Alphabets, mots, langages

Exercice 1 (lemme de Levi)
Soit Σ un alphabet.
1. Soient u, v, x, y des mots sur Σ. Montrer que $uv = xy$. si et seulement s'il existe un mot m tel que ($x = um$ et $v = my$) ou ($u = xm$ et $mv = y$). Montrer que m est unique.
2. Soient u et v deux mots sur Σ. Montrer que u et v commutent si et seulement s'il existe un mot w, deux entiers p et q tels que $u = w^p$ et $v = w^q$.

Exercice 2 (lemme d'Arden)
Soit un alphabet Σ.
1. Soient A et B des langages sur Σ. Montrer que A^*B est le plus petit langage au sens de l'inclusion solution de l'équation $L = A.L \cup B$. Montrer que, si $A \neq \emptyset$, cette solution est unique.
2. Soient, pour tout $1 \leq i \leq n$, $1 \leq j \leq n$, $A_{i,j}$ des langages non vides, B_i des langages. Montrer que l'équation $\forall 1 \leq i \leq n,\ L_i = \bigcup_{j=1}^{n} A_{i,j} L_j \cup B_i$ possède une unique solution.
3. Résoudre l'équation $L_1 = \{1\}.L_1 \cup \{0\}L_2 \cup \{\epsilon\}$, $L_2 = \{0\}.L_1 \cup \{1\}L_2$.

Exercice 3
Soient L_1, L_2, L_3 des langages sur un alphabet Σ. Dire parmi les égalités suivantes celle qui sont vraies en vous justifiant :
1. $\bigcup_{i>0} L_1^i = L_1^\star \backslash \{\epsilon\}$.
2. $L_1.L_1^\star = L_1^\star \backslash \{\epsilon\}$.
3. $L_1^\star = \{\epsilon\} \cup L_1.L_1^\star$.
4. $(L_2.L_1)^\star L_2 = L_2(L_1 L_2)^\star$.
5. $(L_1 \cup L_2).L_3 = (L_1.L_3) \cup (L_2.L_3)$.

Exercice 4
Soit Σ un alphabet à deux éléments a et b (ou (,)). On appelle mot de Dyck tout mot m sur Σ qui contient autant de a que de b et tel que tout préfixe de m contient au moins autant de a que de b. On note, pour tout $n \in \mathbb{N}$, C_n le nombre de mots de Dyck de taille $2n$.
1. Déterminer C_0, C_1, C_2, C_3.
2. Montrer que, pour tout $n \in \mathbb{N}$, $C_{n+1} = \sum_{k=0}^{n} C_k C_{n-k}$.
3. On considère f la somme de la série entière $\sum C_n x^n$ et R son rayon de convergence.
a. Montrer que, pour tout $x \in]-R, R[$, $f(x) = 1 + x f(x)^2$.
b. En déduire une expresion de f sur $]-R, R[$.
4. Soit $g : x \mapsto \frac{1-\sqrt{1-4x}}{2x}$.
a. Montrer que g est développable en série entière sur $]-\frac{1}{4}, \frac{1}{4}[$ et que, pour tout $x \in]-\frac{1}{4}, \frac{1}{4}[$, $g(x) = \sum_{n=0}^{+\infty} \frac{1}{n+1}\binom{2n}{n} x^n$.
b. Montrer que, pour tout $n \in \mathbb{N}$, $C_n = \frac{1}{n+1}\binom{2n}{n}$.

Exercice 5
On dit que deux mots u et v sont conjugués s'il existe deux mots x, y tels que $u = xy$ et $v = yx$.
1. Montrer que la relation de conjugaison est une relation d'équivalence sur l'ensemble des mots.
2. Montrer que u et v sont conjugués si et seulement s'il existe un mot w tel que $uw = wv$.

Expressions régulières

Exercice 6
Soit $\Sigma = \{a, b\}$. Trouver des expressions régulières dénotant les langages suivant :
1. L'ensemble des mots.

2. L'ensemble des mots non vides.
3. L'ensemble des mots contenant (au moins) un a.
4. L'ensemble des mots contenant exactement un a.
5. L'ensemble des mots se terminant par b.
6. L'ensemble des mots dont tout a est suivi directement d'un b.
7. L'ensemble des mots dont tout b n'est pas suivi directement d'un a.
8. L'ensemble des mots dont tout b n'a aucun a a sa droite.

