TD 02 – Machines de Turing et codage

Exercice 1. $\langle M \rangle$

Donner le code $\langle M \rangle$ de la machine de Turing $M = (Q, \Sigma, \Gamma, \delta, q_0, B, q_F)$, d'ensemble d'états $Q = \{q_0, q_1, q_F\}$, d'alphabets $\Sigma = \{0, 1\}$ et $\Gamma = \{0, 1, B\}$, et dont la fonction de transition est donnée par :

δ	0	1	B
q_0	$(q_1, 1, \rightarrow)$	$-$	$-$
q_1	$-$	$(q_1, 0, \rightarrow)$	$(q_F, B, \leftarrow)$

Exercice 2. *L'arrêt*

Indiquer si chacun des énoncés qui suit est vrai ou faux, en justifiant.

1. $\nexists M_{halt}, \forall M, \forall w : M_{halt}(\langle M \rangle, w) = halt(\langle M \rangle, w)$.
2. $\forall M, \forall w, \nexists M_{halt} : M_{halt}(\langle M \rangle, w) = halt(\langle M \rangle, w)$.

Exercice 3. *Espace et temps*

Supposons qu'une machine de Turing s'arrête au bout de t étapes de calcul en visitant (possiblement plusieurs fois chacune) s cases du ruban. Quelle(s) relation(s) existe(nt) entre t et s ?

Exercice 4. *Codage*

On définit une fonction de codage *code*$_1$ des couples d'entiers de la manière suivante : étant donnés deux entiers x et y, on pose

$$\mathit{code}_1(x, y) = x_1 0 x_2 0 \ldots x_{n-1} 0 x_n \, 1 \, y_1 y_2 \ldots y_m,$$

où $x_1 x_2 \ldots x_n$ et $y_1 y_2 \ldots y_m$ sont les représentations binaires respectives de x et y.

1. Ce codage est-il injectif ? (Autrement dit, un tel code correspond-il bien à un unique couple d'entiers ?)
2. Donnez le codage du couple $(10, 5)$.
3. Quel est le couple associé au codage 100010100010111001110 ?
4. Quel est le nombre de bits utilisés pour coder un couple (x, y) ?

On définit un nouveau codage *code*$_2$ en posant maintenant

$$\mathit{code}_2(x, y) = t_1 0 t_2 0 \ldots t_{\ell-1} 0 t_\ell \, 1 \, x_1 x_2 \ldots x_n \, y_1 y_2 \ldots y_m,$$

où $x_1 x_2 \ldots x_n$ et $y_1 y_2 \ldots y_m$ sont les représentations binaires respectives de x et y, et où $t_1 t_2 \ldots t_\ell$ est la représentation binaire de l'entier n (qui est lui-même la taille de la représentation binaire de x).

5. Ce codage est-il injectif ?
6. Donnez le codage du couple $(10, 5)$.

7. Quel est le couple (x, y) associé au codage 100011100011001 ?

8. Quel est le nombre de bits utilisés pour coder un couple (x, y) ?

On souhaite généraliser ce codage aux tuples d'entiers $(x^1, \ldots, x^k)$ pour k quelconque.

9. Proposez un tel codage.

10. Donnez le codage du tuple $(6, 11, 10, 3)$.

11. Donnez le nombre de bits utilisés pour coder un tuple arbitraire $(x^1, \ldots, x^k)$.

Exercice 5. *MT : conventions*

Objectif : voir que l'on peut utiliser d'autres conventions.

1. Peut-on décider exactement les mêmes langages si la tête de lecture est initialement placée sur la case la plus à droite du mot d'entrée ? (Justifier)

2. Peut-on décider exactement les mêmes langages si l'on autorise des transitions pour lesquelles la tête de lecture/écriture ne bouge pas (ni $\leftarrow$ ni $\rightarrow$ mais $\downarrow$) ? (Justifier)

3. Peut-on décider exactement les mêmes langages si l'on ajoute la restriction $\Sigma = \{0, 1\}$? Et si en plus $\Gamma = \{0, 1, B\}$? (Justifier)

Exercice 6. *MT : multi-ruban*

Objectif : montrer que le modèle des machines de Turing à plusieurs rubans et plusieurs têtes de lecture (une tête de lecture indépendante par ruban, et un seul état pour toute la machine) est équivalent au modèle des machines de Turing. Le multi-ruban est très pratique !

En fonction de l'état et du symbole lu sur chacun des rubans, la machine peut
— changer d'état,
— écrire un symbole sur chaque ruban,
— déplacer chaque tête vers la droite ou la gauche indépendamment les unes des autres.

1. Donner le type de la fonction de transition δ, et donner un exemple de transition.

Dans l'état initial, l'entrée est écrite sur le premier ruban et tous les autres rubans sont vides. Un mot est accepté si et seulement si la machine entre dans l'état final q_F au cours du calcul.

2. Comment simuler une MT *multi-ruban* avec une MT *mono-ruban* ?

Exercice 7. *MT et pseudo-code*

Soit le pseudo-code suivant.

```
h(x : tableau de bits de taille n)
  i <- n
  tant que ( x[i] == 1 et i >= 1 ) faire
    i <- i-1
  fin tant que
  si i == 0 alors
    accepter
  sinon
    rejeter
  fin si
```

1. Quel est le langage reconnu/accepté par l'algorithme h ?

2. Donner une machine de Turing qui décide le même langage que h, et qui s'arrête toujours.
3. Donner une machine de Turing qui calcule la fonction $f : \mathbb{N} \to \mathbb{N}$ définie par :

$$f(n) = \begin{cases} n/2 & \text{si } n \text{ est pair,} \\ n & \text{sinon.} \end{cases}$$

4. Donner une machine de Turing qui calcule la fonction $g : \mathbb{N} \to \mathbb{N}$ telle que $g(n) = 2^n$ pour tout $n \in \mathbb{N}$.
5. Sauriez-vous construire une machine de Turing pour convertir un entier codé en binaire en un entier de même valeur codé en unaire ?