Informatique théorique III
Série 2 solution

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 $w \in \Sigma^*$. □

Solution.

1. $L \cup L' = \{abb, b, a, \epsilon, ba, baa\}$
   $L \cap L' = \emptyset$
   $LL' = \{abbba, bba, aba, ba, abbbaa, bbaa, abaa, baa\}$
   $L'L = \{baabb, baaabb, bab, baab, baa, baaa, ba, baa\}$
   $L^0 = \{\epsilon\}$
   $L^2 = \{abbabb, babb, aabb, abb, abbb, bb, ab, b, abba, ba, aa, a, abb, b, a, \epsilon\}$
2. La fonction $|w|_a$ est la plus petite fonction satisfaisant les équations suivantes,

$$
\begin{aligned}
|\epsilon|_a &\triangleq 0 \\
|b \cdot w|_a &\triangleq |w|_a && \text{si } b \neq a \\
|b \cdot w|_a &\triangleq 1 + |w|_a && \text{si } b = a
\end{aligned}
$$

■

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. □

Solution.

1. $G = (\{S\}, \{a, b\}, P, S)$ avec P tel que

$$
\begin{aligned}
S &\to aSa \\
S &\to bSb \\
S &\to a \\
S &\to b \\
S &\to \epsilon
\end{aligned}
$$

2. La forme des productions satisfait les contraintes imposées par le type 2.
3. $S \Rightarrow aSa \Rightarrow abSba \Rightarrow abba$
   $S \Rightarrow aSa \Rightarrow abSba \Rightarrow ababa$
   $S \Rightarrow \epsilon$
   $S \Rightarrow a$

■

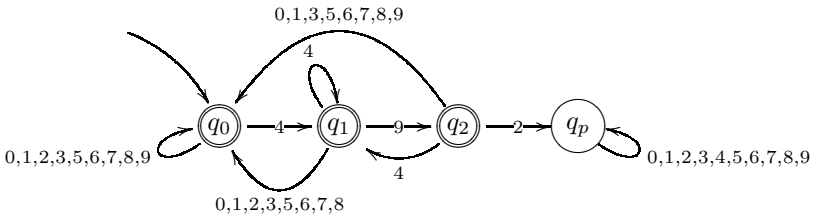
3. Construction d'AFD

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 $m \geq 1$. □

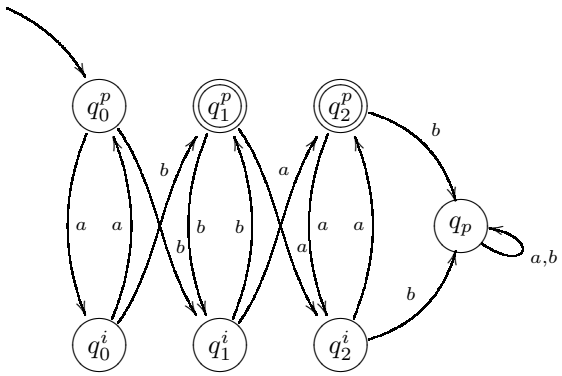
Solution.

1. L'idée est de piéger dans un état puits q_p tous les mots contenant 492.



2. L'idée ici est de dédoubler un automate qui reconnaît les mots $a^n b^m a^r$ avec $m \geq 1$. On obtient donc deux ensembles d'états $\{q_0^p, q_1^p, q_2^p\}$ et $\{q_0^i, q_1^i, q_2^i\}$. Seul l'état puits q_p, qui permet de piéger les mots commençant par $a^n b^m a^r$ puis qui continuent par b suivi d'une ou plusieurs lettres, n'est pas dédoublé.

   A chaque lecture de lettre, on alterne entre les deux ensembles d'états de manière à ce que lorsqu'on se trouve dans un état q_j^p on a lu un nombre pair de lettres et lorsqu'on se trouve dans un état q_j^i on a lu un nombre impair de lettres. Les états finaux sont les états finaux de l'automate initial, mais qui reconnaissent en plus des mots de longueur paire : ce sont donc q_1^p et q_2^p.



■

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.

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

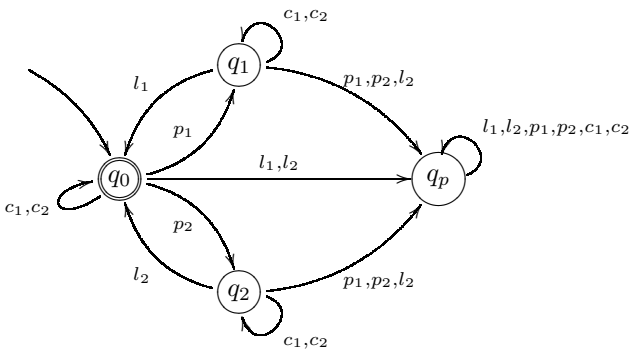
2. Donner le graphe d'un automate A_μ qui reconnaît exactement les exécutions qui satisfont la propriété d'exclusion mutuelle.
3. Soit A_μ un automate qui reconnaît exactement les exécutions qui satisfont l'exclusion mutuelle.
   (a) Caractérisez à l'aide de A_μ (i) l'ensemble des exécutions qui satisfont 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 ?

