

Contrôle continu - Compilation (L3 Info)
Durée : 2h - documents interdits

Conventions

- Axiome : symbole non terminal de la partie gauche de la première production de la grammaire
- Symboles non terminaux : lettres *MAJUSCULES ITALIQUES*
- Symboles terminaux : lettres **minuscules true-type** ou caractères spéciaux simples

Exercice 1 - Vrai ou faux (4 pts)

Pour chacune des affirmations suivantes dites si elle est vraie ou fausse en justifiant brièvement et de façon convaincante votre réponse (seules les réponses accompagnées d'une justification seront prises en compte).

1. Les grammaires hors-contexte ambiguës ne peuvent pas être reconnues par un automate à pile.
2. La grammaire $A \rightarrow XY, X \rightarrow a|\varepsilon, Y \rightarrow a$ n'est pas ambiguë.
3. Certains langages hors-contextes peuvent être reconnus par un automate à pile déterministe.
4. Seules les grammaires récursives peuvent générer des langages infinis.

Exercice 2 - Écriture de grammaire (6 pts)

On définit le langage des expressions régulières sur un alphabet Σ de la manière suivante :

- \emptyset est une expression régulière
- ε est une expression régulière
- x (tel que $x \in \Sigma$) est une expression régulière
- Si P et Q sont des expressions régulières alors :
 - $P + Q$ est une expression régulière
 - $P.Q$ est une expression régulière
 - P^* est une expression régulière
 - (P) est une expression régulière
- rien d'autre n'est une expression régulière.

1. Ecrire une grammaire G permettant de générer les expressions régulières sur l'alphabet $\Sigma = \{a, b\}$.
2. Dessiner un arbre de dérivation de l'expression $(a + b.a)^*$
3. Est ce que G est ambiguë ? justifiez votre réponse
4. On définit l'ordre de priorité suivant sur les opération régulière : $*$ est prioritaire sur $.$ qui est prioritaire sur $+$. Ecrire une grammaire non ambiguë G' qui prend en compte ces priorités et dessiner l'arbre de dérivation de l'expression $(a + b.a)^*$

Exercice 3 - Analyse $LL(1)$ (6 pts)

Soit la grammaire G_1 suivante :

$$\begin{array}{l} 1 \quad N \rightarrow E [L] \\ 2 \quad L \rightarrow N L \\ 3 \quad L \rightarrow \varepsilon \\ 4 \quad E \rightarrow a \\ 5 \quad E \rightarrow b \\ 6 \quad E \rightarrow c \end{array}$$

1. Décrire le langage généré par G_1 en une phrase ou deux et donner deux mots appartenant à ce langage.
2. Calculer les ensembles PREMIER et SUIVANT des symboles de G_1
3. Construire la table $LL(1)$ de G_1
4. Simuler les 10 premières étapes de l'analyse $LL(1)$ du mot $a[b[]c[a[]]]$. Les configurations sont représentées par le triplet (w, α, y) , où w est la partie de la bande de lecture qui commence au caractère sous la tête de lecture (ce qui reste à analyser), α est la pile et y est la séquence de symboles de sortie (numéros des productions appliquées).

Exercice 4 - Récursivité à gauche (4 pts)

Soit la grammaire G_2 :

$$\begin{array}{l} A \rightarrow A a \mid B a \mid a \\ B \rightarrow A b \mid B b \mid \varepsilon \\ C \rightarrow C c \mid c \end{array}$$

1. Appliquer l'algorithme d'élimination de la récursivité gauche indirecte pour produire une grammaire G'_2 non récursive à gauche telle que $L(G_2) = L(G'_2)$, en respectant l'ordre suivant sur les symboles non terminaux : $A < B < C$.