Université Bordeaux 1
Licence, module Compilation, 2017/2018

TD1-Analyse lexicale

Théorie

Exercice 1.T1

Soit F un ensemble fini de mots sur l'alphabet $\{a, b, c\}$. Soit L l'ensemble des mots de $\{a, b, c\}^*$ qui ont au moins un facteur dans F. Montrer que L est rationnel.

Exercice 1.T2

Soit L l'ensemble des mots de $\{a, b\}^*$ possédant un nombre pair de a et un nombre impair de b. Construire un automate fini déterministe reconnaissant ce langage.

Exercice 1.T3

On considère l'ensemble L des mots sur l'alphabet $\{a, b, c\}$ qui comportent le même nombre de a de b et de c.
Ce langage formel L est-il rationnel ? algébrique ?

Exercice 1.T4

On considère les langages

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

On appelle décomposition d'un mot $w \in \{a, b, c\}$ sur (R_1, R_2, R_3) tout t-uple de mots $(w_0, \ldots, w_i, \ldots w_n)$ tel que :

$$w = w_1 \cdots w_i \cdot w_{i+1} \cdots w_n, \quad \text{et } \forall i \in [1, n],\ w_i \in R_1 \cup R_2 \cup R_3$$

(pour $w = \varepsilon$ la suite vide, de longueur $n = 0$, est une décomposition).
1- Chaque mot de $\{a, b, c\}^*$ a-t-il *au moins une* décomposition sur ces langages ? *exactement une* décomposition ?

Une décomposition est dite *gloutonne* ssi pour tout $i \in [1, n]$, w_i est le *plus long* préfixe de $w_i \cdot w_{i+1} \cdots w_n$ qui appartient à $R_1 \cup R_2 \cup R_3$.
2- Montrer que tout mot $w \in \{a, b, c\}^*$ a une décomposition gloutonne. Montrer que, pour chaque mot w, cette décomposition gloutonne est unique.

Exercice 1.T5

Donner un automate fini, déterministe, *avec sorties*, qui calcule l'homomorphisme de monoides libres,$\{a, b, c\}^* \to \{a, b, c\}^*$ défini par :

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

Exercice 1.T6

À tout mot w sur l'alphabet $\{a, b\}$, on associe le mot $f(w)$ suivant : soit

$$D(w) = (v_1, v_2, \ldots, v_n),$$

la décomposition (gloutonne) définie à l'exercice 1.T4 ; 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,

$$f(aaaaabababbbbaab) = a \cdot bbbbc \cdot cdcdd \cdot b \cdot b \cdot bbc$$

1- Soient $u, v \in \{a, b\}^*$. Est-il toujours vrai que $f(u)$ est préfixe de $f(uv)$?
2- Peut-on calculer f avec un automate fini déterministe (sur les entrées) et avec sorties ?
3- Soit $\#$ une nouvelle lettre (i.e. qui n'appartient pas à $\{a, b, \}$) et soit $g : \{a, b\}^*\# \to \{a, b, c, d, \#\}^*$ l'application définie par

$$g(w \cdot \#) := f(w) \cdot \#.$$

Peut-on calculer g avec un automate fini déterministe (sur les entrées) et avec sorties ?
Aide : remarquer que $g(a^{2n}\#) = a^{2n}\#$ et $g(a^{2n}b\#) = b^{2n}c\#$.

Programmes FLEX

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

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

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

Exercice 1.5

Décomposer un mot en produit de mots appartenant aux langages

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

(voir l'exercice 1.T4 ci-dessus). On demande au programme d'imprimer en sortie, pour w, la décomposition "gloutonne" de w, où les facteurs successifs sont entourés de crochets. Par exemple, sur l'entrée `aaaaabababbbbaab`, le programme devra retourner
`[a][aaaab][ababb][b][b][aab]`

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

$$D(w) = v_1, v_2, \ldots, v_n,$$

la décomposition définie à l'exercice 1.T4 ;
- 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.
- de plus les images des facteurs successifs sont entourées de crochets.

Par exemple, sur l'entrée `aaaaababbbbbaab`, le programme devra retourner
`[a][bbbbc][cdcdd][b][b][bbc]`