TP 12 : Langages réguliers

Ce TP est la suite du TP précédent, continuez à la suite de ce dernier.

I. Expressions régulières en OCaml

I. A. Implémentation

Question 1. Déclarez le type récursif `exp_reg`, qui permet de construire des expressions régulières, dont les symboles seront de type `char`. Vous utiliserez les constructeurs `Vide`, `Epsilon`, `Symbole`, `Union`, `Concat` et `Etoile`, dont vous déterminerez vous-même l'arité et les données si nécessaire.

Question 2. En exemple, construisez l'expression régulière $e = ab^* \mid ba^*$, en considérant a et b comme des symboles de type `char`.

Question 3. Écrire une fonction récursive `appartient : string → exp_reg → bool`, telle que l'appel `appartient u e` renvoie `true` si le mot `u` peut être engendré par l'expression régulière `e` (c'est-à-dire si `u` appartient au langage dénoté par `e`), et `false` sinon. *Indication : On peut utiliser un filtrage par motif sur l'expression régulière* `e`. *Les cas de la concaténation et de l'étoile seront traités de manière naïve.*

I. B. Application : Recherche de motifs[1]

On s'intéresse dans cette partie à la construction d'expression régulières qui filtrent certains motifs. On souhaite tester tous les mots d'une liste donnée avec ces expressions régulières. Nous allons utiliser le fichier `dinosaures.ml`, vous pouvez en recopier le contenu dans votre fichier. Vous trouverez dans le fichier :

- l'objet `list_of_dinosaurs` de type `string list` ;
- l'objet `alphabet`, de type `exp_reg`. Il s'agit d'une expression régulière qui dénote le langage $\{\texttt{A}, \ldots, \texttt{Z}, \texttt{a}, \ldots, \texttt{z}\}$.

Question 4. Écrire une expression régulière de type `exp_reg` exprimant tous les noms finissant par « raptor » . Utiliser cette expression pour trouver tous les noms de raptor dans la liste.

Question 5. Écrire une expression régulière de type `exp_reg` exprimant tous les noms finissant par « saurus » et commençant par un « A » majuscule. Quels mots de la liste respectent ces critères ?

Question 6. Écrire une expression régulière de type `exp_reg` exprimant tous les noms commençant par « Mega » ou finissant par « saurus ». Quels mots de la liste respectent ces critères ?

II. TD

II. A. Langages réguliers

Exercice 7. Soit Σ un alphabet. Montrer que les langages suivants sont réguliers.

- $\{\varepsilon\}$;
- Σ (interprété comme le langage des mots d'une seule lettre) ;
- Le langage des mots de longueur paire ;
- Le langage des mots de longueur impaire ;
- Le langage des mots qui contient le facteur aba.

Exercice 8 : *Langages finis*. Soit Σ un alphabet. Montrer que tout langage fini sur Σ est régulier.

Exercice 9 : *Facteurs*. Soit Σ un alphabet et u un mot non vide sur Σ. Montrer que les langages suivants sont réguliers :

1. Le langage $F_1(u)$ des mots contenant au moins une occurrence du facteur u.
2. Le langage $F_2(u)$ des mots contenant au moins deux occurrences disjointes du facteur u.
3. Le langage $F_2'(u)$ des mots contenant au moins deux occurrences non-disjointes du facteur u, c'est-à-dire deux occurrences qui « se chevauchent ». Par exemple, le mot $ababa$ contient deux occurences non-disjointes du facteur aba.

[1] Adaptation de l'exercice disponible ici : https://www.normalesup.org/~doulcier/teaching/python/Ex03_regexp_dino.html

II. B. Expressions régulières

Exercice 10. Donner une description par une phrase des langages correspondant aux expressions régulières suivantes :

$$\Sigma\Sigma \quad ; \quad (\varepsilon \mid \Sigma)(\varepsilon \mid \Sigma) \quad ; \quad (\Sigma\Sigma)^* \quad ; \quad \Sigma^* a \Sigma^* \quad ; \quad \Sigma^* ab \Sigma^* \quad ; \quad \Sigma^* a \Sigma^* b \Sigma^* \quad ; \quad (ab)^*$$

Exercice 11. Soit $\Sigma = \{a, b\}$. Donner tous les mots de longueur au plus 4 des langages engendrés par les expressions régulières suivantes :
1. $(a \mid ba)^*$
2. $a \mid ba^*$
3. $a^*(b \mid a)b*$
4. $a(aa \mid b(ab)^*a)^*a$

Exercice 12. Soit $\Sigma = \{a, b, c, d\}$. Donner des expressions régulières décrivant :
1. L'ensemble des mots non-vides sur Σ.
2. L'ensemble des mots non-vides commençant par b et terminant par d sur Σ.
3. L'ensemble des mots comportant exactement deux b, où chaque a est suivi par un c, et qui se terminent par b.

Soit maintenant $\Sigma = \{a, b\}$. Donner des expressions régulières décrivant :
4. L'ensemble des mots qui ont toujours un nombre pair de b entre deux occurrences de a.
5. L'ensemble des mots tels que toutes les occurrences de a apparaissent avant toutes les occurrences de b.

Exercice 13. Démontrer le théorème suivant : « Un langage L est régulier si et seulement s'il existe une expression régulière e telle que $L = \mathcal{L}(e)$. »

Exercice 14 : *Complémentaire*. Pour chacune de ces expressions régulières sur l'alphabet $\Sigma = \{a, b\}$, donner une expression régulière reconnaissant son langage complémentaire :
1. $(a \mid b)^*b$
2. $((a \mid b)(a \mid b))^*$
3. $(b \mid ab^*a)^*$
4. $(b \mid ab)^*$

Exercice 15 : *Mots binaires*. Soit $\Sigma = \{0, 1\}$. Un mot sur Σ est di *mot binaire*. Si un mot est soit le mot 0, soit un mot commençant par un 1, il est dit *normalisé*. On construit également l'application *valeur* $V : \Sigma^* \to \mathbb{N}$ comme suit :
- $V(0) = 0$ et $V(1) = 1$
- Pour tout $n > 1$ et $u = u_{n-1}...u_0 \in \Sigma^*$, $V(u) = \sum_{i=0}^{n-1} V(i)2^i$.

Donner une caractérisation de chacun des langages suivants, puis montrer qu'ils sont réguliers en exhibant une expression régulière qui les engendrent.
1. L_1, le langages des mots binaires normalisés.
2. L_2, le langages des mots binaires dont la valeur est paire.
3. L_3, le langages des mots binaires normalisés dont la valeur est paire.
4. L_4, le langages des mots binaires dont la valeur est une puissance de 2.