Exercice 7
Soient Σ un alphabet et w un mot. Donner des expressions régulières qui dénote les langages suivants :
1. L'ensemble des mots qui contient w comme facteur.
2. L'ensemble des mots qui contient une unique fois w comme facteur.
3. L'ensemble des mots qui contiennent (au moins) deux occurrences disjointes de w.
4. L'ensemble des mots qui contiennent (au moins) deux occurrences non nécessairement disjointes de w.
5. L'ensemble des sous-mots de w.
6. L'ensemble des mots dont w est un sous-mot.

Automates

Exercice 8
Soient $\Sigma = \{a, b\}$ et $A = (Q, i, F, \delta)$ l'automate fini défini par $Q = \{0, 1, 2\}$, $i = \{0\}$, $F = \{1, 2\}$, $\delta(0, a) = \delta(0, b) = 1$, $\delta(1, a) = \delta(1, b) = 2$, $\delta(2, a) = \delta(2, b) = 0$.
1. Dessiner l'automate.
2. Trouver une expression régulière qui dénote le langage reconnu par l'automate.

Exercice 9
Soient $\Sigma = \{a, b\}$ et $A = (Q, I, F, \delta)$ l'automate fini défini par $Q = \{0, 1, 2, 3\}$, $I = \{0, 1\}$, $F = \{2, 3\}$, $\delta(0, a) = \{1, 3\}$, $\delta(0, b) = \emptyset$, $\delta(1, a) = \{0, 2\}$, $\delta(1, b) = \emptyset$, $\delta(2, a) = \{3\}$, $\delta(2, b) = \{1\}$, $\delta(3, a) = \{1\}$, $\delta(3, b) = \emptyset$.
1. Dessiner l'automate.
2. Trouver une expression régulière qui dénote le langage reconnu par l'automate.

Exercice 10
Soient $\Sigma = \{a, b, c\}$ et $A = (Q, I, F, \delta)$ l'automate fini sur Σ définie par $Q = \{0, 1, 2, 3\}$, $I = \{0\}$, $F = \{2, 3\}$, $\delta(0, a) = \{0, 1\}$, $\delta(0, b) = \{0\}$, $\delta(0, c) = \{3\}$, $\delta(1, a) = \{2\}$, $\delta(1, b) = \{2\}$, $\delta(1, c) = \{3\}$, $\delta(2, a) = \delta(2, b) = \emptyset$, $\delta(2, c) = \{3\}$, $\delta(3, a) = \delta(3, b) = \delta(3, c) = \{3\}$.
1. Dessiner l'automate.
2. Trouver une expression régulière qui dénote le langage reconnu par l'automate.

Exercice 11
Eliminer les transitions spontanées de l'automate $A = (Q, I, F, \delta)$ sur $\Sigma = \{a, b, c\}$ défini par $Q = \{1, 2, 3\}$, $I = \{1\}$, $F = \{3\}$, $\delta(1, a) = 1$, $\delta(2, b) = 2$, $\delta(3, c) = 3$, $\delta(1, \epsilon) = 2$, $\delta(2, \epsilon) = 3$.

Exercice 12
Soient $\Sigma = \{a, b\}$ et $A = (Q, I, F, \delta)$ l'automate fini non déterministe défini par $Q = \{0, 1, 2, 3, 4\}$, $I = \{0\}$, $F = \{0, 3\}$, $\delta(0, a) = \{1, 2\}$, $\delta(0, b) = \{3\}$, $\delta(1, a) = \emptyset$, $\delta(1, b) = \{3\}$, $\delta(2, a) = \{4\}$, $\delta(2, b) = \emptyset$, $\delta(3, a) = \delta(3, b) = \emptyset$, $\delta(4, a) = \emptyset$, $\delta(4, b) = \{1\}$, $\delta(3, \epsilon) = \{2\}$, $\delta(2, \epsilon) = \{4\}$.
1. Dessiner cet automate.
2. Représenter un automate fini déterministe reconnaissant le même langage.

Exercice 13
Soit e l'expression régulière $b(ab)^\star|(ba)^\star b$.
1. Trouver un automate reconnaissante le langage dénoté par cette expression.
2. Déterminiser cet automate.

