TD 2 - Machines de Turing

Dans ce TD, on considère des machines opérant sur un ruban semi-infini (indexé par les entiers naturels). On note # le symbole case vide, et on suppose que la configuration initiale associée au mot w d'une machine ayant comme état initial q_0 est $q_0\#w$: autrement dit le pointeur est initialement sur la première case du ruban, qui est initialement laissée vide.

Première partie – Manipulation du ruban

Exercice 1. Déplacements sur le ruban

Question 1. - *Avancer jusqu'à un symbole* - Soit $\Sigma = \{\mathtt{a}, \mathtt{b}, \ell\}$ un alphabet d'entrée. Écrire une machine de Turing A_ℓ qui avance la tête de lecture jusqu'à la première occurence du symbole ℓ, puis recule pour se placer sur la case immédiatement à gauche. Cette machine refuse les mots qui ne contiennent pas le symbole ℓ.

Question 2. - *Remplacer* - Soit $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ l'alphabet d'entrée. Écrire une machine $R_{\mathtt{a},\mathtt{b}}$ qui échange les $\mathtt{a}$ et les $\mathtt{b}$ du mot d'entrée (en laissant les $\mathtt{c}$ inchangés) et qui s'arrête sur la première case du ruban.

Exercice 2. Copies et effacements

On fixe pour cet exercice l'alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$. Les symboles ℓ et w seront utilisés respectivement pour dénoter un symbole de Σ et un mot de $\Sigma^\star$.

Question 1. - *Effacer* - Écrire une machine E qui, en partant d'un ruban $\underline{\#}\ell w$, s'arrête avec le ruban $\underline{\#}w$.

Question 2. - *Effacer jusqu'à un symbole* - Écrire une machine E_ℓ qui efface toutes les lettres de son entrée jusqu'au premier $\ell \in \Sigma$ inclus.

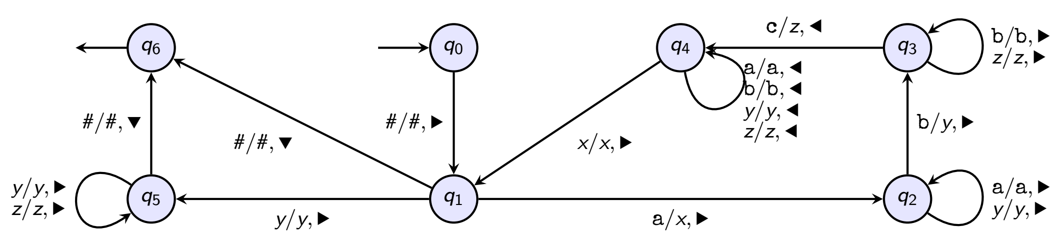
Question 3. - *Insérer* - Écrire trois machines I_ℓ, I'_ℓ, et I''_ℓ qui prennent en entrée un mot $w \in \Sigma^\star$ (donc un ruban $\underline{\#}w$) et s'arrêtent respectivement avec les rubans $\underline{\#}\ell w$, $\underline{\#}w\ell$, et $\#w\ell\underline{\#}$.

Question 4. - *Copier* - Soit Γ un alphabet de travail contenant Σ, et $\ell, m \in \Gamma \setminus (\Sigma \cup \{\#\})$ deux symboles de travail. Écrire une machine C_ℓ^m qui prend en entrée un mot $w_1\ell w_2$, avec $w_1, w_2 \in \Sigma^\star$, et s'arrête avec le ruban $\underline{\#}w_1\ell w_2 m w_1$.

Deuxième partie – Reconnaissance de langages

Exercice 3. Langage mystère

Quel est le langage sur l'alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ reconnu par la machine suivante ?



Exercice 4. Langages réguliers

Question 1. - *Sous-mot* - Écrire une machine de Turing acceptant les mots sur l'alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ qui contiennent le sous-mot $\mathtt{aab}$ et se terminent par un $\mathtt{b}$ (et refusant tous les autres mots).

Question 2. Écrire une machine de Turing acceptant le langage $[\![\mathtt{a}^\star\mathtt{b}\mathtt{a}^\star\mathtt{b}]\!]$.

Question 3. Écrire une machine de Turing acceptant le langage

$$\left\{ w \in \{\mathtt{a},\mathtt{b},\mathtt{c}\}^\star \;\middle|\; w \text{ contient } \mathtt{abc} \text{ mais pas } \mathtt{bb} \right\}.$$

Exercice 5. Langages hors-contextes

Question 1. Écrire une machine de Turing acceptant les mots bien parenthésés sur l'alphabet $\Sigma = \{\mathtt{a}, (,)\}$, c'est à dire le langage décrit par la grammaire $S \to \varepsilon \mid \mathtt{a} \mid SS \mid (S)$.

