Feuille 5 : Langage et Automates finis

Exercice 1 : Généralités

1. Compter les occurrences des lettres a et b dans les mots suivants : a^3cbbca, $aabgjdd$, $titi$, $babc$.
2. Donner l'ensemble des couples (u, v) tels que $u \cdot v = abaac$.
3. l'intérieur de v : v s'écrit $w_1 \cdot u \cdot w_2$ pour certains mots w_1 et w_2. Un mot u est un *sous-mot* d'un mot v si on peut obtenir u à partir de v par 'effacement' de certaines lettres (pas forcément consécutives) de v. Le nombre d'*occurrences* d'un facteur (resp. sous-mot) u dans le mot v est le nombre de façons de voir u comme facteur (resp. sous-mot) de v.

   Donner le nombre d'occurrences du facteur aba dans le mot $v = ababab$. Donner le nombre d'occurrences du sous-mot aba dans le mÎme mot v.

Exercice 2 : Opérations sur les langages

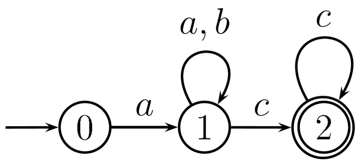
1. Calculer $\mathcal{L} \cdot \mathcal{M}$ pour les ensembles suivants :
   - $\mathcal{L} = \{a, ab, bb\}$ et $\mathcal{M} = \{\varepsilon, b, a^2\}$;
   - $\mathcal{L} = \emptyset$ et $\mathcal{M} = \{a, ba, bb\}$;
   - $\mathcal{L} = \{\varepsilon\}$ et $\mathcal{M} = \{a, ba, bb\}$;
   - $\mathcal{L} = \{aa, ab, ba\}$ et $\mathcal{M} = A^*$.
2. Montrer que le produit est une opération distributive par rapport à l'union, c'est-à-dire que, pour tous langages $\mathcal{L}$, $\mathcal{M}$ et $\mathcal{N}$, on a : $\mathcal{L} \cdot (\mathcal{M} \cup \mathcal{N}) = (\mathcal{L} \cdot \mathcal{M}) \cup (\mathcal{L} \cdot \mathcal{N})$. Montrer que le produit n'est pas distributif par rapport à l'intersection.
3. Parmi les égalités suivantes, lesquelles sont correctes (prouvez ou donnez un contre-exemple) ?
   - $\mathcal{M}^* = \mathcal{M}^* \cdot \mathcal{M}^*$
   - $\mathcal{M}^* = (\mathcal{M} \cdot \mathcal{M})^*$
   - $\mathcal{M}^* = \mathcal{M} \cdot \mathcal{M}^*$
   - $\mathcal{M}^* = (\mathcal{M}^*)^*$
   - $\mathcal{M} \cdot (\mathcal{N} \cdot \mathcal{M})^* = (\mathcal{M} \cdot \mathcal{N})^* \cdot \mathcal{M}$
   - $(\mathcal{M} \cup \mathcal{N})^* = \mathcal{M}^* \cup \mathcal{N}^*$
   - $(\mathcal{M} \cap \mathcal{N})^* = \mathcal{M}^* \cap \mathcal{N}^*$
   - $(\mathcal{M} \cup \mathcal{N})^* = (\mathcal{M}^* \cdot \mathcal{N}^*)^*$
   - $(\mathcal{M} \cup \mathcal{N})^* = (\mathcal{M}^* \cdot \mathcal{N})^* \cdot \mathcal{M}^*$

Exercice 3 : Commutation

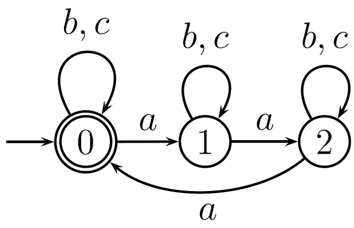
Soient u et v deux mots. On dit que u et v *commutent* si $u \cdot v = v \cdot u$.

Montrer que u et v commutent si et seulement s'il existe un mot w et deux entiers positifs ou nuls m et n tels que $u = w^m$ et $v = w^n$. Pour le sens $\Rightarrow$, on pourra procéder par récurrence sur $|u| + |v|$.

Exercice 4 : Langages reconnus par des automates



(a) Automate $\mathcal{A}_1$



(b) Automate $\mathcal{A}_2$

1. Décrire pour chacun des deux automates les ensembles d'états initiaux/terminaux et la fonction de transition.
2. Les mots abc, $abbbc$ et $abacabcc$ sont-ils reconnus par l'automate $\mathcal{A}_1$? Sont-ils reconnus par l'automate $\mathcal{A}_2$?
3. Décrire les langages reconnus par chacun des automates.

Exercice 5 : Construction d'automates Montrer que les langages suivants sont reconnaissables en donnant pour chaque langage un automate qui le reconnait :
- $\mathcal{L}_1 = \{u \in A^* :$ toute occurrence de b dans u est immédiatement suivie d'au moins deux occurrences de $a\}$,
- $\mathcal{L}_2 = \{u \in A^* : u$ ne contient pas deux a successifs$\}$,
- $\mathcal{L}_3 = \{u \in A^* :$ le nombre d'occurrences de a dans u est pair$\}$,
- $\mathcal{L}_4 = \{u \in A^* :$ les blocs de a dans u sont alternativement de longueur paire et impaire$\}$.

Exercice 6 : Construction d'automates (2) Construire un automate déterministe pour chacun des deux langages suivants :
- le langage $\mathcal{L}_1$ formé des mots contenant le facteur aba ;
- le langage $\mathcal{L}_2$ formé des mots terminant soit par le suffixe aa, soit par le suffixe abb.

Exercice 7 : Digicode

On veut écrire 2 automates déterministes qui reconnaissent l'entrée du “mot de passe” d'un digicode. Il n'y a que des chiffres possibles en entrée. Le code est 11654.

1. Construire un automate qui arrive dans un état final pour toute séquence tapée qui finit par le bon code.
2. Constuire un automate qui lit un code de taille 5, l'accepte si c'est le bon, refuse sinon, et permet ensuite de retenter sa chance.