

Université des Sciences et de la Technologie Houari Boumediene

Faculté d'Informatique



Cours Théorie des langages (THL)

Chapitre 3 : Grammaires

2 ième année ING 2023/2024

Dr H.BELHADI

hib.belhadi@gmail.com

19/02/2024

GRAMMAIRE

Un langage peut être décrit comme étant un ensemble de mots satisfaisant un certain nombre de règles appelées grammaire.

Définition

Une grammaire (ou système de substitution) est un quadruplet

$G=(T, N, S, P)$ où :

- T est un **ensemble non vide de terminaux** (l'alphabet sur le quel est défini le langage).

Les symboles de T sont désignés par les lettres minuscules de l'alphabet latin (a, b, c,...).

- N est un **ensemble de non-terminaux** tel que $T \cap N = \emptyset$, ce sont des symboles intermédiaires pour produire de nouveaux objets (c'est les symboles qu'il faut encore définir).

Ils sont désignés par les lettres majuscules de l'alphabet Latin (A, B, D,...).

- $S \in N$ est appelé **axiome** (le point de départ).

Définition (Grammaire) Suite

- P est un ensemble de règles de productions ou de réécritures.

Chaque règle est de la forme $\alpha \rightarrow \beta$

où $\alpha \in (T \cup N)^* N (T \cup N)^*$ et $\beta \in (T \cup N)^*$

Remarque : α et β sont des combinaisons entre terminaux et non-terminaux.
De plus, α contient au moins un non-terminal.

Une règle de production $\alpha \rightarrow \beta$ signifie que :

la séquence de symboles α peut être remplacée par la séquence de symboles β

- α est appelé membre gauche et β membre droit.

Exemple $G=(T, N, S, P)$

$$T=\{a\}$$

$$N=\{S\}$$

$$P= \{S \rightarrow aS, S \rightarrow a\}$$

Intuitivement, cette grammaire permet de générer les mots :

$$S \rightarrow a$$

$$S \rightarrow aS \rightarrow aa$$

$$S \rightarrow aS \rightarrow aaS \rightarrow aaa$$

$$S \rightarrow aS \rightarrow aaS \rightarrow \dots\dots$$

Donc les mots : a, a^2, a^3, \dots ainsi le langage = $\{a^n / n \geq 1\}$.

Notation:

Plusieurs règles ayant même membre gauche :

- Seront regroupées en écrivant une seule fois le membre gauche
- A droite du symbole \rightarrow les différents membres droits séparés par $/$.

Exemple : les trois règles suivantes ont le même membre gauche

$$A \rightarrow Ba$$

$$A \rightarrow bA$$

$$A \rightarrow aA$$

On notera les 3 règles suit : $A \rightarrow Ba / bA / aA$

Remarque : Le symbole $/$ signifie un choix qui n'induit aucun sens de priorité.

Définition (Dérivation directe)

Soit $G=(T, N, S, P)$ une grammaire.

Soient $w_1 \in (T \cup N)^* N (T \cup N)^*$ et $w_2 \in (T \cup N)^*$.

w_1 dérive (ou produit) directement w_2

(ou w_2 dérive directement à partir de w_1) si et seulement si :

il existe une production $\alpha \rightarrow \beta$ dans P telle que : Soit (u et $v \in (T \cup N)^*$)

$w_1 = u \alpha v$ (α est un facteur de w_1) et $w_2 = u \beta v$

(α est remplacé par β dans w_1) avec $u, v \in (T \cup N)^*$.

