Cours: Théorie des langages & Compilation

Chapitre 2: Les grammaires

Réalisé par:
Dr. Sakka Rouis Taoufik

1

I. Gammaire

Définition

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

La définition d'une grammaire permet de:

➔Raisonner sur le langage
➔Construire des algorithmes efficaces pour le traitement des langages
➔Faciliter l'apprentissage des langages.

2

I. Gammaire

Exemple introductif

Pour analyser une classe de phrases simples en français, on peut supposer qu'une phrase est construite de la manière suivante :

– PHRASE → ARTICLE SUJET VERBE ARTICLE COMPLEMENT
– ARTICLE → "un" ou "le "
– SUJET → "garçon" ou "fille"
– VERBE → "voit" ou "mange" ou "porte "
– COMPLEMENT → ARTICLE NOM ADJECTIF
– NOM → "livre" ou "plat" ou "wagon"
– ADJECTIF → "bleu" ou "rouge" ou "vert "

En faisant des substitutions (on remplace les parties gauches par les parties droites) on arrive à générer les deux phrases suivantes.

Le garçon voit un livre rouge
Une fille mange le plat vert

I. Gammaire

Définition Formelle

Une grammaire est définit par un quadruplet <T, N, S, R>

– T est un ensemble fini de symboles dits terminaux, on l'appelle également **vocabulaire terminal** ;

– N est un ensemble fini (disjoint de T) de symboles dits **non-terminaux** ou encore **concepts** ;

– S **un** **non-terminal** particulier appelé axiome (point de départ de la dérivation) ;

– R est un ensemble de règles de productions de la forme g → d tel que $g \in (T \cup N)^+$ et $d \in (T \cup N)^*$.

La notation g → d est appelée une dérivation (ou production) et signifie que g peut être remplacé par d.

Ch2: Les grammaires

I. Gammaire

Conventions:
Les lettres majuscules sont utilisées pour les non-terminaux
Les lettres minuscules sont utilisées pour représenter les terminaux.
Les règles de la forme " ε →… **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 1:
Soit la grammaire **G = < {a}, {S}, S, {S → aS| ε "} >.**
➔On peut construire la chaîne de dérivation suivante :
S → aS → aaS → aaaS ...

5

Ch2: Les grammaires

I. Gammaire

Exemple 2:
Pour l'exemple des phrases en français, on peut définir la grammaire suivante:
G = <T, N, S, R> Avec
T= {garçon, fille, voit, mange, porte, un, le, livre, plat, wagon, bleu, rouge, vert**}**
N= {PHRASE, ARTICLE , SUJET, VERBE, COMPLEMENT, NOM, ADJECTIF**}**
S= PHRASE
R= { PHRASE → ARTICLE SUJET VERBE ARTICLE COMPLEMENT
SUJET → "garçon" ou "fille"
VERBE → "voit" ou "mange" ou "porte«
COMPLEMENT → ARTICLE NOM ADJECTIF
ARTICLE → "un" ou "le"
NOM → "livre" ou "plat" ou "wagon"
ADJECTIF → "bleu" ou "rouge" ou "vert " **}**

6

I. Gammaire

Définition : Soit une grammaire G = <T, N, S, R>.
On dit que le mot $\omega \in T^*$ est dérivé (ou généré) à partir de la grammaire **G** s'il existe une chaîne de dérivation qui mène de l'axiome vers ω. ($S \overset{*}{\rightarrow} \omega$)

Le langage de tous les mots générés par la grammaire G est noté L(G).

Notez Bien: une expression, dérivée à partir de l'axiome, n'est considérée appartenant à L(G) que si elle ne comporte **aucun non-terminal**.

Exemple 3 :
Soit la grammaire **G = <{a, b}, {S, T}, S, {S → aS| aT, T → bT| b}>.**
➔ **G génère** le mot abb parce que S → aT → abT → abb
➔ **G génère** le mot aab parce que S → aS → aaT → aab.
➔On peut conclure que le langage généré par cette grammaire est:

$$L(G)= \{\omega \in \{a, b\}^* \ / \ \omega= a^m b^n , m > 0, n > 0\}.$$

II. Arbre syntaxique d'une Gammaire

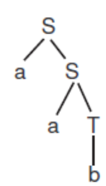
Définition : Étant donnée une grammaire G = <T, N, S, R>,
Les arbres de syntaxe de G sont des arbres où les noeuds internes sont étiquetés par des symboles de **N**, et les feuilles étiquetés par des symboles de **T**, tels que, si le noeud p apparaît dans l'arbre et si la règle $p \rightarrow a_1|\ldots|a_n$ (a_i terminal ou non terminal) est utilisée dans la dérivation, alors le noeud p possède n fils correspondant aux symboles a_i.

