

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 – 6 points

Soit la grammaire $G_1 = (\{S, X, Y, Z\}, \{a, b, c\}, S, \{S \rightarrow XaYaX, X \rightarrow \lambda | bX, Y \rightarrow aYa | bbX | Z, Z \rightarrow ZZc | c\})$

1. Trouvez une grammaire équivalente à G_1 sous forme normale de Greibach. Donnez toutes les étapes de la transformation.
2. Décrivez les mots engendrés par cette grammaire.

Exercice 2 – 4 points

Proposez un automate à pile permettant de reconnaître les mots de longueur supérieure ou égale à 1, comportant uniquement des a et des b, pouvant commencer par a ou b, pouvant terminer par a ou b, formés de séries de a et de b de longueur paire dans n'importe quel ordre.

Exemples de mots corrects : aabbbbbaaaabbbb, aabb, aa, bb, bbbbaabbaaaaaa, aabbaabbaabbaabbaa

Exemples de mots incorrects : abab, baba, bba, bab, baabbb

Exercice 3 – 4 points

Soit la grammaire $G_3 = (\{S, X, Y\}, \{a, b, c\}, S, \{S \rightarrow XY | XA, A \rightarrow a, B \rightarrow b, X \rightarrow AB, Y \rightarrow AS\})$

1. A partir de G_3 , construisez la pyramide de Coke-Kasami-Younger pour le mot abaaba.
2. D'après la pyramide obtenue, quels sous-mots de abaaba appartiennent au langage engendré par G_3 ?

Exercice 4 – 6 points (à rendre sur une feuille à part)

1) Donnez l'expression régulière représentant les mots de longueur supérieure ou égale à 1 comportant n'importe quel caractère sauf #, & et \n, mais pouvant comporter la sous-chaîne \#, et ne pouvant commencer ni par un chiffre ni par \

2) Écrivez les programmes Lex et Yacc qui permettent de calculer le nombre de lettres total des mots reconnus. Dans le fichier d'entrée on ne doit trouver que des mots correspondant à l'expression de la question 1, séparés par des \n. Il ne peut y avoir plusieurs \n à la suite. Le fichier ne termine pas forcément par un \n.

Exemples :

abcd\#efgh\n abcd1234 \nreconnu\n → fichier valide, 3 mots reconnus, $10+10+7 = 27$ lettres

abcd&efgh... → fichier invalide, caractère interdit &

1nonreconnu\n... → fichier invalide, configuration interdite chiffre en 1ère position d'un mot

mot correct \n\n autre... → fichier invalide, deux \n à la suite

a) Pour la partie Lex, donnez la description régulière des mots reconnus, la ou les règles Lex, ainsi que les unités lexicales, l'envoi des informations nécessaires au programme Yacc. Le programme n'effectue pas de calcul, la somme des nombres de caractères se fera dans les règles de Yacc ;

b) Pour la partie Yacc, donnez le ou les *token*, le symbole de départ, les règles de grammaire et les expressions du calcul du nombre total de caractères.