

---

**TD 01 – Machinations**


---

**Exercice 1.***Échauffements*

Construire (explicitement) des machines de Turing effectuant les opérations suivantes, et donner leur temps de calcul :

1. somme de deux entiers binaires séparés par une case blanche ;
2. produit de deux entiers unaires séparés par une case blanche.

**Exercice 2.***Football*

Trouver la manière la plus efficace pour simuler une machine de Turing

1. à  $k$  rubans par une machine à un ruban (bonus : peut-on faire mieux avec 2 rubans ?) ;
2. travaillant sur un alphabet  $\Gamma$  par une machine travaillant sur l'alphabet  $\{0, 1, \square\}$  ;
3. à ruban bi-infini par une machine à ruban mono-infini.

**Définition.** Une machine de Turing est dite *oublieuse*<sup>1</sup> si sur une entrée  $x$ , la position de la tête de lecture à l'instant  $i$  ne dépend que de  $i$  et de  $|x|$ .

4. Expliquer comment simuler une machine de Turing par une machine oublieuse, et estimer le temps de la simulation.

**Exercice 3.***Les machines rament*

Le but de cet exercice est de montrer que les modèles de machine de Turing et de RAM sont équivalents. Il existe plusieurs variantes de RAM, nous en fixons donc une.

**Définition.** Une *Random Access Machine* (RAM) sur l'alphabet  $\{a_1, \dots, a_k\}$  est un ensemble infini de registres  $\{R_1, R_2, \dots\}$  pouvant chacun contenir un mot, un ensemble fini ordonné de noms de ligne distincts  $(N_i)_{i \geq 1}$  et une instruction par ligne du type :

1. `addj R` : ajoute  $a_j$  à la droite du registre  $R$  ( $1 \leq j \leq k$ ) ;
2. `del R` : efface la lettre la plus à gauche du registre  $R$  si elle existe ;
3. `clr R` : met le mot vide dans le registre  $R$  ;
4.  $R \leftarrow R'$  : copie le mot de  $R'$  dans  $R$  en le laissant aussi dans  $R'$  ;
5. `jmp N` : va à la ligne  $N$  ;
6.  $R \text{ jmp}_j N$  : va à la ligne  $N$  si la première lettre du mot dans  $R$  est  $a_j$ , sinon va à la ligne suivante ( $1 \leq j \leq k$ ) ;
7. `stop` : fin du programme.

Une RAM restreinte aux règles (1), (2), (6) et (7) et ne disposant que d'un unique registre est appelée une *Single Register Machine* (SRM).

1. Soient  $u_1, \dots, u_n$  des mots sur l'alphabet  $\{a_1, \dots, a_k\}$ . Construire une RAM sur l'alphabet  $\{a_1, \dots, a_k, \#\}$  qui sur l'entrée  $u_1\#u_2\#\dots\#u_n$  renvoie  $u_2\#\dots\#u_n\#u_1$ . Et sur SRM ?
2. Montrer qu'on peut simuler une RAM par une SRM.
3. Montrer qu'on peut simuler toute RAM sur une machine de Turing. Évaluer le temps de la simulation. (Bonus : et dans l'autre sens ?)

---

1. *Oblivious*, en anglais.