□

Solution.

1. (2) point (b). (3) point (c). (4) point (a).
2. L'automate A_μ sur Σ donné ci-dessous reconnaît les exécutions qui garantissent l'exclusion mutuelle. L'état q_0 représente l'état du système lorsque aucun programme n'a pris la ressource. Pour $i \in \{1, 2\}$, l'état q_i représente la prise de la ressource par le programme P_i. L'état puits q_p permet d'éviter la reconnaissance par l'automate de certaines exécutions qui ne satisfont pas l'exclusion mutuelle.

$\Sigma \triangleq \{p_1, l_1, c_1, p_2, l_2, c_2\}$
$Q \triangleq \{q_0, q_1, q_2, q_p\}$



3.

(a) (i) C'est le langage $\mathrm{L}(A_\mu)$ reconnu par A_μ.
(ii) C'est le complément $\overline{\mathrm{L}(A_\mu)}$ du langage reconnu par A_μ ($\Sigma^* \setminus \mathrm{L}(A_\mu)$).

(b) Toutes les exécutions caractérisée par l'automate A doivent satisfaire l'exclusion mutuelle, donc $\mathrm{L}(A) \subseteq \mathrm{L}(A_\mu)$.

■

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 $u \cdot w' = w' \cdot v$. □

Solution.

1. On montre que R est une relation d'équivalence :
   (a) R est réflexive :
   Soit $u \in \Sigma^*$. On pose $s = u$ et $t = \epsilon$. On a alors $u = s \cdot t = t \cdot s$. Donc uRu.
   Le mot u étant quelconque, on a donc démontré $\forall u \in \Sigma^* : uRu$, c'est-à-dire que R est réflexive.
   (b) R est symétrique :
   Soit $u, v \in \Sigma^*$ et supposons que uRv. On veut montrer que vRu.
   Par définition de uRv, il existe $s, t \in \Sigma^*$ tels que $u = s \cdot t$ et $v = t \cdot s$.
   Posons $s' = t$ et $t' = s$. On a alors $v = s' \cdot t'$ et $u = t' \cdot s'$ et donc vRu.
   Les mots u et v étant quelconques, on a démontré que $\forall u, v \in \Sigma^* : uRv \Rightarrow vRu$. On en conclut que R est symétrique.

(c) R est transitive :

Soit $u, v, w \in \Sigma^*$. Supposons que uRv et vRw et montrons que uRw.

Par définition de uRv et vRw, il existe $s, t, s', t' \in \Sigma^*$ tels que $u = s \cdot t$, $v = t \cdot s$, $v = s' \cdot t'$ et $w = t' \cdot s'$.

Comme l'ordre sur les entiers est total, on a maintenant deux cas possibles :

– $|s'| \leq |t|$

On a $v = t \cdot s = s' \cdot t'$.

Les mots s' et t sont des préfixes de v et s' est plus petit que t, donc s' est aussi préfixe de t. Il existe donc $s'' \in \Sigma^*$ tel que $t = s' \cdot s''$.

En remplaçant dans v, on a donc $s' \cdot s'' \cdot s = s' \cdot t'$. On en déduit donc que $t' = s'' \cdot s$.

On a donc $u = s \cdot t = s \cdot (s' \cdot s'') = (s \cdot s') \cdot s''$ et $w = t' \cdot s' = (s'' \cdot s) \cdot s' = s'' \cdot (s \cdot s')$. En posant $s_0 = s \cdot s'$ et $t_0 = s''$, on a donc $u = s_0 \cdot t_0$ et $w = t_0 \cdot s_0$.

On a donc uRw.

– $|t| \leq |s'|$

Un raisonnement similaire au cas ci-dessus s'applique.

On a $v = t \cdot s = s' \cdot t'$.

Comme ci-dessus, on en déduit que t est un préfixe de s'. Il existe donc $t'' \in \Sigma^*$ tel que $s' = t \cdot t''$.

En remplaçant dans v, on obtient $t \cdot s = t \cdot t'' \cdot t'$. On en déduit que $s = t'' \cdot t'$.