On écrit alors $w_1 \Rightarrow^{(1)} w_2$ ou simplement $w_1 \Rightarrow w_2$

Exemple (Dérivation directe)

Soit $G=(\{0, 1\}, \{S\}, S, \{S \rightarrow 0S1/01\})$

- S dérive directement $0S1$:

$S \Rightarrow^{(1)} 0S1$ (Règle $S \rightarrow 0S1$)

- $0S1$ dérive directement 0011 :

$0S1 \Rightarrow^{(1)} 0011$ (Règle $S \rightarrow 01$)

- $0S1$ dérive directement $00S11$:

$0S1 \Rightarrow^{(1)} 00S11$ (Règle $S \rightarrow 0S1$)

Définition (Dérivation indirecte)

Soit $G=(T, N, S, P)$ une grammaire. Soient $w_1 \in (T \cup N)^*N(T \cup N)^*$ et $w_2 \in (T \cup N)^*$. On dit que **w_1 dérive (ou produit) indirectement w_2**

(ou w_2 dérive indirectement à partir de w_1) si et seulement si :

w_2 peut être obtenu par une succession de zéro, une ou plusieurs dérivations directes à partir de w_1 . On écrit **$w_1 \Rightarrow^* w_2$** .

Remarques :

- Dans le cas d'une dérivation de longueur zéro, aucune règle de la grammaire n'est utilisée. Donc, on a $w_2 = w_1$.
- On peut indiquer la longueur n de la dérivation (nombre de dérivations directes) comme suit : $w_1 \Rightarrow^{(n)} w_2$

Exemple : En considérant la grammaire précédente

Soit $G=(\{0, 1\}, \{S\}, S, \{S \rightarrow 0S1/01\})$ on a

○ $S \Rightarrow^{(1)} 0S1$ et $0S1 \Rightarrow^{(1)} 0011$

donc $S \Rightarrow^* 0011$

○ $S \Rightarrow^{(1)} 0S1$ et $0S1 \Rightarrow^{(1)} 00S11$

donc $S \Rightarrow^* 00S11$ ou $S \Rightarrow^{(2)} 00S11$

○ $S \Rightarrow^* 000111$ car

$S \Rightarrow^{(1)} 0S1 \Rightarrow^{(1)} 00S11 \Rightarrow^{(1)} 000111$

LANGAGE

Définition

Le langage engendré par une grammaire $G=(T,N,S,P)$, noté $L(G)$, est exactement l'ensemble des mots appartenant à T^* générés (directement ou indirectement) à partir de l'axiome.

$$L(G)=\{w/ S \Rightarrow^* w \text{ et } w \in T^*\} \text{ ou } L(G)= \{w/ S \Rightarrow^* w\} \in T^*$$

Le langage généré par G contient exactement :

- les mots dérivables à partir de l'axiome.
- ne contenant que des symboles terminaux.

LANGAGE

Exemple : Soit $G = (\{a, b\}, \{S\}, S, \{S \rightarrow aSb / ab\})$

On distingue deux types de règles :

Une règle récursive : $S \rightarrow aSb$

Le non-terminal S apparaît dans le membre gauche ainsi que dans le membre droit.

Cette règle va être utilisée de manière récursive comme suit

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaaSbbb \Rightarrow \dots \Rightarrow a^n S b^n$$

Donc, $S \Rightarrow * a^n S b^n$ avec $n \geq 0$

Notons que le mot obtenu n'est pas un mot du langage généré par la grammaire car il contient un non-terminal.

Une règle d'arrêt : $S \rightarrow ab$

Il n'y a pas de non-terminal S dans le membre droit. Dans ce cas précis, que des terminaux dans le membre droit.

LANGAGE

On peut utiliser la règle d'arrêt à tout moment, donc :

$$S \Rightarrow^* a^n S b^n \Rightarrow a^n a b b^n = a^{n+1} b^{n+1} \text{ avec } n \geq 0$$

Donc, $S \Rightarrow^* a^{n+1} b^{n+1}$ avec $n \geq 0$

Dans ce cas, le mot obtenu ne contient que des terminaux et donc c'est un mot du langage généré par la grammaire.

