

## TP 01 – Machines de Turing

---

Pour ce TP nous allons utiliser le simulateur de machine de Turing `awmorp/jsturing!`

Deux options :


- sur internet : `http://morphett.info/turing/`
- en local : `git clone https://github.com/awmorp/jsturing.git`  
`firefox jsturing/turing.html`

N'oubliez pas de sauvegarder le code de vos machines...

**Attention** : par rapport aux définitions et notations vues en cours, les machines de Turing de ce TP n'auront qu'un seul ruban, en lecture écriture, et qui sert à la fois de ruban d'entrée, de ruban de travail, et de ruban de sortie! C'est équivalent du point de vue des classes de complexité P et NP, pour peu que l'on augmente l'alphabet de ruban.

### Exercice 1.

*nombre de rubans*

 S'assurer de bien comprendre la remarque ci-dessus relative au nombre de rubans.

### Exercice 2.


*Programmer*

Pour chacun des problèmes suivants, il faut :

- (a) donner le langage associé,
  - (b) programmer une machine de Turing pour le décider (on utilisera deux états `halt-accepte` et `halt-rejette`),
  - (c) donner une borne supérieure et une borne inférieure (les plus proches possibles) à leur temps de calcul dans le pire cas en fonction de la taille de l'entrée.
1. **Parité**; entrée : un entier  $x \in \{0, 1\}^*$  codé en binaire; question :  $x$  contient-il un nombre pair de lettre 1?
  2. **Mod4**; entrée : un entier  $x \in \{0, 1\}^*$  codé en binaire; question :  $x$  est-il un multiple de 4?
  3. **Préfixe**; entrée : deux mots  $x, y \in \{a, b\}^*$  séparés par un symbole `#`; question :  $x$  est-il un préfixe<sup>1</sup> de  $y$ ?
  4. **Sous-mot**; entrée : deux mots  $x, y \in \{a, b\}^*$  séparés par un symbole `#`; question :  $x$  est-il un facteur<sup>2</sup> de  $y$ ?

### Exercice 3.

↑

 Programmer une machine de Turing qui ne s'arrête pas.

### Exercice 4.

*Machines non-déterministes*

Pour chacun des problèmes suivants, il faut :

- (a) donner le langage associé,
- (b) programmer une machine de Turing non-déterministe pour le décider,

---

1.  $x = x_1, \dots, x_n$  est un préfixe de  $y = y_1, \dots, y_m$  si et seulement si  $n \leq m$  et  $\forall j, 1 \leq j \leq n : x_j = y_j$ .  
 2.  $x = x_1, \dots, x_n$  est un facteur du mot  $y = y_1, \dots, y_m$  si et seulement si il existe un indice  $i, 1 \leq i \leq m - n + 1$ , tel que  $\forall j, 0 \leq j \leq n - 1 : x_{1+j} = y_{i+j}$ .

- (c) donner une borne supérieure et une borne inférieure à leur temps de calcul dans le pire cas en fonction de la taille de l'entrée,
  - (d) comparer au temps de calcul d'une machine déterministes décidant le même langage.
1. **Sous-mot** (voir exercice 2.4). Indice : on pourra utiliser la machine déterministe qui décide **Préfixe** (exercice 2.3).
  2. **Sous-palindrome4**; entrée : un mot  $x \in \{a, b\}^*$ ; question : est-ce que  $x$  contient un palindrome de taille 4 comme facteur ?

**Exercice 5.**

*Machines qui calculent*

Programmer une machine de Turing qui s'arrête sur toute entrée  $x_1, \dots, x_n$  et, lorsqu'elle s'arrête, laisse inscrit sur son ruban le résultat  $f(x)$ , et seulement cela (peu importe la position de la tête lors de l'arrêt), pour chacune des fonctions suivantes.

1.  $f_1 : \mathbb{N} \rightarrow \mathbb{N}$  définie par  $f(x) = 8x + 3$  avec  $x$  un entier écrit en binaire sur l'alphabet  $\{0, 1\}$ .
2.  $f_2 : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$  définie par  $f(x, y) = x + y$  avec  $x$  et  $y$  écrit en binaire et séparés par un symbole #, sur l'alphabet  $\{0, 1, \#\}$ .

**Exercice 6.**


*Machines non-déterministes+*

Même questions que l'exercice 4 avec le problème suivant.

1. **Part somme**; entrée : un ensemble d'entiers  $S = \{x_1, \dots, x_n\}$  codés en binaire et séparés par des symboles #, sur l'alphabet  $\{0, 1, \#\}$ ; question :  $\exists T \subseteq S$  tel que  $\sum_{x_i \in T} x_i = \sum_{x_i \in S \setminus T} x_i$  ?

**Exercice 7.**

???

 Lancer la machine suivante sur l'entrée qui ne contient que des symboles 0 (il faut donc remplacer 0 par \_ avec jsturing) et deviner son comportement (la machine s'arrête lorsqu'elle est dans l'état  $q_5$  et lit un 0).

