

Module : **Théorie des Langages et Techniques de Compilation** Classe(s) :3 info A et 4INFOB

Documents autorisés : NON

Nombre de pages : 2

Date : MAI 2014

Durée :1H30

Exercice 1 Grammaires Automates à pile (6 pts)

Considérons le langage $L = \{w \in \{x,y,z\}^* \mid w = x^n y^{3k} z^{3n} \text{ avec } n > 0 \text{ et } k \geq 0\}$

1. Donner une grammaire hors contexte permettant d'engendrer le langage L. (2 pts)
2. Donner une dérivation la plus à gauche du mot $w = xyxyz$. (1 pt)
3. Construire un automate à pile permettant de reconnaître le langage L. (2 pts)
4. Donner une trace d'exécution de l'automate permettant d'accepter le mot $w = xyxyz$ (1 pt)

Exercice 2 Analyseur lexical (3 pts)

On veut concevoir un vérificateur qui vérifie que des chaînes saisies peuvent correspondre à une adresse IP (IPv4).

On vérifie donc que la chaîne saisie commence par un caractère '[' ensuite 1 ou 2 ou 3 chiffres ensuite le caractère '.' ensuite 1 ou 2 ou 3 chiffres ensuite le caractère '.' ensuite 1 ou 2 ou 3 chiffres ensuite le caractère '.' ensuite 1 ou 2 ou 3 chiffres ensuite le caractère ']'

Exemple de mots : [192.168.32.12] [193.163.2.1], on supposera que l'adresse [999.999.999.999] est valide même si elle est techniquement impossible.

1. Donner une définition régulière ou une expression régulière spécifiant le langage des adresses IP (1,5 pts)
2. Construire un automate à état fini déterministe reconnaissant ce langage. (1,5pts)

Exercice 3 Ambiguïté, élimination récursivité à gauche et factorisation (4 pts)

Soit la grammaire $G(V_N, V_T, R, A)$: avec A est l'axiome $V_N = \{S, A, B\}$, $V_T = \{a, b\}$, et R défini par:

$S \rightarrow SAB \mid BAS \mid BSA \mid B \mid ba$

$A \rightarrow abBA \mid abA \mid a$

$B \rightarrow bB \mid BbA \mid ba$

1. Donner une dérivation pour le mot baababa. Que pouvez-vous conclure ? (1pt)
2. Donner la grammaire G' obtenue en éliminant la récursivité gauche de G ? (1,5)
3. Donner la grammaire G'' obtenue en factorisant G' (1,5)

Exercice 4 Analyse Syntaxique (7 pts)

Soit la grammaire $G(V_N, V_T, R, S)$:

avec $V_N = \{S, C, C', D, D', E\}$, $V_T = \{\text{while, do, if, then, else id, =, or, and, bid, (,), ;}\}$

S est l'axiome et R défini par:

$S \rightarrow \text{while } C \text{ do } S ;$

$S \rightarrow \text{if } C \text{ then } S \text{ else } S ;$

$S \rightarrow \text{id} = \text{id}$

$C \rightarrow D C'$

$C' \rightarrow \text{or } D C' \mid \epsilon$

$D \rightarrow E D'$

$D' \rightarrow \text{and } E D' \mid \epsilon$

$E \rightarrow \text{bid}$

$E \rightarrow (E)$

1. Calculer les ensembles Premier et Suivant pour tous les non terminaux de G (3 pts)
2. Construire la table de l'analyseur prédictif non récursif (table d'analyse LL(1)) de la grammaire G . (2 pts)
3. Donnez les étapes d'analyse de la chaîne **if bid then id=id else id=id;** en montrant à chaque pas, le contenu de la pile, la partie non encore lue de la chaîne et la sortie générée. (1 pt)
4. En déduire la dérivation gauche associée (1 pt)