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 : $w_1 \rightarrow w_2 \rightarrow w_3 \rightarrow \dots \rightarrow w_n$ 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
$S \rightarrow aT \rightarrow abT \rightarrow abb$
$S \rightarrow aS \rightarrow aaT \rightarrow aab$.
.....
$S \rightarrow aS \rightarrow aaT \rightarrow aaT \rightarrow aaaT \rightarrow \dots \rightarrow aaa\dots aT$.
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 $|\alpha| \leq |\beta|$. 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 $\subset$ type 2 $\subset$ type 1 $\subset$ 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 : $S \to aA \mid bB$; $A \to a \mid ab$; $B \to b \mid cb$
- P2 : $S \to bA$; $A \to aA \mid \varepsilon$
- P3 : $S \to aAb \mid \varepsilon$; $A \to aSb$; $Ab \to \varepsilon$
- P4 : $S \to AB \mid aS \mid a$; $A \to Ab \mid \varepsilon$; $B \to AS$
- P5 : $S \to 0S \mid 1B$; $B \to 0C \mid 1S \mid \varepsilon$, $C \to 0B \mid 1C$
- P6 : $S \to 0X$; $X \to S1$; $S \to \varepsilon$
- P7 : $S \to \varepsilon \mid a \mid abS \mid bS$
- P8 : $S \to AB$; $A \to \varepsilon \mid 0$; $B \to 10B \mid C$; $C \to \varepsilon \mid 1$

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 :

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

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

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

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

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

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

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

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

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

j) $L_3 = \{ ab^na / n \geq 0 \}$