UNIVERSITE DE BOURGOGNE — Année 2022-2023

U.F.R. Sciences et Techniques — 22 juin 2023

Filière : Licence 3 Informatique

Session : 2

EPREUVE : Langages Formels et Compilation

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

Les 4 exercices sont indépendants. Le barème est donné à titre indicatif.

Exercice 1 – 4,5 points

Soit la grammaire G_1=({S}, {a, b}, S, {S→Sab|ab})

1. Transformez cette grammaire de manière à obtenir une grammaire équivalente sous forme normale de Chomsky. Soit G_1' la grammaire obtenue.
2. Utilisez G1' pour effectuer une analyse du mot abababab par l'algorithme de Coke-Kasami-Younger (inutile d'écrire toutes les explications). N'oubliez pas de conclure l'analyse et de justifier votre conclusion.
3. Quels autres mots ont été reconnus ?

Exercice 2 – 4 points

Soit l'automate à pile suivant :

AAP_2 = ({q}, {a, c, i, t, e}, {S, X, Y, Z}, t, q, S, ∅),

avec t définie par :

(1) t(q, S, a) = (q, λ)
(2) t(q, S, i) = (q, XSZY)
(3) t(q, S, i) = (q, SZY)
(4) t(q, S, a) = (q, S)
(5) t(q, S, i) = (q, SXSZY)
(6) t(q, S, i) = (q, SSZY)
(7) t(q, X, e) = (q, S)
(8) t(q, Y, c) = (q, λ)
(9) t(q, Z, t) = (q, λ)

1. Cet automate est-il déterministe ou non ? Justifiez votre réponse.
2. Effectuez l'analyse du mot ictictaea par cet automate (simulez le fonctionnement de l'automate). N'oubliez pas de conclure et de justifier votre conclusion.

Exercice 3 – 3,5 points

Soit la grammaire suivante :

G_3 = ({S, A, B, C, X}, {a, b, c}, S, {S → aSBc|aBc, cB→Bc, aB→abb, bB→bbb})

1. De quel type est cette grammaire ? Justifiez votre réponse.
2. Donnez 3 mots engendrés par cette grammaire, et les dérivations permettant de les obtenir.
3. Quel est le langage reconnu ?

Exercice 4 – 8 points (à rendre sur une feuille à part)

(note : la question 5 est indépendante des résultats des 4 premières)

On veut décoder et stocker des valeurs de couleurs. Une couleur est définie par 3 entiers séparés par des virgules. Par exemple : 123,456,789. Comme plusieurs couleurs peuvent être listées, on utilise un # à la fin de chaque ligne, un fichier plus compliqué de couleurs peut être :

123,456,789#

1,23,45,67,89,88,77,66,55#

1,23,45,67,89,88,77,66,55,14,22,46,62,83,81,74,66,51,13,21,41,61,84,58,27,16,35#

Dans la 2e ligne, 3 couleurs sont décrites 1,23,45 puis 67,89,88 puis 77,66,55. Un seul # est possible par ligne, chaque valeur est définie sur 3 chiffres au maximum, avec 0 significatif : 4567 est une erreur, 021 également (la question 5 nécessitera d'autres contraintes sur ces valeurs).

1. construire un programme Lex qui permette de détecter **chaque** valeur, chaque virgule, chaque #. Utilisez des définitions régulières, affichez les erreurs,
2. construire le programme Yacc qui permette de vérifier un fichier de couleurs, à partir d'une grammaire que vous écrirez,
3. on veut stocker les valeurs entières pour chaque couleur dans un tableau d'entiers, modifier le programme Yacc et ses actions, afficher toutes les valeurs du tableau en fin d'analyse,

   Exemple de mots et sorties correspondantes :

   123,456,789#

   1,23,45,67,89,88,77,66,55#

   1,23,45,67,89,88,77,66,55,14,22,46,62,83,81,74,66,51,13,21,41,61,84,58,27,16,35#

   Donne :

   77 66 55 67 89 88 1 23 45 77 66 55 67 89 88 1 23 45 77 66 55 67 89 88 1 23 45 77 66 55 67 89 88 1 23 45 123 456 789

4. compter le nombre de lignes de couleurs, et dans chaque ligne le nombre de couleurs. Afficher un message d'erreur si une ligne ne contient pas autant de couleurs que la précédente (sauf la 1ère),
5. en réalité, une valeur doit être un entier, compris entre 0 et 255. Donner l'automate permettant de valider une valeur de couleur, ainsi que l'expression régulière correspondante.