TD 11 : Fonctions séquentielles

Exercice 1 (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 à un calcul 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 (Normalisation)**.** Soient $k > 1$ et $h \geq k$ deux bases entières ; on note $K = \{0, \ldots, k-1\}$ et $H = \{0, \ldots, h-1\}$ les alphabets d'écriture correspondants. On fait correspondre à un mot $u = a_n \cdots a_0$ de H^* écrit en base k une *valeur* $\mathsf{val}_k(u)$ définie comme

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

La fonction de *normalisation* $\nu_{h,k}$ de H^* dans K^* préserve les valeurs en base k :

$$\mathsf{val}_k(\nu_{h,k}(u)) = \mathsf{val}_k(u) \ .$$

Par exemple pour $h = 4$ et $u = 333$, $\mathsf{val}_2(u) = 21$, $\nu_{4,2}(u) = 10101$.

1. Donner une fonction co-séquentielle pour $\nu_{4,2}$, c-à-d. séquentielle si on procède de droite à gauche (cela revient aussi dans notre cas à mettre le chiffre de poids faible en premier).
2. Montrer que $\nu_{h,k}$ est en général co-séquentielle.
3. En déduire que l'addition de deux nombres en base k et que la multiplication par une constante n sont co-séquentielles.
4. Appliquer cette construction à l'addition en base 2.

Exercice 3 (Addition d'Avizienis)**.** Soit encore $k > 1$. Un système d'Avizienis est un procédé de numération en base k qui utilise des chiffres positifs et négatifs. On associe comme dans l'exercice précédent à un mot $u = a_n \cdots a_0$ sa valeur $\mathsf{val}_k(u)$, à la différence près que les chiffres a_i peuvent être négatifs (on notera dans ce cas $\overline{a}$ plutôt que $-a$), par exemple $\mathsf{val}_2(10\overline{1}) = 3$ et $\mathsf{val}_{10}(3\overline{3}) = 27$.

Soit $h = \lfloor \frac{k+1}{2} \rfloor$ et $H = \{-h, \ldots, 0, \ldots, h\}$. Le système d'Avizienis correspond aux mots de H^*. Dans ce système, la représentation n'est pas unique : par exemple $\mathsf{val}_4(12\overline{2}) = \mathsf{val}_4(112) = 22$. On appelle à nouveau *normalisation* la fonction de H^* dans K^* qui préserve les valeurs.

1. Montrer que cette normalisation est elle aussi co-séquentielle.
2. Montrer qu'il existe une fonction séquentielle $f : K^* \to H^*$ qui préserve les valeurs.
3. On considère maintenant l'alphabet $H' = \{-2h, \ldots, 0, \ldots, 2h\}$. Montrer qu'il existe une fonction séquentielle $g : H'^* \to H^*$ qui préserve les valeurs. Comment l'utiliser pour réaliser l'addition de manière séquentielle ?

Exercice 4 (Fonction non séquentielle)**.** Soit $B = \{0, 1\}$; montrer que la fonction $\cdot^2 : B^* \to B^*$ qui à un entier codé en binaire associe le codage de son carré n'est ni séquentielle ni co-séquentielle.

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

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

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, c-a-d. que pour toute formule de Presburger φ, il existe un automate fini $\mathcal{A}_\varphi$ qui reconnaît exactement les codages 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 (on pourra dans un premier temps réaliser $(x, y) \mapsto (2x, y)$).
3. Montrer que toute fonction affine est séquentielle.
4. Montrer que toute formule de Presburger est reconnaissable par automate pour le codage avec bits de poids faible en premier.