Exercice 14
Soit $\Sigma = \{a, b\}$.
1. Trouver un automate reconnaissant le langage des mots sur Σ comportant un nombre pair de a.

2. Trouver une expression régulière dénotant ce langage.

Exercice 15
Soit Σ un alphabet.
1. Trouver, pour chacun des langages $\{\epsilon\}$, $\emptyset$, $\{a\}$ pour $a \in \Sigma$, un automate fini déterministe le reconnaissant.
2. Soient L_1 et L_2 deux langages reconnus par les automates finis déterministes A_1 et A_2. Trouver un automate fini non déterministe reconnaissante $L_1.L_2$ et $L_1 \cup L_2$. Quelle condition garantit que cet automate soit déterministe ?
3. Trouver un automate non déterministe reconnaissant L_1^*. Est-il déterministe ? Si non, à quelle condition l'est-il ?
4. Donner une nouvelle preuve de l'une des implications du théorème de Kleene.

Exercice 16
Soit Σ un alphabet. Soit $A = (Q, I, F, \delta)$ un automate fini non déterministe. Pour tout $q \in Q$, on note L_q l'ensemble des mots reconnus par l'automate $(Q, \{q\}, F, \delta)$. Pour tout $(q, r) \in Q^2*$, on note $A_{q,r} = \{w \in \Sigma \,/\, r \in \delta(q, w)\}$. Enfin, pour tout $q \in Q$, on pose $B_q = \{\epsilon\}$ si $q \in F$ et $B_q = \emptyset$ si $q \notin F$.
1. Exprimer le langage reconnu par A à l'aide des L_q.
2. Montrer que, pour tout $q \in Q$, $L_q = \bigcup_{q \in Q} A_{q,r}.L_r \cup B_q$.
3. A l'aide du lemme d'Arden, montrer que le langage reconnu par A est régulier.
4. En déduire une implication du théorème de Kleene.

Langages réguliers

Exercice 17
Soient Σ un alphabet et w un mot.Si $w = w_1 \cdots w_n$, on note $w^\top = w_n \cdots w_1$. Soit L un langage. On note $L^\top = \{w^\top \,/\, w \in L\}$, $P(L)$ l'ensemble des préfixes de mots de L, $S(L)$ l'ensemble des suffixes de mots de L, $F(L)$ l'ensemble des facteurs de mots de L.
Montrer que, si L est régulier, $L^\top$, $P(L)$, $S(L)$, $F(L)$ sont réguliers.

Exercice 18
Soit Σ un alphabet.
1. Montrer que l'ensemble des langages réguliers sur Σ est dénombrable.
2. Montrer que $\mathcal{P}(\Sigma^\star)$ n'est pas dénombrable.
3. En déduire que tous les langages ne sont pas réguliers.

Exercice 19
Soit $\Sigma = \{a, b\}$ un langage à deux lettres. Déterminer si les langages suivants sont réguliers :
1. $\{a^n \,/\, n = 0 \mod 2\}$.
2. $\{a^n \,/\, n = 2 \mod 3\}$.
3. $\{a^n b^n \,/\, n \in \mathbb{N}\}$.
4. $\{a^n b a^m \,/\, (n, m) \in \mathbb{N}^2\}$.
5. $\{a^n b a^m \,/\, n < m\}$.
6. $\{a^n b a^m \,/\, n \leq m\}$.
7. $\{a^p \,/\, p \text{ premier}\}$.
8. $\{a^n b^m \,/\, n < m\}$.
9. $\{w \in \Sigma^* \,/\, |w|_a < |w|_b\}$.
10. $\{a^n b^m \,/\, n + m \mod 2\}$.
11. $\{a^n b^m \,/\, n = m \mod 3\}$.
12. L'ensemble des palindromes.

Implémentation en OCaml

Exercice 20
Dans tout l'exercice, on cherchera à représenter une lettre par son numéro d'apparition dans l'ordre lexico-graphique.
1. La fonction `int_of_char` associe à une lettre son code ASCII.
a. Ecrire une fonction `entier_vers_lettre : int -> char` qui associe à un entier i la i-ème lettre de l'alphabet.
b. Ecrire une fonction `lettre_vers_entier` réciproque de la précédente.
On considèrera un alphabet Σ de m lettres constitué des m premières lettres de l'alphabet. On représentera un

automate fini déterministe complet à n états sur Σ par un enregistrement de trois champs : un champ `initial` qui est un entier entre 0 et $n-1$ représentant l'état initial, un champ finals qui est une liste d'entiers entre 0 et $n-1$, un champ `transitions` qui est une matrice $n \times m$ tel que `transitions.(i).(j)` est la transition depuis `i` étiquetée par `j`.

