

SÉRIE D'EXERCICES n° 1

EXERCICE 1 : Soit la grammaire $G = (\{a, b, c\}, \{S, A\}, P, S)$

où P contient les règles suivantes :

$$S \rightarrow aS \mid bA$$

$$A \rightarrow cA \mid \varepsilon$$

- 1) Déterminer si les mots $w_1 = abac$, $w_2 = aabccc$, $w_3 = cabbac$ et $w_4 = ab$ sont dans $L(G)$.
- 2) Trouver le langage généré par G (qu'on note $L(G)$).

EXERCICE 2 : Trouver pour chacune des grammaires $G_i = (\{a, b, c\}, \{S, A, R, T\}, P_i, S)$ le langage engendré par celle-ci : ($i=1, \dots, 5$)

1) $P_1 : S \rightarrow aS \mid aA ; A \rightarrow bAc \mid \varepsilon$

2) $P_2 : S \rightarrow aSc \mid A$
 $A \rightarrow bAc \mid \varepsilon$

3) $P_3 : S \rightarrow aSbS \mid \varepsilon$

4) $P_4 : S \rightarrow aSc \mid bA$
 $A \rightarrow bA \mid \varepsilon$

5) $P_5 : S \rightarrow aRbc \mid abc$
 $R \rightarrow aRTb \mid aTb ; Tb \rightarrow bT ; Tc \rightarrow cc$

EXERCICE 3 : Quel est le type de la grammaire $G_i = (\{a, b\}, \{S, A\}, P_i, S)$

1) $P_1 : S \rightarrow aAS \mid a$
 $A \rightarrow SbA \mid SS \mid ba$

2) $P_2 : S \rightarrow aAS \mid SA$
 $aA \rightarrow a$

EXERCICE 4 : Pour chacun des langages suivants, donner une grammaire qui l'engendre :

- | | |
|-----------------------------------------------------------|-------------------------------------------------------|
| a) $L_1 = \{0^{2n} / n \geq 0\}$ | f) $L_6 = \{a^m b^n a^n b^m / n \geq 1, m \geq 1\}$ |
| b) $L_2 = \{0^n 1^n / n \geq 0\}$ | g) $L_7 = \{w \in \{a, b\}^+ / w \equiv 0[3]\}$ |
| c) $L_3 = \{a^n b^{2n} / n \geq 0\}$ | h) $L_8 = \{0^i 1^j / i \geq j \geq 0\}$ |
| d) $L_4 = \{a^n b^m / n \leq m \leq 2 \times n\}$ | i) $L_9 = \{0^i 1^j / i \neq j, i \geq 0, j \geq 0\}$ |
| e) $L_5 = \{0^n w w^R 1^n / n \geq 0, w \in \{a, b\}^*\}$ | j) $L_{10} = \{a^{2^n} / n \geq 0\}$ |

EXERCICE 5 : Soit L un langage de type $i \in \{0, 1, 2, 3\}$. Est-il possible qu'un langage $L' \subset L$ ne soit pas de type i ?

indication : on sait que le langage $\{0^n 1^n / n \geq 0\}$ n'est pas régulier.

EXERCICE 6 : Soit le langage L défini comme suit :

$$L = \{ a^{2n} b c^{2m+1} / n, m \geq 0 \}.$$

- 1) Montrer que L est de type 3 en trouvant une grammaire de type 3 qui l'engendre.
- 2) Trouver une grammaire de type 2, et qui n'est pas de type 3, qui engendre L.

EXERCICE 7 : Soit le langage L défini comme suit :

L = ensemble de tous les mots de $\{0, 1\}^*$ qui contiennent un nombre pair de « 1 ».

Mêmes questions, pour L, que l'exercice 6.

EXERCICE 8 : Soit la grammaire G dont les règles de production sont :

$$S \rightarrow aAb \mid \varepsilon$$

$$A \rightarrow aSb$$

$$Ab \rightarrow \varepsilon$$

- 1) Déterminer $L(G)$.
- 2) Construire une grammaire de type 2 équivalente à G.

EXERCICE 9 : Soit la grammaire G dont les règles de production sont :

$$S \rightarrow RD$$

$$R \rightarrow aRb \mid A$$

$$Ab \rightarrow bbA$$

$$AD \rightarrow \varepsilon$$

Mêmes questions, pour G, que l'exercice 8.

EXERCICE 10 : Soit l'alphabet terminal $\pi = \{a, (,), +, *\}$.

- 1) Soit L le langage des expressions arithmétiques construites sur l'alphabet π .
Trouver une grammaire, de type 2, pour L.
- 2) Soit L' le langage des expressions arithmétiques défini sur π tel que l'on ne peut pas avoir des parenthèses imbriquées directement l'une dans l'autre. À titre d'exemple $(a^*(a+a))$ ou (a) sont dans L' mais $(a^*((a+a)))$ ou $((((a))))$ ne le sont pas. Trouver une grammaire, de type 2, pour L'.

EXERCICE 11 : Ecrire une grammaire pour générer les identificateurs d'un langage comme Pascal. On considérera qu'un identificateur est valide s'il commence par une lettre alphabétique (majuscule ou minuscule) qui peut éventuellement être suivie d'une ou plusieurs lettres alphabétiques et/ou chiffres. Pour ce qui est des non terminaux de cette grammaire on pourra utiliser par exemple <Id1>, <Id2>, <Id3>, <Lettre>, <Chiffre>, ...

EXERCICE 12 : Soit L le langage défini sur l'alphabet $\pi = \{a, b, c, \#\}$ comme suit : $L = \{ a^n b c^n \# / n \geq 0 \}$.

L peut être engendré par la grammaire suivante : $G = (\pi, \{S, A\}, P, S)$ où P est donné par :

$$S \rightarrow A\#$$

$$A \rightarrow aAc \mid b$$

On demande d'écrire, à partir de G, un programme Pascal qui lit des chaînes de caractères et détermine si elles appartiennent à L ou non.

Indication : associer à chaque non terminal de G une procédure Pascal.