On a alors $u = s \cdot t = (t'' \cdot t') \cdot t = t'' \cdot (t' \cdot t)$ et $w = t' \cdot s' = t' \cdot (t \cdot t'') = (t' \cdot t) \cdot t''$.

En posant $s_0 = t''$ et $t_0 = t' \cdot t$, on a donc $u = s_0 \cdot t_0$ et $w = t_0 \cdot s_0$ et donc uRw.

Dans les deux cas, on a montré uRw.

Les mots u, v et w étant quelconques, on en déduit que R est transitive.

De 1a, 1b et 1c, on conclut que R est bien une relation d'équivalence.

2. Soit $u, v \in \Sigma^*$ et $m \geq 1$.

Comme $m \geq 1$, il existe $m' \in \mathbb{N}$ tel que $m = 1 + m'$.

Montrons maintenant que $uRv \Leftrightarrow u^m R v^m$.

(a) Supposons uRv et montrons $u^m R v^m$.

Par définition de uRv, il existe $s, t \in \Sigma^*$ tel que $u = s \cdot t$ et $v = t \cdot s$.

On a $u^m = (s \cdot t)^m = (s \cdot t)^{1+m'} = \underbrace{(s \cdot t)(s \cdot t) \ldots (s \cdot t)}_{m \text{ fois}} = s \cdot \underbrace{(t \cdot s) \ldots (t \cdot s)}_{m' \text{ fois}} \cdot t =$ $s \cdot (t \cdot s)^{m'} \cdot t$ et $v^m = (t \cdot s)^m = (t \cdot s)^{1+m'} = (t \cdot s)^{m'} \cdot t \cdot s$. En posant $s' = s$ et $t' = (t \cdot s)^{m'} \cdot t$, on a donc $u^m = s' \cdot t'$ et $v^m = t' \cdot s'$ donc $u^m R v^m$.

(b) Supposons $u^m R v^m$ et montrons uRv.

Par définition de $u^m R v^m$, il existe $s, t \in \Sigma^*$ tel que $u^m = s \cdot t$ et $v^m = t \cdot s$.

On raisonne par cas suivant que u est le mot vide ou non.

– Premier cas : $u = \epsilon$.

Alors $u^m = \epsilon$ et $v^m = \epsilon$ donc $v = \epsilon$. Comme R est réflexive, uRv.

– Second cas : $u \neq \epsilon$.

Alors $|u| > 0$.

On peut alors faire la division euclidienne de $|s|$ par $|u|$. Soit donc $q, r \in \mathbb{N}$ tels que $|s| = q * |u| + r$ avec $0 \leq r < |u|$.

Il existe alors $s' \in \Sigma^*$ tel que $s = u^q \cdot s'$ et $|s'| = r$. De même, il existe $t' \in \Sigma^*$ tel que $t = t' \cdot u^{m-q-1}$. De plus, on a $s' \cdot t' = u$. (cela signifie juste que la décomposition de u^m en $s \cdot t$ tombe au milieu du $(q+1)^{\text{e}}$ u, faites un dessin pour le voir)

Maintenant, réécrivons s et t dans v^m.

On a $v^m = t \cdot s = t' \cdot u^{m-q-1} \cdot u^q \cdot s' = t' \cdot u^{m-1} \cdot s' = t' \cdot u^{m'} \cdot s'$.
En remplaçant $u = s' \cdot t'$ dans cette dernière égalité, on obtient $v^m = t' \cdot (s' \cdot t')^{m'} \cdot s' = t' \cdot s' \cdot (t' \cdot s')^{m'}$. Comme u^m et v^m sont conjugués, on a $|u^m| = |v^m|$, c'est-à-dire $m|u| = m|v|$. Comme $m \neq 0$, on a donc $|u| = |s' \cdot t'| = |v|$.
Comme $v^m = v \cdot v^{m'} = t' \cdot s' \cdot (t' \cdot s')^{m'}$ et $|v| = |s'| + |t'|$, on conclut que $v = t' \cdot s'$ (car v et $t' \cdot s'$ sont deux préfixes de même longueur de v^m) : on vient de démontrer que uRv.

Dans les deux cas, on a démontré uRv.

De 2a et 2b, on conclut que $uRv \Leftrightarrow u^m R v^m$.
D'où le résultat demandé.