Il n'y a pas d'autres dérivations possibles, donc :

$$\begin{aligned} L(G) &= \{a^{n+1} b^{n+1} / n \geq 0\} \\ &= \{a^n b^n / n \geq 1\} \end{aligned}$$

Grammaires équivalentes

Définition

Deux grammaires G_1 et G_2 sont dites équivalentes, notées $G_1 \equiv G_2$, si elles engendrent le même langage.

$$G_1 \equiv G_2 \Leftrightarrow L(G_1) = L(G_2)$$

Exemple :

Montrer que les deux grammaires G_1 et G_2 sont équivalentes:

$$G_1 = (\{a, b\}, \{S, A, B\}, S, \{S \rightarrow AB, A \rightarrow aA/\varepsilon, B \rightarrow bB/\varepsilon\})$$

$$G_2 = (\{a, b\}, \{S, B\}, S, \{S \rightarrow aS/B, B \rightarrow bB/\varepsilon\})$$

Remarque : Un langage peut être généré par plusieurs grammaires, mais une grammaire ne génère qu'un seul langage.

Grammaires équivalentes

$G_1 = (\{a, b\}, \{S, A, B\}, S, \{S \rightarrow AB, A \rightarrow aA/\epsilon, B \rightarrow bB/\epsilon\})$
 $S \Rightarrow AB$

$A \Rightarrow aA \Rightarrow aaA \Rightarrow aaaA \dots \Rightarrow a^n A, \quad n \geq 0$
 $A \Rightarrow^* a^n A \Rightarrow a^n, \quad n \geq 0$

$B \Rightarrow bB \Rightarrow bbB \Rightarrow bbbB \dots \Rightarrow b^m B, \quad m \geq 0$
 $B \Rightarrow^* b^m B \Rightarrow b^m, \quad m \geq 0$

$S \Rightarrow AB \Rightarrow^* a^n b^m, \quad n, m \geq 0$

$L(G_1) = \{a^n b^m \mid n, m \geq 0\}$

Grammaires équivalentes

$$G_2 = (\{a, b\}, \{S, B\}, S, \{S \rightarrow aS/B, B \rightarrow bB/\epsilon\})$$

$$S \Rightarrow aS \Rightarrow aaS \Rightarrow aaaS \dots$$

$$S \Rightarrow^* a^n S \Rightarrow a^n B, n \geq 0$$

$$B \Rightarrow^* b^m B \Rightarrow b^m, m \geq 0$$

$$S \Rightarrow^* a^n B \Rightarrow a^n b^m, n, m \geq 0$$

$$L(G_2) = \{a^n b^m \mid n, m \geq 0\}$$

Classification des grammaires

Noam Chomsky a décomposé les grammaires formelles en catégories de pouvoir d'expression croissant, c'est-à-dire en groupes successifs pouvant chacun générer une variété de langages plus large que le groupe précédent.

On parle de **Hiérarchie (Classification) de Chomsky**.

Chomsky a défini quatre types de grammaires formelles suivant la nature des règles de production des grammaires.

Classification des grammaires

Type 3 (Grammaires régulières) :

Une grammaire $G=(T, N, S, P)$ est de type 3 ssi elle est soit régulière **droite** soit régulière **gauche**.

○ Grammaire régulière droite :

Toutes les productions dans P sont de la forme :

$$A \rightarrow wB \quad \text{ou} \quad A \rightarrow w \quad \text{avec} \quad A, B \in N \quad \text{et} \quad w \in T^*$$

Exemples : $A \rightarrow aabB$ $A \rightarrow B$ $A \rightarrow aa$ $A \rightarrow \varepsilon$

○ Grammaire régulière gauche :

Toutes les productions dans P sont de la forme :

$$A \rightarrow Bw \quad \text{ou} \quad A \rightarrow w \quad \text{avec} \quad A, B \in N \quad \text{et} \quad w \in T^*$$

Exemples : $A \rightarrow Baab$ $A \rightarrow B$ $A \rightarrow aa$ $A \rightarrow \varepsilon$

Classification des grammaires

Type 3 (Grammaires régulières) :

Remarques :

Les trois dernières règles sont droites et gauches en même temps.

Une grammaire de type 3 ne doit pas contenir en même temps :

- une règle régulière droite ($A \rightarrow wB$) et
- une règle régulière gauche ($A \rightarrow Bw$).

