TD 11 : Fonctions séquentielles

Exercice 1 (**À préparer -** Machine de MOORE)**.** Une *machine de* MOORE de A dans B est un tuple $\mathcal{M} = \langle Q, A, B, \delta, \gamma, q_0 \rangle$ où Q est un ensemble fini d'états, $q_0 \in Q$ un état initial, A un alphabet d'entrée, B un alphabet de sortie, δ une fonction partielle de transition de $Q \times A$ dans Q, et γ une fonction de sortie de Q dans B^*. La fonction partielle $[\![\mathcal{M}]\!] : A^* \to B^*$ définie par $\mathcal{M}$ retourne la séquence des sorties associées par un run dans $\mathcal{M}$.

1. Montrer que si $\mathcal{M}$ est une machine de MOORE, alors on peut donner une fonction séquentielle équivalente.
2. Montrer que si $\mathcal{A}$ est une fonction séquentielle (pure, ou machine de MEALY), alors on peut donner une machine de MOORE équivalente.

Exercice 2 (**À préparer -** Fonction séquentielle pure)**.** Soient $\theta : A^* \to B^*$ une fonction et \$ un symbole qui n'appartient pas à A. On définit la fonction $\tau : (A \cup \{\$\})^* \to B^*$ par, pour tout w de A^* :

$$\tau(w\$) = \theta(w)$$

où $\mathsf{dom}(\tau) = \mathsf{dom}(\theta) \cdot \{\$\}$. Montrer que θ est séquentielle si et seulement si τ est séquentielle pure.

Exercice 3 (**À préparer -** Codes à délai de déchiffrage borné)**.**

1. Soit $\beta_1 : \{x, y\}^* \to A^*$ le morphisme défini par $\beta_1(x) = a$ et $\beta_1(y) = aba$. Montrer que ${\beta_1}^{-1}$ est une fonction séquentielle.
2. Même question pour $\beta_2 : \{x, y, z\}^* \to A^*$ défini par $\beta_2(x) = ab$, $\beta_2(y) = abb$ et $\beta_2(z) = baab$.
3. Généralisation. Soit $\beta : B^* \to A^*$ un morphisme. Par définition, $X = \beta(B)$ est un code si β est injectif. Le code X est dit à *délai de déchiffrage* d si, lorsqu'un mot $f = x_1 x_2 \dots x_{d+1} \in X^{d+1}$ est un préfixe d'un mot $g = y_1 y_2 \dots y_r \in X^*$, alors $x_1 = y_1$. (En particulier, un code est dit *préfixe* s'il a un délai de déchiffrage 0.)
   Montrer que l'ensemble $X = \beta(B)$ est un code à délai de déchiffrage fini si et seulement si β^{-1} est une fonction séquentielle.

Exercice 4 (Addition d'Avizienis)**.** Soit $k > 1$ un entier. Un système d'Avizienis est un procédé de numération en base k qui utilise des chiffres positifs et négatifs. De façon générale, si $D \subseteq \mathbb{Z}$ est un ensemble fini de chiffres, à chaque mot $u = a_n \cdots a_1 a_0 \in D^*$ (avec $a_i \in D$) on associe sa *valeur*, i.e., l'entier représenté par u en base k :

$$\mathsf{val}_k(u) = \sum_{i=0}^{n} a_i k^i \ .$$

Pour plus de lisibilité dans l'écriture d'un mot on utilisera de préférence $\bar{a}$ à la place du chiffre $-a$. Par exemple, $\mathsf{val}_2(10\bar{1}) = 3$ et $\mathsf{val}_{10}(3\bar{3}) = 27$.

Avec $A = \{0, \dots, k-1\}$, les mots de $X = (A \setminus \{0\})A^* \cup \{\varepsilon\}$ correspondent à l'écriture usuelle des entiers en base k en commençant par le chiffre de poids fort. Cette écriture étant unique, l'application $\mathsf{val} : X \to \mathbb{N}$ est une bijection.

Soit $h = \lfloor \frac{k+1}{2} \rfloor$ et $B = \{-h, \dots, 0, \dots, h\}$. Le système de numération d'Avizienis correspond aux mots de B^* (on peut éventuellement se restreindre à ceux qui ne commencent pas par 0). Dans ce système, la représentation d'un entier n'est pas unique. Par exemple, $\mathsf{val}_4(12\bar{2}) = \mathsf{val}_4(112)$.

1. Montrer qu'il existe une fonction séquentielle $f : A^* \to B^*$ qui préserve les valeurs, i.e., telle que $\mathsf{val}_k(u) = \mathsf{val}_k(f(u))$ pour tout $u \in A^*$.
2. On considère maintenant l'alphabet $C = \{-2h, \dots, 0, \dots, 2h\}$. Montrer qu'il existe une fonction séquentielle $f : C^* \to B^*$ qui préserve les valeurs, i.e., telle que $\mathsf{val}_k(u) = \mathsf{val}_k(f(u))$ pour tout $u \in C^*$.
   Indication : On pourra prendre comme ensemble d'états $Q = B^2 \setminus \{\bar{h}\bar{h}, hh\}$.
3. Existe-t-il une fonction séquentielle $g : (B \times B)^* \to B^*$ qui réalise l'addition dans le système d'Avizienis en commençant par le bit de poids fort, i.e., telle que pour tout $w = (a_n, b_n) \cdots (a_0, b_0) \in (B \times B)^*$ on a $\mathsf{val}_k(g(w)) = \mathsf{val}_k(a_n \cdots a_0) + \mathsf{val}_k(b_n \cdots b_0)$?

Exercice 5 (Formules de Presburger)**.** On considère les *fonctions affines* de la forme

$$\psi(x_1, \dots, x_n) = a_0 + \sum_{i=1}^{n} a_i x_i$$

où les a_i sont des entiers naturels, et les x_i des variables à valeurs dans $\mathbb{N}$. On appelle *formule atomique* une formule de la forme $f(\psi_1, \psi_2)$, avec ψ_1 et ψ_2 des fonctions affines et f un opérateur de comparaison du type $<, \leq, =, \geq, >$.

On définit l'ensemble des formules de Presburger comme l'ensemble obtenu à partir des formules atomiques en le fermant par combinaison booléenne (opérateurs $\wedge$, $\vee$ et $\neg$) et par quantification existentielle ($\exists$) et universelle ($\forall$).

L'objectif de cet exercice est de montrer que les formules de Presburger sont reconnaissables par automate, *i.e.* que pour toute formule de Presburger φ, il existe un automate fini $\mathcal{A}_\varphi$ qui reconnaît exactement les codages, avec bits de poids faible en premier, en binaire satisfaisant la formule φ.

1. Montrer que l'on peut se restreindre à l'opérateur de comparaison $=$.
2. Donner un automate séquentiel qui réalise la fonction $(x, y) \mapsto 2x + y$ sur les codages binaires.
3. Montrer que toute fonction affine est séquentielle.
4. Montrer que toute formule de Presburger est reconnaissable par automate.

5. En déduire que l'arithmétique de Presburger est décidable, i.e., qu'on peut décider si une formule close est valide. Quelle est la complexité de cette procédure de décision ?

Exercice 6. Faire *Fonctions séquentielles et logique temporelle* de Examen 2011.