

Session : 1

EPREUVE : Langages Formels et Compilation

Durée : 2 h 00 – (documents papiers - sauf livres - autorisés ; appareils électroniques interdits)

Les exercices sont indépendants. Le barème est donné à titre indicatif.

Exercice 1 – 3.5 points

Soit l'automate à pile suivant : $A_1 = (\{q_0, q_1, q_2\}, \{a, b, c\}, \{z_0, a, b, c\}, \delta, q_0, z_0, \{q_2\})$ avec δ définie par :

- | | | | |
|-----|--------------------------------------|------|--|
| (1) | $\delta(q_0, a, z_0) = (q_1, z_0a)$ | (8) | $\delta(q_1, b, c) = (q_1, c)$ |
| (2) | $\delta(q_0, b, z_0) = (q_1, z_0)$ | (9) | $\delta(q_1, b, z_0) = (q_1, z_0)$ |
| (3) | $\delta(q_0, c, z_0) = (q_1, z_0c)$ | (10) | $\delta(q_1, c, a) = (q_1, \lambda)$ |
| (4) | $\delta(q_1, a, a) = (q_1, aa)$ | (11) | $\delta(q_1, c, c) = (q_1, cc)$ |
| (5) | $\delta(q_1, a, c) = (q_1, \lambda)$ | (12) | $\delta(q_1, c, z_0) = (q_1, z_0c)$ |
| (6) | $\delta(q_1, a, z_0) = (q_1, z_0a)$ | (13) | $\delta(q_1, \lambda, z_0) = (q_2, z_0)$ |
| (7) | $\delta(q_1, b, a) = (q_1, a)$ | | |

1. Donnez toutes les étapes de l'analyse des mots $accba$ et $accbabcb$ par A_1 . N'oubliez pas de conclure et de justifier vos conclusions.
2. Décrivez les mots acceptés par cet automate.

Exercice 2 – 2.5 points

Soit la grammaire $G_2 = (\{S, B, W, X\}, \{a, b, c\}, S, \{S \rightarrow abc, S \rightarrow aSBc, cB \rightarrow WB, WB \rightarrow WX, WX \rightarrow BX, BX \rightarrow Bc, bB \rightarrow bb\})$

1. De quel type est G_2 ? Justifiez votre réponse.
2. Donnez deux mots du langage engendré par G_2 .

Exercice 3 – 3.5 points

Soit l'automate à mémoire linéairement bornée $A_3 = (\{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}\}, \{a, b, c, <, >\}, \{a, b, c, X, Y, <, >\}, \delta, q_0, \{q_8\})$ avec δ définie par :

- | | | | |
|------|--------------------------------|------|--------------------------------------|
| (1) | $\delta(q_0, <) = (q_1, <, D)$ | (13) | $\delta(q_4, X) = (q_6, X, D)$ |
| (2) | $\delta(q_1, a) = (q_2, X, D)$ | (14) | $\delta(q_5, a) = (q_5, a, G)$ |
| (3) | $\delta(q_1, b) = (q_9, X, D)$ | (15) | $\delta(q_5, b) = (q_5, b, G)$ |
| (4) | $\delta(q_2, a) = (q_2, a, D)$ | (16) | $\delta(q_5, X) = (q_1, X, D)$ |
| (5) | $\delta(q_2, b) = (q_2, b, D)$ | (17) | $\delta(q_6, c) = (q_7, c, D)$ |
| (6) | $\delta(q_2, c) = (q_3, c, D)$ | (18) | $\delta(q_7, Y) = (q_7, Y, D)$ |
| (7) | $\delta(q_3, Y) = (q_3, Y, D)$ | (19) | $\delta(q_7, >) = (q_8, >, G)$ |
| (8) | $\delta(q_3, a) = (q_4, Y, G)$ | (20) | $\delta(q_9, a) = (q_9, a, D)$ |
| (9) | $\delta(q_4, Y) = (q_4, Y, G)$ | (21) | $\delta(q_9, b) = (q_9, b, D)$ |
| (10) | $\delta(q_4, c) = (q_4, c, G)$ | (22) | $\delta(q_9, c) = (q_{10}, c, D)$ |
| (11) | $\delta(q_4, a) = (q_5, a, G)$ | (23) | $\delta(q_{10}, Y) = (q_{10}, Y, D)$ |
| (12) | $\delta(q_4, b) = (q_5, b, G)$ | (24) | $\delta(q_{10}, b) = (q_4, Y, G)$ |

1. Donnez les 5 premières étapes de l'analyse du mot $abcab$ (application de 5 transitions).
2. Quand passe-t-on dans l'état q_2 ?
3. Quel but cherche-t-on à atteindre lorsqu'on est dans cet état (soyez précis(e)) ?

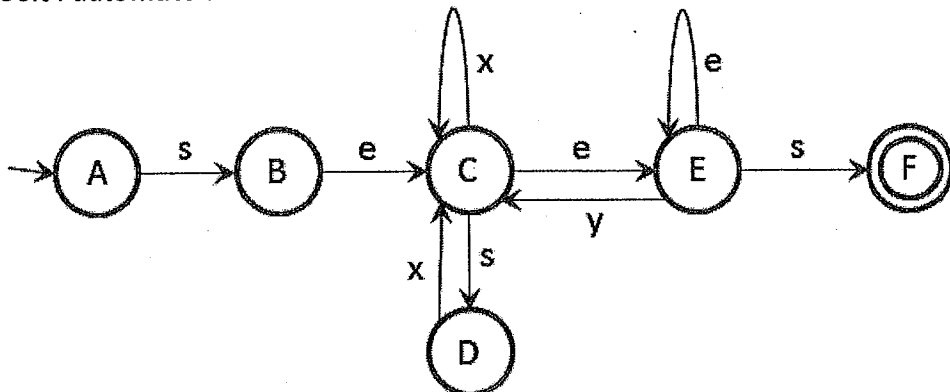
Exercice 4 – 6 points

Soit la grammaire $G_4 = (\{S, X, Y\}, \{a, b, c\}, S, \{S \rightarrow abb \mid aXbb, X \rightarrow aXbb \mid Y \mid \lambda, Y \rightarrow c \mid cY\})$

1. Transformez G_4 de manière à obtenir une grammaire équivalente sous forme normale de Chomsky. Soit G_4' la grammaire obtenue.
2. A partir de G_4' , construisez la pyramide de Cocke-Kasami-younger pour le mot $aacbbb$.
3. D'après la pyramide obtenue, quels sous-mots de $aacbbb$ appartiennent au langage engendré par G_4 ?

Exercice 5 – 4.5 points

Soit l'automate fini ci-dessous.



1. Trouvez l'expression régulière associée à cet automate par la méthode des définitions régulières vue en cours.

Cet automate reconnaît les commentaires de la forme `/* commentaire */` dans les programmes C.

Pour que l'expression régulière soit plus pratique à trouver, on a remplacé

* par e

/ par s

« n'importe quel caractère sauf * et / » par x

« n'importe quel caractère sauf / » par y

2. On veut générer un analyseur lexical qui supprime¹ les commentaires d'un programme C (uniquement ceux qui ont la forme indiquée ci-dessus). Ecrivez le programme lex qui va générer cet analyseur.

Remarque : L'analyseur voulu peut être obtenu avec un programme lex de 2 lignes. Plus le programme lex que vous proposez est long et comporte des parties inutiles, plus vous perdez des points.

¹ Supprimer les commentaires revient à afficher tout le programme sauf les commentaires.