Les grammaires

(Système générateur de langage)

Définition : Une grammaire est un moyen permettant de décrire la construction des mots d'un langage

Définition formelle : On appelle grammaire le quadruplet (V,N, X, R)

- **V** : est un ensemble fini de symboles dits ***terminaux***, (vocabulaire terminal) ;
- **N** : est un ensemble fini (disjoint de V) de symboles dits non-terminaux
- **S** : un non-terminal particulier appelé axiome (point de départ de la dérivation) ;
- **R** : est un ensemble de règles de productions de la forme $\alpha \rightarrow \beta$ tel que $\alpha \in (V + N)^+$ et $\beta \in (V + N)^*$.
  La notation $\alpha \rightarrow \beta$ est appelée une dérivation et signifie que α peut être remplacé par β.

NB

- On utilisera les lettres *majuscules* pour les *non-terminaux*, et les lettres ***minuscules*** pour représenter les ***terminaux***.
- Les règles de la forme $\varepsilon \rightarrow \alpha$ sont *interdites*.
- Soit une suite de dérivations : $w_1 \rightarrow w_2 \rightarrow w_3 \rightarrow \ldots \rightarrow w_n$ alors on écrira : $w_1 \rightarrow^* w_n$.
  On dit alors qu'il y a une chaîne de ***dérivation*** qui mène de w_1 vers w_n.

Exemple : Soit la grammaire
$G = (\{a\}, \{S\}, S, \{S \rightarrow aS|\varepsilon\})$. On peut construire la chaîne de dérivation suivante :
$S \rightarrow aS \rightarrow aaS \rightarrow aaaS...$

Les mots générés par une grammaire

Soit une grammaire G = (V,N, S, R).
On dit que le mot $u \in V^*$ est ***dérivé*** (ou bien ***généré***) à partir de **G** s'il existe une suite de dérivation qui, partant de l'axiome **S**, permet d'obtenir ***u***, noté ' $S \rightarrow^* u$ '

Le langage engendré par une grammaire

- Le langage engendré par une grammaire G est l'ensemble de tous les mots générés par la grammaire G est noté L(G).
- Deux grammaires G et G' sont équivalentes si L(G) = L(G').

Exemple : Soit la grammaire $G = (\{a, b\}, \{S, T\}, S, \{S \rightarrow aS|aT, T \rightarrow bT|b\})$.
Elle génère les mots abb et aab parce que
$S \rightarrow aT \rightarrow abT \rightarrow abb$
$S \rightarrow aS \rightarrow aaT \rightarrow aab$.
…..
$S \rightarrow aS \rightarrow aaT \rightarrow aaT \rightarrow aaaT \rightarrow \ldots\ldots \rightarrow aaa\ldots aT$.
On peut facilement voir alors que le langage généré par cette grammaire est : tous les mots sur {a, b} de la forme $a^m b^n$ avec m, n > 0.

Classification de Chomsky

Comment mesurer la complexité d'une grammaire ou d'un langage ?

- Noam Chomsky remarquer que la complexité d'une grammaire (et celle du langage aussi) ***dépend*** de la *forme des règles* de production
- Chomsky a ainsi proposé **quatre** classes (hiérarchiques) de grammaires (et de langages) de sorte qu'**une grammaire de type i génère un langage de type j tel que j ≥ i.**

Soit G = (V,N, S, R) une grammaire, les classes de grammaires de Chomsky sont :

- **Type 3** ou **grammaire régulière** (*à droite* 1) : toutes les règles de production sont de la forme $\alpha \to \beta$ où $\alpha \in N$ et $\beta = aB$ / tel que $a \in V^*$ et $B \in N \cup \{\varepsilon\}$;
- **Type 2** ou **grammaire hors-contexte** : toutes les règles de production sont de la forme $\alpha \to \beta$ où $\alpha \in N$ et $\beta \in (V + N)^*$;
- **Type 1** ou **grammaire contextuelle** : toutes les règles sont de la forme $\alpha \to \beta$ tel que $\alpha \in (N + V)^+$, $\beta \in (V + N)^*$ et $|\alpha| \le |\beta|$. De plus, si ε apparaît à droite alors la partie gauche doit seulement contenir S (l'axiome).

On peut aussi trouver la définition : toutes les règles sont de la forme $\alpha B\beta \to \alpha\omega\beta$ tel que $\alpha,\beta \in (V+N)^*$, $B \in X$ et $\omega \in (V+N)^*$

- **Type 0** : *aucune restriction*. Toutes les règles sont de la forme : $\alpha \to \beta$, $\alpha \in (V + N)^+$, $\beta \in (V+N)^*$

Remarques

- Il existe une relation <u>d'inclusion</u> entre les types de grammaires :
  type 3 ⊂ type 2 ⊂ type 1 ⊂ type 0

- Pour trouver la classe d'un langage on procède cependant comme suit :
  - Chercher une grammaire de <u>type 3</u> qui le génère, si elle existe, le langage est de type 3 (ou **régulier**)
  - <u>Sinon</u>, chercher une grammaire de type 2 qui le génère, si elle existe, le langage est de type 2 (ou **algébrique**)
  - <u>Sinon</u>, chercher une grammaire de type 1 qui le génère, si elle existe, le langage est de type 1 (ou **contextuel**)
  - <u>Sinon</u>, le langage est de type 0.

