Informatique théorique III
Série 2

Prof. Nestmann, 2004

1. Mots et Langages

1. Soit $\Sigma \triangleq \{a, b\}$ et les langages $L = \{abb, b, a, \epsilon\}$ et $L' = \{ba, baa\}$, calculer les langages suivant, $L \cup L', L \cap L', LL', L'L, L^0, L^2$.
2. Étant donné un alphabet Σ, définir par induction sur la structure des mots la fonction $|w|_a$ qui calcule le nombre d'occurences d'une lettre $a \in \Sigma$ dans un mot $w \in \Sigma^*$. □

2. Grammaires

Soit $\Sigma \triangleq \{a, b\}$.

1. Donner une grammaire G qui engendre les palindromes sur Σ, c'est à dire les mots $w \in \Sigma^*$ tels que $\overline{w} = w$ où $\overline{w}$ est le mot miroir de w défini dans la série 1.
2. Donner le type de G. Justifier.
3. Montrer que les mots $abba$, $ababa$, ϵ et a sont engendrés par G en exhibant une dérivation pour chacun de ceux-ci. □

3. Construction d'AFD

Pour chacun des langages suivants, donner le diagramme d'un AFD le reconnaissant.

1. L'ensemble des mots sur $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ qui ne contiennent pas le sous-mot 492.
2. L'ensemble des mots sur $\{a, b\}$ de la forme $a^n b^m a^r$ tel que $n + m + r$ est pair et $m \geq 1$. □

4. Exclusion mutuelle

Dans un atelier, afin d'éviter les incidents, il n'est généralement pas souhaitable que deux ouvriers utilisent en même temps le même outil (marteau, escabeau, ...). En programmation concurrente, il est souvent souhaitable que deux programmes concurrents n'accèdent pas simultanément à la même ressource (fichier, imprimante, mémoire, ...). C'est le problème de *l'exclusion mutuelle*.

Nous souhaitons exprimer ce problème avec un AFD pour le cas simple d'un système à une ressource accédée en parallèle par deux programmes P_1 et P_2. On suppose que ces derniers sont exécutés par un seul processeur (système à *temps partagé*). Ceci a pour conséquence que les actions de P_1 et P_2 sont *entrelacées*, deux actions atomiques ne sont pas réellement exécutées au même instant. Les exécutions finies de ce système peuvent donc être représentée par un mot w qui caractérise la séquence des actions atomiques exécutées par P_1 et P_2 sur le processeur.

Nous considérons les actions $\Sigma \triangleq \{p_1, l_1, c_1, p_2, l_2, c_2\}$. Pour $i \in \{1, 2\}$, le symbole p_i représente la **p**rise de la ressource par le programme P_i, l_i la **l**ibération de la ressource par P_i et c_i l'exécution d'une opération atomique de **c**alcul par P_i.

Une exécution w satisfait la propriété d'exclusion mutuelle si et seulement si,

(a) Le nombre de prises de la ressource par P_i est égal au nombre de libérations de la ressource par P_i.

(b) Il n'y a pas de libération de la ressource par P_i sans une prise préalable par P_i, ni deux libérations par P_i sans prise intermédiaire.

(c) Il n'y a pas de prise simultanée par P_1 et P_2 de la ressource, ni deux prises par P_i sans libération intermédiaire.

Résolvez les points suivants,

1. Quelles sont les chaînes données ci-dessous qui ne satisfont pas la propriété d'exclusion mutuelle. Indiquer quel(s) point(s) elles ne satisfont pas.

$$c_1c_1c_2p_1c_2c_1c_1l_1c_2p_2l_2c_2p_1l_1 \quad (1)$$
$$c_1p_2c_1c_1l_2c_2c_1l_1c_2c_1 \quad (2)$$
$$c_2p_1c_2c_1c_1p_2c_2l_2l_1 \quad (3)$$
$$c_2p_1c_2c_2l_1c_1c_1p_2c_1c_2 \quad (4)$$

2. Donner le graphe d'un automate A_μ qui reconnaît exactement les exécutions qui satisfont la propriété d'exclusion mutuelle.
3. Soit A_μ un automate qui reconnaît exactement les exécutions qui satisfont l'exclusion mutuelle.
   (a) Caractérisez à l'aide de A_μ (i) l'ensemble des exécutions qui satisfont l'exclusion mutuelle (ii) l'ensemble des exécutions qui ne satisfont pas l'exclusion mutuelle.
   (b) Prenons A un automate quelconque sur Σ. L'ensemble $\mathrm{L}(A)$ caractérise toutes les exécutions finies d'un système formé de deux programmes P_1 et P_2 particuliers.
   Sous quelle condition (dépendante de A_μ) toutes les exécutions de ce système satisfont la propriété d'exclusion mutuelle ?

□

5. Mots conjugués

Soit Σ un alphabet. On définit sur Σ^* la relation R suivante,

$$uRv \Leftrightarrow \exists s, t \in \Sigma^* : u = st \wedge v = ts$$

Si $u, v \in \Sigma^*$ sont tels que uRv, on dit que v est le conjugué de u.
Montrer que :

1. R est une relation d'équivalence.
2. $\forall u, v \in \Sigma^* : \forall m \geq 1 : uRv \Leftrightarrow u^m R v^m$
3. $\forall u, v \in \Sigma^* : uRv \Leftrightarrow \exists w \in \Sigma^* : uw = wv$

Indication : Pour montrer 3, utilisez (puis démontrez) le résultat suivant :
Soit $u, v \in \Sigma^*$ avec $u \neq \epsilon$. S'il existe $w \in \Sigma^*$ tel que $u \cdot w = w \cdot v$ alors il existe $w' \in \Sigma^*$ tel que $|w'| < |u|$ et $u \cdot w' = w' \cdot v$. □