Question 2. Écrire une machine de Turing acceptant le langage $\{\mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}\}$.

Exercice 6.

On considère le langage $L := \left\{\mathtt{a}^{2^n} \;\middle|\; n \in \mathbb{N}\right\}$.

Question 1. Proposer une machine reconnaissant L.

Question 2. Montrer que ce langage n'est pas régulier.

Exercice 7. Les carrés

Les machines de Turing peuvent reconnaître plus que les langages hors-contexte. C'est le cas par exemple du langage $L := \left\{ww \;\middle|\; w \in \{\mathtt{a},\mathtt{b}\}^\star\right\}$ des carrés. On peut montrer, par exemple avec le lemme de la double étoile, que ce langage n'est pas hors-contexte : il ne peut être défini à l'aide d'une grammaire hors-contexte.

Question 1. Écrire une machine de Turing reconnaissant le langage $w\mathtt{X}w$, où $w \in \{\mathtt{a},\mathtt{b}\}^\star$.

Question 2. Écrire une machine de Turing qui rejette les mots de longueur impaire, et pour les mots de longueur paire insère le symbole $\mathtt{X}$ entre les deux moitiés du mot. Par exemple, le mot $\mathtt{abcd}$ est transformé en $\mathtt{abXcd}$.

Question 3. Combiner les machines des deux questions précédentes pour en produire une reconnaissant L.

Question 4. Montrer que ce langage n'est pas rationnel.

Troisième partie – Arithmétique

Exercice 8. Calculs unaires

On se donne un alphabet $\{\mathtt{a},\mathtt{b}\}$. On code l'entier n sur cet alphabet par le mot $\mathtt{a}^n$, c'est à dire le mot constitué de n fois la lettre $\mathtt{a}$. On pourra également coder une séquence finie d'entiers $\langle n_1, \ldots, n_m\rangle$ par le mot $\mathtt{a}^{n_1}\mathtt{b}\mathtt{a}^{n_2}\cdots\mathtt{b}\mathtt{a}^{n_m}$.

Question 1. Écrire une machine qui « calcule zéro », c'est à dire qu'elle prend en entrée le codage d'un entier et produit en sortie le codage de 0.

Question 2. Écrire une machine qui incrémente son entrée d'une unité, c'est à dire qu'elle prend en entrée le codage d'un entier n et produit en sortie le codage de $n+1$.

Question 3. Écrire une machine qui calcule l'addition, c'est à dire qu'elle prend en entrée le codage d'une paire d'entiers $\langle n, m\rangle$ et produit en sortie le codage de $n+m$.

Exercice 9. Calculs binaires

On se donne l'alphabet $\{0, 1, |\}$. On code l'entier n en binaire sur cet alphabet, avec le bit de poids fort à droite. Par exemple, les 8 premiers entiers sont codés comme suit :

$$\begin{array}{llll} 0 \mapsto 0 & 2 \mapsto 01 & 4 \mapsto 001 & 6 \mapsto 011 \\ 1 \mapsto 1 & 3 \mapsto 11 & 5 \mapsto 101 & 7 \mapsto 111. \end{array}$$

On note $[n]_2$ le code de l'entier n. On pourra également coder une séquence finie d'entiers $\langle n_1, \ldots, n_m\rangle$ par le mot $[n_1]_2 \,|\, [n_2]_2 \cdots | \,[n_m]_2$. Par exemple la paire $\langle 2, 5\rangle$ est représentée par $01|101$.

Question 1. Écrire une machine qui « calcule zéro », c'est à dire qu'elle prend en entrée le codage d'un entier et produit en sortie le codage de 0.

Question 2. Écrire une machine qui incrémente son entrée d'une unité, c'est à dire qu'elle prend en entrée le codage d'un entier n et produit en sortie le codage de $n + 1$.

Question 3. Écrire une machine qui calcule l'addition, c'est à dire qu'elle prend en entrée le codage d'une paire d'entiers $\langle n, m\rangle$ et produit en sortie le codage de $n + m$.

Question 4. Écrire une machine qui calcule la multiplication, c'est à dire qu'elle prend en entrée le codage d'une paire d'entiers $\langle n, m\rangle$ et produit en sortie le codage de $n \times m$.