On déclare donc le type suivant :

```
type afd = {initial : int ; finals : int list ; trans : int array array }
```

1. On suppose dans cette question que $\Sigma = \{a, b, c, d, e\}$ et on identifie une lettre de l'alphabet à sa place dans l'ordre usuel des lettres.
a. Représenter l'automate défini par

```
let a= { initial=0 ; finals=[2] ;
transitions = [| [| 1 ; 0 ; 0 ; 4 ; 4 |] ; [| 2 ; 4 ; 4 ; 1 ; 4 |] ;
[| 3 ; 2 ; 2 ; 4 ; 4 |] ; [| 0 ; 4 ; 4 ; 4 ; 3 |] ; [| 4 ; 4 ; 4 ; 4 ; 4 |] |]}
```

b. Définir via le type précédent l'automate de l'exercice 8.
2. Définir une fonction `delta : afd -> int -> char -> int` prenant en argument un automate fini déterministe, un état q, une lettre l renvoyant $\delta(q, l)$.
3. Ecrire une fonction `delta_etoile : afd -> int -> string -> int` prenant en argument un automate fini déterministe, un état q, un mot w renvoyant $\delta^\star(q, w)$.
4. Ecrire une fonction `reconnait : afd -> string -> bool` prenant en argument un automate A, un mot w et renvoyant vrai si $w \in L(A)$, faux sinon.
Pour représenter un automate fini non déterministe sans transitions spontanées, on utilisera un enregistrement de trois champs : un champ `initiaux` qui sera la liste des états initiaux, un champ `finals` qui sera la liste des états finals, `transitions` qui sera une matrice de taille $n \times m$ telle que `transitions.(i).(j)` sera la liste des états de $\delta(i, j)$.

On déclaire donc le type suivant :

```
type afnd = {initiaux : int list ; finals : int list ; transitions : int list array array }
```

5. On suppose dans cette question que $\Sigma = \{a, b, c\}$ et on utilise la même identifications qu'à la question 1.
a. Représenter l'automate défini par

```
let a={ initiaux = [ 0 ; 1] ; finals = [ 1 ; 2] ;
transitions = [| [| [] ; [1;2] ; [0] |]; [| []; [1]; [0;1] |] ;
[| [] ; [0;1;2] ; [0] |] ; [| [] ; [1;2;3] ; [3] |] |] }
```

b. Définir un objet de type afnd représentant l'automate de l'exercice 10.
6. Ecrire une fonction `est_deterministe : afnd -> bool` prenant en argument un automate fini non déterministe indiquant si l'automate représente un automate fini déterministe.
7. Ecrire une fonction `enleve_doublons : 'a list -> 'a list` prenant en argument une liste qui renvoie une liste ayant le même ensemble d'éléments que la liste en argument et telle que chaque élément n'apparaisse qu'au plus une fois.
8. Ecrire une fonction `delta : afnd -> int -> char -> int` prenant en argument un automate fini non déterministe, un état et une lettre qui renvoie la liste sans doublons des états atteints depuis l'état en argument par la lettre en argument.
9. Ecrire une fonction `delta_ens : afnd -> int list -> char -> int` prenant en argument un automate fini non déterministe, un ensemble d'états et une lettre qui renvoie la liste sans doublons des états atteints depuis les états en argument par la lettre en argument.
10. Ecrire une fonction `delta_etoile : afnd -> int list -> string -> int` prenant en argument un automate fini non déterministe, un état et un mot qui renvoie la liste sans doublons des états atteints depuis l'état en argument par le mot en argument.
11. Ecrire une fonction `reconnait : afnd -> string -> bool` prenant en argument un automate fini non déterministe, un mot renvoyant vrai si le mot est reconnu par l'automate, faux sinon.
12. Ecrire une fonction `determinise : afnd -> afd` prenant en argument un automate fini non déterministe renvoyant un automate fini déterministe reconnaissant le même langage.