TD n° 9

Machines de Turing

BIJECTIONS

Exercice 9.1. Soient A et B deux ensembles, une fonction $f : A \to B$ est
- injective ssi $\forall\, a, a' \in A : f(a) = f(a') \Rightarrow a = a'$ (ou la contraposée),
- surjective ssi $\forall\, b \in B : \exists\, a \in A : b = f(a)$,
- bijective ssi elle est à la fois injective et surjective.

1. Donner dix éléments de l'ensemble $\mathbb{N} \times \{0, 1, a\}$.
2. Donner une bijection de $\mathbb{N} = \{0, 1, 2, \ldots\}$ dans $2\mathbb{N} = \{0, 2, 4, \ldots\}$.
3. Donner une bijection de $\mathbb{N}$ dans $\mathbb{N} \times \mathbb{N}$.
4. Donner une bijection de $\mathbb{N}$ dans $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$.
5. Donner une bijection de Σ^* dans $\mathbb{N}$.

PROGRAMMER AVEC DES MACHINES DE TURING

Exercice 9.2.
Objectif : voir que l'on peut sauvegarder un symbole (ou plusieurs) dans l'état.
Soit $M = (Q, \Sigma, \Gamma, \delta, q_0, B, q_F)$ la machine de Turing où
- $Q = \{q_0, q_a, q_b, q_F\}$,
- $\Sigma = \{a, b\}$, $\Gamma = \{a, b, B\}$,
- δ est donnée par

$$\begin{array}{llll}
(q_0, a) \mapsto (q_a, a, R) & & (q_0, b) \to (q_b, b, R) & \\
(q_a, a) \mapsto (q_a, a, R) & (q_a, b) \mapsto (q_a, b, R) & (q_b, a) \to (q_b, a, R) & (q_b, b) \to (q_b, b, R) \\
(q_a, B) \mapsto (q_F, a, R) & & (q_b, B) \to (q_F, b, R) &
\end{array}$$

1. Dessiner cette machine sous la forme d'un automate.
2. Dans quelle configuraiton est-on à la fin de l'exécution de M sur le mot d'entrée *abab* ?
3. Quelle fonction est calculée par cette machine ?

Exercice 9.3.
Objectif : voir que l'on peut encoder plusieurs rubans en un seul (en augmentant l'alphabet).
Nous allons définir une machine de Turing qui reconnait le langage

$$L = \{w \in \{a, b\}^* \mid w = w_0 w_1 \ldots w_{2k} \text{ pour } k \in \mathbb{N}, \text{ et } w_k = a\}.$$

L'idée est la suivante. On utilise un ruban sur deux lignes :
- la seconde ligne contient w
- sur la première nous utilisons un marqueur $\surd$ ou le symbole B.

Inialement il n'y a que des B sur le premier ruban, et un mot $u_1 u_2 \ldots u_j$ sur le second ruban. Nous allons marquer la case u_0, puis u_j, puis u_1, puis u_{j-1}, puis u_2, etc, jusqu'à trouver le milieu ! Il ne reste alors qu'à dire si c'est bien un symbole a.

1. Donner les alphabets d'entrée et de ruban pour cette machine.
2. Dessiner l'automate d'une machine qui reconnait ce langage.

Exercice 9.4.
Objectif : voir que l'on peut facilement faire un décalage avec une MT.
Soit $\Sigma = \{a, b\}$, et la fonction $f : \Sigma^* \to \Sigma^*$ définie par

$$\text{pour tout } w \in \Sigma^* \text{ on a } f(w) = aw$$

Le but étant de décaler tous les symboles d'une case sur la droite (afin de réutiliser cette machine ensuite).

1. Dessiner l'automate d'une machine qui calcule f.

Exercice 9.5.
Objectif : voir que l'on peut composer des MT.
Soit $\Sigma = \{a, b\}$, et la fonction $g : \Sigma^* \to \Sigma^*$ définie par

$$\text{pour tout } w_0 w_1 \dots w_{2k} \in \Sigma^* \text{ on a } g(w) = w_0 \dots w_{k-1} a w_k \dots w_{2k}$$

Pour simplifier on supposera que les entrées sont toujours de longueur impaire.

1. Dessiner l'automate d'une machine qui calcule g (il est possible de composer deux machines simples pour cela !).

Exercice 9.6.
Objectif : voir que l'on peut utiliser d'autres conventions.

1. Peut-on calculer exactement les mêmes fonctions / 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 calculer les mêmes fonctions / décider les mêmes langages si l'on autorise des transitions pour lesquelles la tête de lecture/écriture ne bouge pas (ni L ni R) ? (Justifier)
3. Peut-on calculer les mêmes fonctions / décider les mêmes langages si l'on ajoute la restriction $\Sigma = \{0, 1\}$? Et si en plus $\Gamma = \{0, 1, B\}$? (Justifier)

PROPRIETES DE CLOTURE

Exercice 9.7. Démontrer les résultats suivants.

1. la famille des langages récursifs est close par complémentation ;
2. les familles des langages récursifs et récursivement énumérables sont closes par union et intersection ;
3. Un langage $L \subseteq \Sigma^*$ est récursif si et seulement si L et $\Sigma^* \setminus L$ sont récursivement énumérables.