

A.3 Transformation de grammaires

1. Soit l'alphabet $X = \{+, =, a\}$. (1) Donner une grammaire algébrique pour le langage L dont chaque mot représente une addition correcte de deux suites de caractères a . Par exemple L contient le mot $aa + aaaa = aaaaaa$. (2) Donner un automate à pile qui reconnaît le même langage.
2. Soit le langage L_1 sur le vocabulaire $V = \{l', \text{homme}, \text{ours}, \text{qui}, a, \text{vu}\}$ formé de l'ensemble des phrases finies de la forme *l'homme qui a vu l'ours*, *l'homme qui a vu l'homme qui a vu l'ours*, *l'homme qui a vu l'ours*, *l'homme qui a vu l'homme qui a vu l'homme qui a vu l'ours* ... *qui a vu l'ours*.
 - (a) Donner une grammaire algébrique (*context-free*) engendrant L_1 .
 - (b) Donner une grammaire régulière engendrant L_1 .

Soit L_2 le langage engendré par la grammaire \mathcal{G}_2 :

S	→	NP Rel
NP	→	l'homme
		l'ours
Rel	→	qui a vu NP Rel
		qui a vu l'ours

- (c) Mettre la grammaire \mathcal{G}_2 sous forme normale de Chomsky.
 - (d) Proposer une grammaire \mathcal{G}_3 qui engendre le langage L_3 , sur-ensemble de L_2 mais dans lequel les symboles *ours* et *homme* sont **strictement** interchangeables.
 - (e) Décrire informellement les différences entre les langages L_1 et L_2 .
 - (f) Donner une grammaire algébrique du langage $L_2 \setminus L_1$.
3. Mettre sous forme normale de Chomsky la grammaire définie par les règles de production suivantes

$$S \rightarrow AB \mid aS \mid a$$

$$A \rightarrow Ab \mid \varepsilon$$

$$B \rightarrow AS$$
4. (a) Montrer que la grammaire suivante est ambiguë :

$$S \rightarrow TU; T \rightarrow ST \mid a; U \rightarrow US \mid b$$

5. Grammaires propres :

- (a) Définir l'algorithme qui rend une grammaire ε -libre ; Appliquer l'algorithme à la grammaire $S \rightarrow aSbS \mid bSaS \mid \varepsilon$.
 - (b) Définir un algorithme qui détermine la liste des symboles ayant une contribution ; Utiliser l'algorithme pour simplifier (nettoyer) la grammaire

$$S \rightarrow A \mid B$$

$$A \rightarrow aB \mid bS \mid b$$

$$B \rightarrow AB \mid Ba$$

$$C \rightarrow AS \mid b$$
 - (c) Définir un algorithme qui recherche les symboles inaccessibles ; Utiliser l'algorithme pour simplifier (encore) la grammaire précédente.
 - (d) Proposer un algorithme qui supprime d'une grammaire toutes les productions singulières
6. Eliminer la récursivité gauche de la grammaire ETF.