TD1 - Langages formels

Exercice 1

Question 1. Sur l'alphabet $\Sigma = \{a, b, c\}$, on considère les langages $L = \{\varepsilon, a, bc\}$ et $L' = \{abc, cb\}$. Pour chacun des mots ci-dessous, discuter son appartenance ou non aux langages $L \cdot L'$, $L^* \cdot L'$ et $(L \cdot L')^*$.

- □ **1.1.** abc
- □ **1.2.** $aacb$
- □ **1.3.** $abccb$
- □ **1.4.** cba

Question 2. Soient L et L' deux langages sur un alphabet Σ. Vérifier que l'on a toujours : $(L' \cdot L)^* \cdot L' = L' \cdot (L \cdot L')^*$.

Question 3. Soit L un langage sur un alphabet Σ. Discuter la véracité des affirmations suivantes.

- □ **3.1.** $(L^+ = L^*) \iff (\varepsilon \in L)$
- □ **3.2.** $L^2 \cdot L^+ = L^+$
- □ **3.3.** $\varnothing \cdot L = L$
- □ **3.4.** $L^* \cdot L^* = L^*$
- □ **3.5.** $(L = L^*) \implies (L = \{\varepsilon\})$
- □ **3.6.** $\varnothing^* = \varnothing$
- □ **3.7.** $\varnothing^+ = \varnothing$
- □ **3.8.** $L^+ \cdot L^+ = L^+$
- □ **3.9.** $L^+ \cdot L^* = L^+$
- □ **3.10.** $\{\varepsilon\}^* = \{\varepsilon\}$

Question 4. Sur $\mathcal{P}(\Sigma^*)$, montrer que l'opération de concaténation $\cdot$ distribue $\cup$ mais ne distribue pas $\cap$.

Question 5. Soit Σ un alphabet et $w = a_1 \dots a_n$ un mot sur Σ. On appelle *mot miroir* de w le mot $a_n \dots a_1$ noté w^{R}. On désigne par L^{R} l'ensemble des mots miroirs des mots du langage L. Peut-on affirmer que tout mot de $L \cdot L^{\mathrm{R}}$ est un palindrome ?

Question 6. Soit L un langage. Est-il vrai que $L = L^{\mathrm{R}}$ si et seulement si L ne contient que des palindromes ?

Exercice 2

Sur l'alphabet $\Sigma = \{a, b\}$, les mots de Fibonacci sont définis par $f_0 = \varepsilon$, $f_1 = a$, $f_2 = b$ et :

$$\forall n \in \mathbb{N}^* \quad f_{n+2} = f_{n+1} \cdot f_n$$

Question 1. Montrer que, pour tout entier $n \geqslant 3$, le suffixe de longueur 2 de f_n est ab si n est pair, ba si n est impair.

Question 2. Pour tout entier $n \geqslant 3$, on désigne par g_n le préfixe de f_n obtenu en supprimant les deux dernières lettres de ce mot. Montrer que g_n est un palindrome.

Exercice 3

On considère les mots construits sur l'alphabet $\Sigma = \{a, b\}$. Les *mots de Dyck* définissent l'ensemble $\mathcal{D}$ des expressions bien parenthésées sur Σ par les règles de construction suivantes : $\varepsilon \in \mathcal{D}$ et $(r, s) \in \mathcal{D}^2 \Rightarrow arbs \in \mathcal{D}$.
La *valuation* d'un mot de Σ^* est définie par le morphisme additif $\sigma\colon \Sigma^* \to \mathbb{Z}$ tel que $\sigma(a) = 1$ et $\sigma(b) = -1$. On montre qu'un mot m de Σ^* appartient à $\mathcal{D}$ si et seulement si $\sigma(m) = 0$ et pour tout préfixe m' de m, $\sigma(m') \geqslant 0$.
En pratique, les lettres a et b sont remplacées respectivement par les parenthèses ouvrante (et fermante). Une expression mathématique syntaxiquement correcte telle que $1 + ((2 - x) - 5 \times (y - 3)) \times (x - y)$ est une *expression bien parenthésée* ; il lui correspond le mot de Dyck $(()())()$ ou encore $aababbab$. Le mot $()($ est mal parenthésé et n'appartient pas à $\mathcal{D}$.
Dans la suite, l'alphabet $\Sigma = \{a, b\}$ est représenté par le type `alphabet` et les mots sont représentés par le type `words`.

```
type alphabet = A | B
type words = alphabet list
```

Question 1. Écrire une fonction `dyck : words -> bool` qui détermine si un mot appartient à $\mathcal{D}$ ou pas.

Question 2. Une *factorisation* d'un mot de $\mathcal{D}$ est un découpage de ce dernier en produit d'expressions bien parenthésées, chacune étant appelée un *facteur* de ce mot. Par exemple, $aababb = (()())$ est composée d'un seul facteur, $abaababb = ()(()())$ est composée de deux facteurs et $aabbabaababb = (())()(()())$ est composée de trois facteurs. Écrire une fonction `nb_fact : words -> int` qui calcule le nombre de facteurs d'un mot de Dyck.

