Théorie des Langages – Feuille n° 2

Langages et Grammaires

Exercice 1 Soient les langages L_1, L_2, L_3 et L_4 suivants construits sur l'alphabet $\Sigma = \{a, b, c\}$. Montrer que ces 4 langages sont tous deux à deux différents.

— $L_1 = (a + b)^*ca^*b^*$
— $L_2 = a^*b^*c(a + b)^*$
— $L_3 = (a^* + b^*)ca^*b^*$
— $L_4 = \{(a + b)^n ca^m b^n,\ n, m \in \mathbb{N}\}$

Exercice 2 - Soient les langages L_1, L_2 et L_3 construits sur l'alphabet $X = \{a, b\}$. On rappelle que $(a + b) = \{a\} \cup \{b\}$.

$$L_1 = \{a^n b(a + b)^n, n \in \mathbb{N}\}$$
$$L_2 = \{(a + b)^n ba^n, n \in \mathbb{N}\}$$
$$L_3 = \{(a + b)^n b(a + b)^n, n \in \mathbb{N}\}$$

1. Montrez que les langages L_1, L_2 et L_3 ne sont pas égaux
2. Soit $L_4 = \{(a + b)^m ba^n, m, n \in \mathbb{N}\}$. Montrez que $L_2 \neq L_4$
3. Donnez les grammaires qui engendrent L_2 et L_4

Exercice 3 - Soit la grammaire $G = \langle V, \Sigma, P, S\rangle$, avec $V = \{a, b, S\}$, $\Sigma = \{a, b\}$ et $P = \{S \to aSa; S \to bSb; S \to \epsilon\}$.

1. Soit $G' = \langle V, \Sigma, P', S\rangle$, avec $P' = P \cup \{S \to SS\}$. Montrez que $aabaab \in \mathcal{L}(G')$. Montrez ensuite que G' est ambigüe.
2. Quel est le langage engendré par G ? Démontrez
3. Pourquoi G n'est pas ambigüe ?

Exercice 4 - Soit la grammaire $G = \langle V, \Sigma, P, A\rangle$, avec $V = \{a, b, A, B\}$, $\Sigma = \{a, b\}$ et $P = \{A \to aA|bB; B \to b|bB\}$.

1. De quel type est la grammaire G ?
2. Construisez l'arbre de dérivation de profondeur 4 de G.
3. A l'aide des règles de production et des mots dérivés de l'arbre de dérivation, déduisez quel est le langage généré par G (écriture en compréhension) ?

Exercice 5 - Soit la grammaire $G = \langle V, \Sigma, P, S\rangle$, avec $V = \{\text{if}, \text{then}, \text{else}, a, b, S\}$, $\Sigma = \{\text{if}, \text{then}, \text{else}, a, b\}$ et $P = \{S \to \text{if } b \text{ then } S \text{ else } S\, ;\ S \to \text{if } b \text{ then } S\, ;\ S \to a\}$.

1. Démontrez que cette grammaire est ambigüe
2. Proposez une solution pour lever l'ambiguïté

Exercice 6 - Donner, pour chacun des langages suivants, une grammaire qui l'engendre.

1. $L_1 = \{0^{2n} \mid n \geq 0\}$
2. $L_2 = \{0^n 1^n \mid n \geq 0\}$
3. $L_3 = \{a^n b^{2n} \mid n \geq 0\}$
4. $L_4 = \{a^m b^n a^n b^m \mid n, m \geq 1\}$
5. $L_5 = \{w \in \{a, b\}^+ \mid \ |w|$ est un multiple de 3 $\}$
6. $L_6 = \{0^i 1^j \mid i \neq j, n, m \geq 0\}$

Exercice 7 - Ecrire une grammaire permettant de générer les identificateurs d'un langage de programmation (comme Python ou C). On considérera qu'un identificateur est valide s'il commence par une lettre alphabétique (uniquement minuscule) qui peut éventuellement être suivie d'une ou plusieurs lettres alphabétiques (minuscule ou majuscule) et/ou chiffres.

Exercice 8

1. Ecrire une grammaire hors-contexte G_{HC} permettant de décrire tous les mots d'un langage sur $\Sigma = \{a, b\}$ qui contiennent les sous-chaînes aa ET bb.
2. En vous basant sur la grammaire hors-contexte définie à la question précédente, construisez une grammaire régulière G_R équivalente.