

Module : Théorie des langages et compilation

Enseignant(s) : Équipe TLA

Classe(s) : 3A1...3A30 & 4SE

Documents non autorisés ; Calculatrice non autorisée ; Internet non autorisée

Date : --/03-04/2024 ; Heure --h-- ; Durée : 1h ; Nombre de pages : 2

Exercice 1 : (7,5 pts)

On considère l'automate fini A défini par $A = \langle \Sigma, S, e_0, \delta, F \rangle$ avec :

- $\Sigma = \{a, b\}$: l'alphabet
- $S = \{1, 2, 3, 4, 5, 6, 7\}$: l'ensemble des états
- $e_0 = 1$: l'état initial
- $F = \{7\}$: l'ensemble des états finaux
- $\delta: S \times \Sigma \rightarrow S$: la fonction de transition définie par la table ci-contre :

	a	b
1	2	-
2	3	-
3	4,5	7
4	4,5	7
5	-	6
6	4,5	7
7	-	-

1. Donner une représentation graphique de l'automate A . (1 pt)
2. Donner l'expression régulière E_A dénotant le langage $L(A)$. (2 pts)
3. L'automate A est-il déterministe? Justifier. Si non, tracer l'automate déterministe A_D équivalent à A . (3 pts)
4. Donner l'expression régulière E_{A_D} dénotant le langage $L(A_D)$. (0,5 pt)
5. Donner l'automate B , tel que $L(B) \cap L(A_D) = \emptyset$ et $L(B) \cup L(A_D) = \Sigma^*$ sur $\Sigma = \{a, b\}$. (1 pt)

Exercice 2 : (6 pts)

Soit E_R , l'expression régulière suivante sur l'alphabet $\Sigma = \{a, b, c\}$: $a^*(bc^+ | ca^*)$

1. En appliquant l'algorithme de *Thompson*, donner l'automate reconnaissant le langage décrit par l'expression E_R . (1,5 pts)
2. Construire, en précisant toutes les étapes, l'automate minimal équivalent. (4,5 pts)

Exercice 3 : (6,5 pts)

1. Donner les expressions régulières correspondantes aux langages suivants :
 - a. Le langage **L₁** de tous les mots **w** construits sur **{a, b}** dont la taille maximale est égale à deux . (0,5 pt)
 - b. Le langage **L₂** de tous les mots **w** construits sur **{a, b}** contenant la séquence **ab** mais pas la séquence **ba**. (1,5 pt)
 - c. Le langage **L₃** de tous les mots **w** construits sur **{a, b}** contenant au plus l'une des deux séquences **ab** ou **ba**. (1 pt)
2. Compléter le fichier de spécification **Flex** suivant pour construire un analyseur lexical reconnaissant les langages **L₁**, **L₂** et **L₃**, permettant de retourner sur la console, à chaque identification d'un lexème, la chaîne reconnue ainsi que la description correspondante. (1,5 pt)

```
.....  
.....  
%%  
{.....} printf(.....) ;  
{.....} printf(.....) ;  
{.....} printf(.....) ;  
{.....} printf("Autre") ;  
%%  
int yywrap(){return 1;}  
main()  
{  
  yylex();  
}
```

3. Donner le résultat d'exécution de l'analyseur lexical sur les instructions suivantes : (2 pts)
 - a. **bbaaabb**
 - b. **abbbbbaaaa**
 - c. **bbababa**
 - d. **aaabaa**

Bon Travail