Classification des grammaires

Type 2 (Grammaires algébriques ou grammaires à contexte libre) :

Une grammaire $G=(T, N, S, P)$ est de type 2 si et seulement si toutes les productions de P sont de la forme :

$$A \rightarrow \alpha \quad \text{avec } A \in N \text{ et } \alpha \in (T \cup N)^*$$

Exemples : $A \rightarrow aBb$ $A \rightarrow aBBa$ $A \rightarrow BB$ $A \rightarrow \varepsilon$

Remarque : La seule condition porte sur le membre gauche qui est constitué d'un seul non-terminal.

Classification des grammaires

Type 1 (Grammaires Contextuelles) :

Une grammaire $G=(T, N, S, P)$ est de type 1 si et seulement si toutes les règles de production de P sont de la forme :

$$\alpha A \beta \rightarrow \alpha w \beta \text{ avec } \alpha, \beta \in (T \cup N)^*, A \in N, w \in (T \cup N)^+$$

et une contrainte sur le mot vide : **seul l'axiome peut générer le mot vide** et dans ce cas il n'apparaît dans aucun membre droit d'une règle de production.

La règle $\alpha A \beta \rightarrow \alpha w \beta$ signifie que le non terminal A est remplacé par w si son contexte gauche est α et son contexte droit est β .

Remarque : Les grammaires contextuelles sont appelées aussi grammaires à contexte lié.

Classification des grammaires

Il existe une autre forme pour les grammaires de Type 1 appelées **grammaires monotones**.

Définition (Grammaires monotones)

Une grammaire $G=(T, N, S, P)$ est monotone si et seulement si toutes les règles de production sont de la forme :

$$\alpha \rightarrow \beta \text{ avec } |\alpha| \leq |\beta|$$

et la même restriction sur le mot vide vue pour la première variante.

La caractéristique des grammaires monotones est : la longueur du mot obtenu après chaque dérivation **ne peut jamais décroître**.

Ainsi, si on cherche à dériver un mot de longueur 6 et qu'on a obtenu un mot de longueur 7 ou plus, on abandonne alors la dérivation en cours. Il faut explorer les autres dérivations.

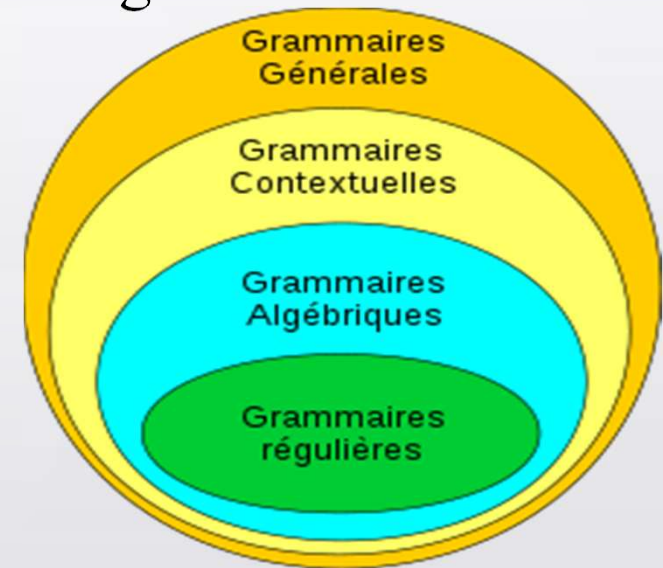
Classification des grammaires

Type 0 (Grammaire sans restriction ou Grammaire Générale)

Une grammaire $G=(T, N, S, P)$ est de type 0 si la forme des règles de production dans P n'est l'objet d'aucune restriction

Ainsi, nous avons la hiérarchie de Chomsky :

type 3 \subseteq type 2 \subseteq type 1 \subseteq type 0.



Remarque : Pour une grammaire G donnée, on cherche à trouver le plus petit type de G au sens de l'inclusion (du plus haut rang)

Classification des grammaires

Exemple : Soit une grammaire $G=(\{a, b\}, \{S, A\}, S, P)$ où:

$$P=\{ S \rightarrow aaS/A, A \rightarrow bbA/bb \}.$$

G est une grammaire de **type 2** car toutes les règles sont de la forme : $A \rightarrow \alpha$ avec $A \in N$ et $\alpha \in (T \cup N)^*$

Mais elle est aussi de **type 3** (régulière droite) car toutes les règles sont de la forme: $A \rightarrow wB$ ou $A \rightarrow w$ avec $A, B \in N$ et $w \in T^*$

On dira qu'elle est de type 3.

