

TD1-Analyse lexicale

Chaque exercice consiste à réaliser un programme exécutable `exo.o` en utilisant le compilateur d'analyseur lexical flex.

Langages rationnels

Exercice 1.1

Reconnaître les mots sur l'alphabet $\{a, b, c\}$ qui ont au moins un facteur abb .

Exercice 1.2

1- Reconnaître les mots sur l'alphabet $\{a, b, c\}$ qui ont au moins un facteur dans l'ensemble $\{cabb, cbab, cbb, cbbba\}$.

2- Donner la liste des couples (position, numero du facteur) dans le mot d'entrée.

Exemple :

Sur l'entrée `cabbbcbabcccacbbba`, le programme devra retourner $(0,0) (5,1) (13,3)$

*3- Reprendre la question 2 pour l' ensemble de facteurs $\{cabbb, bbc\}$.

Sur l'entrée `cabbbcabbbbcabbbc`, le programme devra retourner

$(0,0) (3,1) (5,0) (9,1) (11,0) (14,1)$

Exercice 1.3

Reconnaître les mots de $\{a, b\}^*$ possédant un nombre pair de a et un nombre impair de b.
Solution 1 : On définira deux variables globales entières (des "compteurs") que l'on mettra à jour au fur et à mesure de la lecture du mot.

Solution 2 : On écrira en flex un automate fini reconnaissant ce langage (on utilisera les "start conditions" de flex).

Exercice 1.4

1-Reconnaître l'ensemble L des mots sur l'alphabet $\{a, b, c\}$ qui comportent le même nombre de a de b et de c.

2-Ce langage formel L est-il rationnel? algébrique?

Exercice 1.5

Décomposer un mot en produit de mots appartenant aux langages

$$R_1 := (aa)^+b, \quad R_2 := (ab)^+b, \quad R_3 := \{a, b\}.$$

Chaque mot a-t-il au moins une décomposition? exactement une décomposition?

On demande au programme d'imprimer en sortie, pour w , une décomposition $D(w)$ "glou-tonne" au sens :

Pour $w \neq \varepsilon$, soit w_0 le plus long préfixe de w qui appartient à l'un des 3 langages R_i et s le suffixe tel que $w = w_0 \cdot s$; alors $D(w) := [w_0] \cdot D(s)$.

Pour $w = \varepsilon$, $D(w) := \varepsilon$.

Par exemple, sur l'entrée `aaaaabababbbbaab`, le programme devra retourner

`[a] [aaaab] [ababb] [b] [b] [aab]`

Fonctions rationnelles

Exercice 1.6

Le programme doit, pour tout mot d'entrée sur l'alphabet $\{a, b, c\}$, imprimer en sortie le mot obtenu en effectuant la substitution de lettres :

$$a \mapsto b, b \mapsto c, c \mapsto a.$$

Exercice 1.7

Le programme doit, pour tout mot d'entrée w sur l'alphabet $\{a, b, c\}$, imprimer en sortie le mot $f(w)$ suivant : soit

$$D(w) = v_0, v_1, \dots, v_{n-1},$$

la décomposition définie à l'exercice 5 ; sur chaque facteur $v_i \in R_1$, on applique la substitution $a \mapsto b, b \mapsto c$, sur chaque facteur $v_i \in R_2$, on applique la substitution $a \mapsto c, b \mapsto d$, sur chaque facteur $v_i \in R_3$, on ne change rien.

Par exemple, sur l'entrée `aaaaabababbbbaab`, le programme devra retourner
`[a] [bbbbbc] [cdcd] [b] [b] [bbc]`