Si l'arbre syntaxique a comme racine S, alors il est dit arbre de dérivation du mot ω, avec ω est le mot obtenu en prenant les feuilles de l'arbre dans le sens : gauche vers droite **et** niveau bas vers niveau haut.

Exemple 4 : Soit la grammaire

G = <{a, b}, {S, T}, S, {S → aS| aT, T → bT| b}>

G génère le mot aab
Selon la chaîne de dérivation S → aS → aaT → aab.
Ce qui donne donc l'arbre syntaxique suivant :



Ch2: Les grammaires

III. Classification de Chomsky

La classification de Chomsky est un moyen permettant de maîtriser la complexité des langages ainsi que celle des grammaires qui les génèrent. En effet, certains langage sont plutôt simples et peuvent être décrits par des grammaires facilement compréhensibles (comme l'exemple précédent).

➔Il existe des langages d'une telle complexité que les grammaires qui les génèrent sont trop difficile à appréhender (par exemple, $\{a^n \mid n \text{ est premier}\}$).

➔Il faut savoir que plus une grammaire est complexe à définir, plus les algorithmes qui la manipulent (si jamais ils existent) sont difficiles à apprendre et à mettre en œuvre.

➔On arrive alors à une conclusion bien simple et pleine de bon sens : "utiliser des algorithmes simples pour des langages simples générés par des grammaires simples et réserver les algorithmes difficiles aux langages et grammaires complexes".

9

Ch2: Les grammaires

III. Classification de Chomsky

Une question clé apparaît alors : comment mesurer la complexité d'une grammaire ou d'un langage ?

On peut écarter la cardinalité d'un langage puisqu'elle ne permet pas de distinguer langage simple d'un langage complexe.

Noam Chomsky a eu le génie de remarquer que la complexité d'une grammaire (et celle du langage aussi) dépend de la forme des règles de production de celle-ci.

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.

10

Ch2: Les grammaires

III. Classification de Chomsky

❖**Grammaire de Type 3 (Régulière / Linéaire):**

➢ **(régulière à droite):**

Toutes les règles de production sont de la forme $A \rightarrow \omega B$ ou $A \rightarrow \omega$ où

$A \in N$

$\omega \in T^*$ (Rappelant que $\varepsilon \in T^*$)

$B \in N$ (attention $B \in N$ et pas $B \in N^*$)

➢ **(régulière à gauche):**

Toutes les règles de production sont de la forme $A \rightarrow B\omega$ ou $A \rightarrow \omega$ où

$A \in N$

$\omega \in T^*$ (Rappelant que $\varepsilon \in T^*$)

$B \in N$ (attention $B \in N$ et pas $B \in N^*$)

Ch2: Les grammaires

III. Classification de Chomsky

❖**Grammaire de Type 3 (Régulière / Linéaire):**

Remarque 1: Dans les deux sous types, **pour chaque règle** de production, on a un seul symbole non terminal.

Remarque 2: Si $\varepsilon \in L(G)$ alors $S \rightarrow \varepsilon$ fait partie des règles de production

➢***Exemple***

$$\begin{cases} S \rightarrow aS|T|abU|cc \\ T \rightarrow cT|\varepsilon \\ U \rightarrow abS \end{cases}$$

Ch2: Les grammaires

III. Classification de Chomsky

❖**Grammaire de Type 2 (hors-contexte):**

➢**Toutes** les règles de production sont de la forme $g \rightarrow d$ où

$g \in N$

$d \in (T \cup N)^*$

Remarque : Pour chaque règle de production, **la partie gauche** est composée d'un seul symbole non terminal.

➢***Exemple***

S→ aBa **|** ε

B→ bB **|** b

13

Ch2: Les grammaires

III. Classification de Chomsky

❖**Grammaire de Type 1 (contextuelle):**

➢**Toutes** les règles de production sont de la forme $g \rightarrow d$ tel que

$g \in (T \cup N)^+$

$d \in (T \cup N)^*$

$|g| <= |d|$

De plus, si ε apparaît à droite alors la partie gauche doit seulement contenir S (l'axiome).