C'est le plus petit type au sens de l'inclusion.

Etant donné une grammaire G , on vérifie dans l'ordre :

Si elle est de type 3

Sinon si elle est de type 2

Sinon si elle est de type 1

Sinon elle est de type 0.

19/02/2024

Classification des langages

A chaque type de grammaire est associé un type de langage:

- Les grammaires de **type 3** génèrent les langages **réguliers**.
- Les grammaires de **type 2** génèrent les langages **algébriques ou à contexte libre**
- Les grammaires de **type 1** génèrent les langages **contextuels ou à contexte lié**
- Les grammaires de **type 0** génèrent tous les langages **récursivement énumérables**.

Classification des langages

Définition (Type de d' un langage)

Un langage est **de type i** s'il existe une **grammaire de type i** qui le génère (engendre).

Un langage est **strictement de type i** :

- s'il est engendré par une **grammaire de type i**
- et il **n'existe pas** de grammaire de **type supérieur à i** qui l'engendre.

Exemple: $\{a^n b^m / n, m \geq 0\}$ est de type 3 mais $\{a^n b^n / n \geq 0\}$ est strictement de type 2

Remarques :

- Un langage peut être généré par différentes grammaires qui peuvent être de type différent.
- Un langage **prend le plus petit type au sens de l'inclusion**

Classification des langages

Exemple : Soit le langage $L_1 = \{ ww^R / w \in \{a, b\}^* \}$

L_1 est généré par la grammaire $G_1 = (\{a, b\}, \{S\}, S, P)$ où:

$P = \{ S \rightarrow aSa / bSb / \varepsilon \}$.

G n'est pas de type 3 car la règle $S \rightarrow aSa$ n'est ni régulière droite ni régulière gauche.

Cette grammaire est de type 2 car toutes les règles sont de la forme $A \rightarrow \alpha$ avec $A \in N$ et $\alpha \in (T \cup N)^*$.

Donc L_1 est de type 2 car il est généré par une grammaire de type 2.

Classification des langages

Exemple : Soit le langage $L_2 = \{a^{2n}b^m / n, m \geq 0\}$

L_2 est généré par la grammaire $G_2 = (\{a, b\}, \{S, A, B\}, S, P)$ où
 $P = \{ S \rightarrow AB, A \rightarrow aaA / \varepsilon, B \rightarrow bB / \varepsilon \}$

G_2 n'est pas de type 3 car la règle $S \rightarrow AB$ n'est ni régulière droite ni régulière gauche.

G_2 est de type 2 car toutes les règles sont de la forme
 $A \rightarrow \alpha$ avec $A \in N$ et $\alpha \in (T \cup N)^*$.

L_2 est donc de type 2 car il est généré par G_2 qui est de type 2.

Peut-on trouver une grammaire de type 3 qui le génère ?

Classification des langages

Soit la grammaire $G_3 = (\{a, b\}, \{S, B\}, S, P)$ où
 $P = \{ S \rightarrow aaS / B, B \rightarrow bB / \varepsilon \}.$

G_3 est une grammaire de type 3. En effet, elle est régulière droite. Toutes les règles sont de la forme :

$$A \rightarrow wB \text{ ou } A \rightarrow w \quad \text{avec } A, B \in N \text{ et } w \in T^*.$$

La grammaire G_2 génère le langage L_2 .

Donc, L_2 est de type 3. C'est le plus petit type au sens de l'inclusion. L_2 est strictement de type 3.

Etant donné un langage L , on cherche toujours à déterminer le type le plus petit au sens de l'inclusion.