Question 3. En déduire une fonction `affiche_fact : words -> unit` qui affiche cette factorisation.

Exercice 4

Une feuille de papier rectangulaire est pliée n fois dans le sens vertical, en repliant à chaque fois la moitié droite sur la moitié gauche. Une fois la feuille dépliée, ses plis forment une suite de creux et de bosses.

Question 1. Combien y a-t-il de plis à l'étape n ?

Question 2. Chaque étape du pliage est représentée par un mot. Un creux est codé par un 0 et une bosse par un 1. On pose $w_0 = \varepsilon$, mot vide.
□ **2.1.** Déterminer w_1, w_2 et w_3.
□ **2.2.** Montrer que pour tout entier naturel i, w_i est toujours préfixe de w_{i+1}.

Question 3.
□ **3.1.** Pour tout entier naturel non nul n, proposer un algorithme de construction de w_n à partir de w_{n-1}.
□ **3.2.** Écrire une fonction de calcul de w_n.

Question 4.
□ **4.1.** Écrire une fonction qui renvoie la représentation binaire d'un entier naturel n, poids fort en tête, et un entier donnant le nombre de bits.
□ **4.2.** Les mots de la suite de pliage étant préfixes les uns des autres, on peut considérer le *mot infini* w dont ils sont tous préfixes. Proposer un algorithme qui prend en entrée la représentation binaire d'un entier naturel n, son nombre de bits et qui renvoie le n-ième bit de w.

Exercice 5

Un mot sur un alphabet Σ contient un *facteur carré* s'il peut s'écrire sous la forme rs^2t où r, s, t sont des mots de Σ^* avec $|s| \geqslant 1$. s^2 est le facteur carré. L'objet de cet exercice est l'étude des mots sans facteurs carrés.

Question 1. Si Σ ne contient que deux lettres, montrer que tout mot d'au moins quatre lettres possède un facteur carré.

Question 2. On considère l'alphabet $\Sigma = \{a, b\}$ et le morphisme σ défini par $\sigma(a) = ab$ et $\sigma(b) = ba$. Montrer que pour tout entier naturel non nul n, $\sigma^{n-1}(a)$ est préfixe de $\sigma^n(a)$.

Question 3. Dans la suite, m désigne le mot de longueur infinie, appelé *mot de Thue-Morse*, tel que pour tout entier naturel n, $\sigma^n(a)$ est préfixe de m. On pose $\Sigma_1 = \{ab, ba\}$.
□ **3.1.** Montrer que si $s \in \Sigma_1^*$, alors asa et bsb n'appartiennent pas à Σ_1^*.
□ **3.2.** Montrer que pour tout entier naturel n non nul, $\sigma^n(a) \in \Sigma_1^*$.
□ **3.3.** En déduire que m ne possède pas de facteur de la forme r^2x où r est un mot et x est la première lettre du mot r.

Question 4. Soit à présent le mot infini μ formé du nombre de b compris entre deux a consécutifs de m.
□ **4.1.** Montrer que l'alphabet $\{0, 1, 2\}$ suffit pour l'écrire et que μ ne possède pas de facteurs carrés.
□ **4.2.** Avec le type `type alphabet = A | B`, écrire une fonction `gen : int -> unit` qui permet de le générer.
On rappelle que la fonction `iter` du module `List` permet d'itérer sur une liste.

```
val iter : ('a -> unit) -> 'a list -> unit
  List.iter f [a1; ...; an] applies function f in turn to a1; ...; an.
  It is equivalent to begin f a1; f a2; ...; f an; () end.
```

Exercice 6

Question 1. Soit x, y, u, v quatre mots sur un alphabet Σ tels que $xy = uv$. Montrer qu'il existe un unique mot $t \in \Sigma^*$ tel que l'une des deux conditions suivantes soit réalisée :

- $u = xt$ et $y = tv$;
- $x = ut$ et $v = ty$.

Illustrer les situations décrites par les deux points précédents avant de faire la démonstration
Ce résultat de combinatoire sur les mots, appelé *lemme de Lévi*, est utilisé dans les questions suivantes pour établir deux propriétés liées à la non-commutativité de la concaténation.

Question 2. Soit x, y, z trois mots de Σ^* tels que $xy = yz$ et $x \neq \varepsilon$. Montrer qu'il existe deux mots u et v de Σ^* et un entier naturel k tels que :

$$x = uv \qquad y = \begin{cases} (uv)^k u \\ u(vu)^k \end{cases} \qquad z = vu$$

Question 3. Soit x, y deux mots de Σ^* tels que $xy = yx$ avec $x \neq \varepsilon$ et $y \neq \varepsilon$. Montrer qu'il existe un mot $u \in \Sigma^*$ et deux entiers naturels i et j tels que $x = u^i$ et $y = u^j$.