➢***Exemple***

S→ bBC | a

B→ aC |aBa | S

aC→ aS

14

Ch2: Les grammaires

III. Classification de Chomsky

❖**Grammaire de Type 0 (aucune restriction):**

➢**Toutes** les règles de production sont de la forme $g \rightarrow d$

$g \in (T \cup N)^+$

$d \in (T \cup N)^*$

➢***Exemple***

S→ bBC | ε

B→ aCa |aBa | S

aC→ a

Remarque : De classe 0 si il existe une règle tq la partie gauche est supérieure à celle de la partie droite (aC→ a dans l'exemple ci-dessus)

15

Ch2: Les grammaires

III. Classification de Chomsky

➢**Propriété 1:**

Les grammaires de type 0 englobent les grammaires de type 1 qui englobent les grammaires de type 2 qui englobent les grammaires de type 3.

Les grammaires hors contexte (Type 2) (et donc aussi les grammaires régulières) jouent un rôle particulièrement important en informatique car la plupart des analyseurs syntaxiques performants ont été développés pour cette catégorie de grammaires.

16

IV. Exercices d'application

Exercice 1:
Classifier les grammaires suivantes selon la hiérarchie de Chomsky. Justifier brièvement votre réponse.

A) $S \rightarrow aA$
$A \rightarrow aS \mid aB$
$B \rightarrow bC$
$C \rightarrow bD$
$D \rightarrow b \mid bB$

B) $S \rightarrow aS \mid aAbb$
$A \rightarrow \varepsilon \mid aAbb$

C) $S \rightarrow bXZ \mid aB$
$B \rightarrow XY \mid S$
$aZ \rightarrow aS$
$bZ \rightarrow aB$
$X \rightarrow a \mid \varepsilon$
$Y \rightarrow aa \mid aB$

D) $S \rightarrow \varepsilon$

Ch2: Les grammaires

IV. Exercices d'application

Exercice 2:
Soit G la grammaire formelle suivante:

$S \rightarrow ABC$
$A \rightarrow a$
$aB \rightarrow b$
$C \rightarrow c$

Quel est le langage généré par la grammaire G tout en précisant son type?

IV. Exercices d'application

Exercice 3:
Trouver une grammaire à contexte libre G pour le langage L suivant:
$L=\{ a^{2i+1}c^{j} b^{2i+1} \mid i >=0 \text{ et } 0<=j<=1\}$

Exercice 4:
Trouver une grammaire à contexte libre G pour le langage L suivant:
$L=\{\omega\, \omega^{T} \mid \omega \in \{a, b\}^{*} , \omega^{T} \text{ est le miroir de } \omega \}$

Exercice 5:
a) Trouver une grammaire formelle G1 pour le langage
$L1=\{ a^{i} b^{j} a^{i} \mid i, j >=1\}$
b) Démontrer ensuite que L(G1)=L1 en précisant le type de G1.

Ch2: Les grammaires

IV. Exercices d'application

Exercice 6:
Soit le mot $x = ((acbc)^{R}.baca)^{R}$ (α^{R} désigne le reflet miroir de α)
1) Donner la chaîne de caractères à laquelle x est égal.
2) Quelle est la valeur de $|x|$, $|x|_a$, $|x|_b$ et $|x|_c$?
3) Donner un préfixe propre de x contenant au moins deux lettres 'c'.
4) Donner un suffixe propre de x contenant une seule lettre 'a'.

Exercice 7 : Soit la grammaire G = ({a, b, c} , {S, A} , S , P) ;
où P contient les règles suivantes : $S \rightarrow aS \mid bA$; $A \rightarrow cA \mid \varepsilon$
a) Déterminer si les mots w1 = abac, w2 = aabccc, w3 = cabbac et w4 = ab sont générés par G.
b) Trouver le langage généré par G (qu'on note L(G)).

IV. Exercices d'application

Exercice 8:

Pour chacun des langages suivants, donner des exemples de mots contenus dans chacun des langages, et des grammaires qui engendrent chacun des langages (Li) avec i=1,..,5 :

1) L1 = { w ∈ {a, b, c}* / w commence par la lettre 'a' } ;
2) L2 = { w ∈ {a, b, c}* / w se termine par la lettre 'a' } ;
3) L3 = { w ∈ {a, b, c}* / w contient au moins une occurrence de la lettre 'a' } ;
4) L4 = { w ∈ {a, b, c}* / w contient au moins deux occurrences la lettre 'a' } ;
5) L5 = { w ∈ {a, b, c}* / w contient au moins deux occurrences consécutives de la lettre 'a' }.