3. Soit $u, v \in \Sigma^*$. Montrons que $uRv \Leftrightarrow \exists w \in \Sigma^* : u \cdot w = w \cdot v$.
   (a) Supposons uRv et montrons qu'il existe $w \in \Sigma^*$ tel que $u \cdot w = w \cdot v$.
   Par définition de uRv, il existe $s, t \in \Sigma^*$ tels que $u = s \cdot t$ et $v = t \cdot s$.
   Posons $w = s$. On a alors $u \cdot w = s \cdot t \cdot s = w \cdot v$. D'où l'implication.
   (b) Supposons qu'il existe $w \in \Sigma^*$ tel que $u \cdot w = w \cdot v$ et montrons que uRv.
   On commence par un lemme auxiliaire :
   Lemme 5.1 *Soit $u, v \in \Sigma^*$ et supposons que $u \neq \epsilon$. On suppose qu'il existe $w \in \Sigma^*$ tel que $u \cdot w = w \cdot v$. Alors il existe $w' \in \Sigma^*$ tel que $|w'| < |u|$ et $u \cdot w' = w' \cdot v$.*

   Preuve. On définit le prédicat (sur les entiers) suivant :

$$P(n) \triangleq \exists w \in \Sigma^* : \left\{ \begin{array}{l} |w| \leq n \\ u \cdot w = w \cdot v \end{array} \right. \Rightarrow \exists w' \in \Sigma^* : \left\{ \begin{array}{l} |w'| < |u| \\ u \cdot w' = w' \cdot v \end{array} \right.$$

   On va démontrer que $\forall n \in \mathbb{N} : P(n)$ ce qui permettra de montrer le lemme en utilisant $P(|w|)$ pour w vérifiant l'hypothèse du lemme.
   On raisonne par récurrence sur $n \in \mathbb{N}$.
   – Cas de base : $n = 0$.
   On suppose qu'il existe $w \in \Sigma^*$ tel que $|w| \leq 0$ et $u \cdot w = w \cdot v$.
   Comme $0 \leq |w| \leq 0$, on a $|w| = 0$ (par antisymétrie de $\leq$) et donc $w = \epsilon$.
   Ainsi $u = v$.
   Posons $w' = \epsilon$. On a bien $|w'| < |u|$ car u n'est pas le mot vide. De plus, $u \cdot w' = u = v = w' \cdot v$.
   On a donc montré qu'il existe $w' \in \Sigma^*$ tel que $|w'| < |u|$ et $u \cdot w' = w' \cdot v$.
   – Cas inductif. Soit $n \in \mathbb{N}$ tel que $P(n)$. Montrons $P(n+1)$.
   On suppose qu'il existe $w \in \Sigma^*$ tel que $|w| \leq n+1$ et $u \cdot w = w \cdot v$.
   Deux cas se présentent : ou bien $|w| \leq n$ ou bien $|w| = n+1$.
   – Premier cas : $|w| \leq n$.
   On peut appliquer l'hypothèse de récurrence $P(n)$ et on obtient le résultat.
   – Second cas : $|w| = n+1$.
   Si $|w| < |u|$, on a tout de suite le résultat.
   Sinon, si $|w| \geq |u|$. Par hypothèse, $u \cdot w = w \cdot v$. Comme $|w| \geq |u|$, u est donc un préfixe de w. Il existe donc $w_0 \in \Sigma^*$ tel que $w = u \cdot w_0$. On réécrit dans $u \cdot w = w \cdot v$ et on obtient $u \cdot u \cdot w_0 = u \cdot w_0 \cdot v$. On en déduit que $u \cdot w_0 = w_0 \cdot v$. De plus, comme $|w| = |u \cdot w_0| = |u| + |w_0|$, on a $|w_0| = |w| - |u| < |w| = n+1$. On a donc trouvé w_0 tel que $|w_0| \leq n$ et $u \cdot w_0 = w_0 \cdot v$. On peut donc appliquer l'hypothèse de récurrence $P(n)$ qui nous donne l'existence de $w' \in \Sigma^*$ tel que $|w'| < |u|$ et $u \cdot w' = w' \cdot v$.

Par théorème de récurrence, on conclut. ■

Revenons à la question de l'exercice.

Si $u = \epsilon$, le résultat est trivial car alors $v = \epsilon$.

On suppose donc $u \neq \epsilon$. En utilisant le lemme, on peut supposer que $|w| < |u|$. On a donc $u \cdot w = w \cdot v$ avec $|w| < |u|$. Donc w est un préfixe de u. Il existe donc $w' \in \Sigma^*$ tel que $u = w \cdot w'$. En réécrivant dans l'égalité, on obtient $w \cdot w' \cdot w = w \cdot v$. D'où $v = w' \cdot w$. En posant $s = w$ et $t = w'$, on a donc $u = s \cdot t$ et $v = t \cdot s$, donc uRv.

Dans les deux cas, on a le résultat escompté et donc l'implication.

De 3a et 3b, on déduit l'équivalence demandée.

■