Exemples classiques de langages

Type 3 : $L = \{a^n b^m / n, m \geq 0\}.$

$L = \{\epsilon, a, b, aa, bbb, aaabb, abbb, \dots\}$

Une grammaire de type 3 qui engendre L est :

$$G = (\{a, b\}, \{S, R\}, S, \{S \rightarrow aS / R ; R \rightarrow bR / \epsilon\}.$$

Type 2 : $L = \{a^n b^n / n \geq 0\}$

$L = \{\epsilon, ab, aabb, aaabbb, aaaabbbb, \dots\}$

Une grammaire de type 2 qui engendre L est :

$$G = (\{a, b\}, \{S\}, S, \{S \rightarrow aSb / \epsilon\}$$

Exemples classiques de langages

Type 1 : $L = \{a^n b^n c^n / n \geq 1\}$

$L = \{abc, aabbcc, aaabbbccc, \dots\}$

L est engendré par la grammaire suivante :

$G_1 = (\{a, b\}, \{S, Q\}, S, P)$ où P est défini par :

$S \rightarrow aSQ / abc$

$cQ \rightarrow Qc$

$bQc \rightarrow bbcc$

- Les deux premières règles génèrent $a^n abcQ^n$.
- La 3ième règle déplace Q vers la gauche entre les c .
- La dernière règle remplace Q par bc s'il se trouve dans le contexte (b, c) . b est le contexte gauche et c le contexte droit.

La grammaire G_1 est **monotone** donc elle est de **type 1**.

Exemples classiques de langages

Exemple : $L = \{ a^n b^n c^n / n \geq 0 \}$

Etudions la grammaire suivante :

$G_2 = (\{a, b\}, \{S, Q\}, S, P)$ où P est défini par

$$S \rightarrow aSQ / abc / \varepsilon$$

$$cQ \rightarrow Qc$$

$$bQc \rightarrow bbcc$$

La grammaire G_2 n'est pas monotone car elle ne respecte pas la contrainte du mot vide. L'axiome génère ε et nous avons la règle $S \rightarrow aSQ$ où S apparaît à droite.

Exemples classiques de langages

Exemple : $L = \{ a^n b^n c^n / n \geq 0 \}$

L est engendré par la grammaire suivante :

$G_2 = (\{a, b\}, \{S, R, Q\}, S, P)$ où P est défini par

$$S \rightarrow aRQ / abc / \varepsilon$$

$$R \rightarrow aRQ / abc$$

$$cQ \rightarrow Qc$$

$$bQc \rightarrow bbcc$$

Cette grammaire est **monotone** (l'axiome S génère le mot vide et il n'apparaît pas à droite) donc de type 1.

Exemples classiques de langages

Exemple : $L = \{ a^n b^n c^n / n \geq 1 \}$

Etudions la grammaire suivante

$G = (\{a, b, c\}, \{S, H, B, C\}, S, P)$ où P est défini par :

$S \rightarrow aSBC / aBC$

$CB \rightarrow HB \quad HB \rightarrow HC \quad HC \rightarrow BC$

$aB \rightarrow ab \quad bB \rightarrow bb \quad bC \rightarrow bc \quad cC \rightarrow cc$

Toutes les règles sont de la forme :

$\alpha A \beta \rightarrow \alpha w \beta$ avec $\alpha, \beta \in (T \cup N)^*$, $A \in N$, $w \in (T \cup N)^+$

La condition sur le mot vide est respectée

Ainsi, cette grammaire est **contextuelle** donc de **type 1**.

Langages et automates

Enfin, à **chaque type de langage** est associé un **type d'automate** qui permet de reconnaître les langages de sa classe :

- Les langages de **Type 3** appelés aussi **langages réguliers** sont reconnus par des **automates d'états finis (AEF)**.
- Les langages de **Type 2** appelés aussi **langages algébriques** sont reconnus par des **automates à piles (AàP)**.
- Les langages de **Type 1** appelés aussi **langages contextuels** sont reconnus par des **automates à bornes linéaires (ABL)**.
- Les langages de **Type 0**, appelés aussi **langages récursivement énumérables**, sont reconnus par des **machines de Turing**.