et i >= 0 )
faire
    i <- i-1
fin tant que
si i == -1
alors
    accepter
sinon
    rejeter
fin si

```

1. Donner une machine de Turing décidant le même problème.
2. Quel est le problème décidé par l'algorithme de la question 1 (donner le langage associé)?

Exercice 3.*Taille des entrées*

1. Donner un algorithme en langage de haut niveau pour décider le problème suivant.
Primalité; entrée : un entier $x \in \{0, 1\}^*$ codé en binaire; question : x est-il premier?
2. Borner le temps d'exécution de l'algorithme proposé en 1.
3. Donner un algorithme en langage de haut niveau pour décider le problème suivant.
Primalité unaire; entrée : un entier $x \in \{1\}^*$ codé en unaire; question : x est-il premier?

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

4. Borner le temps d'exécution de l'algorithme proposé en 3.

Exercice 4.

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 5.

Codages « raisonnables »

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

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

en listes d'adjacence

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

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