Grammaires (2)

Exercice 1 (Applications du lemme d'Ogden)
En utilisant le lemme d'Ogden énoncé en cours, montrer les assertions suivantes.
- $L_1 = \{a^n b^n c^n \mid n \geq 0\}$ est un langage non algébrique
- $L_2 = \{a^n b^n c^p d^p \mid n \geq 0,\ p > 0\}$ est un langage algébrique non linéaire
- $L_3 = \{a^n b^n c^m \mid n > 0,\ m \geq 0\} \cup \{a^q b^p c^p \mid p > 0,\ q \geq 0\}$ est un langage algébrique ambigu

---

Exercice 2 (Problèmes indécidables)
Soit L un langage algébrique. Montrer que les problèmes suivants sont indécidables :
- $L = \Sigma^*$?
- L est-il ambigu ?

Exercice 3 (Grammaires réduites)
Soit $G = \langle \Sigma, V, P, S \rangle$ une grammaire algébrique.

1. Montrer que l'on peut :
   - Calculer l'ensemble des variables productives de G ;
   - Décider si $\mathcal{L}_G(S) = \emptyset$;
   - Calculer l'ensemble des variables accessibles de G.
2. On suppose que $\mathcal{L}_G(S) \neq \emptyset$. Montrer que l'on peut effectivement construire une grammaire réduite équivalente à G.

Exemple Réduire la grammaire suivante :

$$\begin{array}{ll} S \longrightarrow aT + bTU + abTS + UV & U \longrightarrow aU + bU \\ T \longrightarrow aU + bT + a & V \longrightarrow aT + bS + a \end{array}$$

Exercice 4 (Grammaires propres)
Soit $G = \langle \Sigma, V, P, S \rangle$ une grammaire algébrique.

1. Montrer que l'on peut calculer l'ensemble des variables x telles que $\varepsilon \in \mathcal{L}_G(x)$.
2. Montrer que l'on peut effectivement construire une grammaire propre G' qui engendre $\mathcal{L}_G(S) \setminus \{\varepsilon\}$.
3. Montrer que l'on peut décider si un mot $u \in \Sigma^*$ est engendré par G.

Exemple Rendre propre la grammaire suivante :

$$\begin{array}{ll} S \longrightarrow TU + VW + X & V \longrightarrow U + W + XaV \\ T \longrightarrow TT + W & W \longrightarrow cW + \varepsilon \\ U \longrightarrow aU + V & X \longrightarrow W + b \end{array}$$

Exercice 5 (Forme normale de Chomsky)
Soit $G = \langle \Sigma, V, P, S\rangle$ une grammaire.

1. Montrer que l'on peut construire une grammaire équivalente G' en forme normale de Chomsky faible (resp. forte).
2. Montrer que l'on peut décider si $\mathcal{L}_G(S)$ est fini.

Exemple Mettre la grammaire suivante sous forme normale de Chomsky forte :

$$\begin{aligned} S &\longrightarrow bA + aB \\ A &\longrightarrow bAA + aS + a \\ B &\longrightarrow aBB + bS + b \end{aligned}$$

Exercice 6 (Forme normale de Greibach)
Soit $G = (\Sigma, V, P, S)$ une grammaire algébrique de langage associé non vide.

– G est en *forme normale presque Greibach* si toutes les productions de P sont de la forme $X \longrightarrow a\alpha$ avec $a \in \Sigma$ et $\alpha \in (V \cup \Sigma)^*$,
– G est en *forme normale de Greibach* si toutes les productions de P sont de la forme $X \longrightarrow a\alpha$ avec $a \in \Sigma$ et $\alpha \in V^*$.

1. Soit G une grammaire algébrique, $X \longrightarrow \alpha_1 Y \alpha_2$ une règle de P et soit
$$Y \longrightarrow \beta_1 + \cdots + \beta_m$$
l'ensemble des règles de P ayant Y comme membre gauche. Montrer que la grammaire G' obtenue en remplaçant la règle $X \longrightarrow \alpha_1 Y \alpha_2$ par les règles $X \longrightarrow \alpha_1 \beta_i \alpha_2$ (pour $1 \leq i \leq m$) engendre le même langage que G.
2. Soit G une grammaire algébrique propre, $X \longrightarrow X\alpha_1 + \ldots X\alpha_r$ l'ensemble des règles de P ayant X comme membre gauche et X comme symbole le plus à gauche dans les membres droits, $X \longrightarrow \beta_1 + \cdots + \beta_s$ le reste des règles de P ayant X comme membre gauche. Montrer que la grammaire G' obtenue en ajoutant une variable Z et en remplaçant l'ensemble des règles ayant X comme membre gauche par les règles
$$X \longrightarrow \beta_1 Z + \ldots \beta_s Z + \beta_1 + \cdots + \beta_s$$
et
$$Z \longrightarrow \alpha_1 Z + \ldots \alpha_r Z + \alpha_1 + \ldots \alpha_r$$
engendre le même langage que G.
3. On considère une grammaire algébrique $G = (\Sigma, V, P, S)$, propre et réduite. On suppose que $V = \{X_1 \ldots X_n\}$ et $S = X_1$. Montrer que l'on peut construire une suite $(G_i)_{0 \leq i \leq n}$ de grammaires engendrant le même langage que G telles que $G_0 = G$ et pour tout $0 \leq i \leq n$, G_i a pour ensemble de non-terminaux $V_i = V \cup \{Z_1 \ldots Z_i\}$ et vérifie que pour toute règle $X_k \longrightarrow \gamma$,
– γ ne commence pas par un non-terminal de $\{X_1 \ldots X_k\} \cup \{Z_1 \ldots Z_i\}$ si $k \leq i$
– γ ne commence pas par un non-terminal de $\{X_1 \ldots X_i\} \cup \{Z_1 \ldots Z_i\}$ si $k > i$
En déduire qu'on peut obtenir à partir de G_n une grammaire algébrique sous forme normale de Greibach engendrant le même langage que G.