Exercioce 01 :

Soient les grammaires G_i = ({a, b, c} , {S, A, B, R, T} , S , P_i), (i=1,..,8) ; où les P_i sont :

- P1 : $S \to aA \mid bB$; $A \to a \mid ab$; $B \to b \mid cb$

Type : 3

$L(G_1)$={aa, aab, bb, bcb}

- P2 : $S \to bA$; $A \to aA \mid \varepsilon$

Type : 3

$L(G_2)=\{w \in \{a, b, c\}/w=ba^n\ \ n\ge 0\}$

- P3 : $S \to aAb \mid \varepsilon$; $A \to aSb$; $Ab \to \varepsilon$

Type : 0

$L(G_3)=\{w \in \{a, b, c\}/w=a^{2n+1}b^{2n}$ ou **$w=a^{2n}b^{2n}\ \ n\ge 0\}$**

- P4 : $S \to AB \mid aS \mid a$; $A \to Ab \mid \varepsilon$; $B \to AS$

Type : 2

$L(G_4)=\{w \in \{a, b, c\}/w \in \{b^n,a\}^*\}$

- P5 : $S \to aS \mid bB$; $B \to aC \mid bS \mid \varepsilon$, $C \to aB \mid bC$

Type : 3

$L(G_5)=\{w \in \{a, b, c\}/w=\}$

- P6 : $S \rightarrow aX; X\rightarrow Sb; S \rightarrow \varepsilon$

Type : 2

$L(G_3)=\{w\in\{a, b, c\}/w=a^{2n}b^{2n}\ n\geq 0\}$

- P7 : $S \rightarrow \varepsilon|a|abS|bS$

Type : 3

$L(G_7)=\{w\in\{a, b, c\}/w\in\{ab, b\}^*$ ou **$w\in\{a, aba, ba\}\ \}$**

- P8 : $S \rightarrow AB; A \rightarrow \varepsilon|a; B \rightarrow baB|C; C \rightarrow \varepsilon|b$

Type : 2

$L(G_8)=\{w\in\{a, b, c\}/w=a(ba)^n$ ou **$w=(ba)^n$** ou **$w=a(ba)^nb$** ou **$w=(ba)^nb\}$**

Pour chacune des grammaires G_i (i=1,..,8) ; donner le type de celle-ci, puis trouver le langage engendré par chacune d'elles.

Exercice 02

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

$\{ S \rightarrow aA \mid bB ; A \rightarrow a \mid ab ; B \rightarrow b \mid cb \}$

a) $L_1 = \{ 0^{2n} / n \geq 0 \}$

$G_1 = (\{0\} , \{S\} , S , \{ S \rightarrow 00S| \varepsilon\})$

b) $L_2 = \{ 0^n1^n / n \geq 0 \}$

$G_2 = (\{0,1\} , \{S\} , S , \{ S \rightarrow 0S1| \varepsilon\})$

c) $L_3 = \{ a^nb^{2n} / n \geq 0 \}$

$G_3 = (\{a,b\} , \{S\} , S , \{ S \rightarrow aSbb| \varepsilon\})$

d) $L_4 = \{ a^nb^mc^{n-m} / n \geq m \geq 0 \}$

e) $L_5 = \{$ palindromes de $\{a, b\}^* \}$

$G_5 = (\{a,b\} , \{S\} , S , \{ S \rightarrow aSa| bSb| \ \varepsilon\})$

f) $L_6 = \{ a^mb^na^nb^m / n \geq 1, m \geq 1 \}$

$G_6 = (\{a,b\} , \{S, A\} , S , \{ S \rightarrow aSb| aAb| \ ; A \rightarrow bAa|ba \})$

g) $L_7 = \{ w \in \{a, b\}^* / |w| \equiv 0[3] \}$

$G_7 = (\{a,b\} , \{S, A, B\} , S , \{ S \rightarrow aA| bA|\varepsilon \ ; A \rightarrow aB|bB ; B \rightarrow aS|bS \})$

h) $L_8 = \{ 0^i1^j / i \geq j \geq 0 \}$

$G_8 = (\{0,1\} , \{S, A\} , S , \{ S \rightarrow 0S| A| \ ; A \rightarrow 0A1|\varepsilon \})$

i) $L_9 = \{ 0^i1^j / i \neq j, i \geq 0, j \geq 0 \}=\{ 0^i1^j / i > j$ ou $0^i1^j / i<j, i \geq 0, j \geq 0 \} =$
$=\{ 0^i1^j / i > j , i , j \geq 0 \}$ ou $\{0^i1^j / i<j, i , j \geq 0 \}$

$G_9 = (\{0,1\} , \{S, A\} , S , \{ S \rightarrow A| B \ ; A \rightarrow 0A|0C ; C \rightarrow 0C1|\varepsilon \ ; B \rightarrow 0B1|1D ; D \rightarrow 1D|\varepsilon \ \})$

j) $L_{10} = \{ ab^na / n \geq 0 \}$

$G_{10} = (\{a,b\} , \{S, A, B\} , S , \{ S \rightarrow aAa; A \rightarrow bA|\varepsilon \})$