Notions Fondamentales de la Théorie des Langages

MAZOUZ Samia & LAICHI Boualem

blaichi@usthb.dz

USTHB – Faculté d'Informatique

2021/2022

ALPHABET

Définition (Alphabet)

Un alphabet X est **un ensemble fini et non vide**.

Les éléments de cet ensemble sont appelés **des lettres** ou **symboles**.

Exemples :

- Alphabet binaire $X = \{0, 1\}$
- Alphabet décimal $X = \{0, 1, \ldots, 9\}$
- Alphabet des gènes (ADN) $X = \{A, T, C, G\}$
- Alphabet des expressions arithmétiques $X = \{+, -, *, \div, (,), 0, 1, \ldots, 9\}$.
- Alphabet latin (Anglais, …) $X = \{a, b, c, \ldots, z\}$

Mots

Définition (Mot)

Un mot sur un alphabet X est une suite (combinaison) finie, éventuellement vide, d'éléments de X.

Exemples :

Alphabet	Mots
{0,1}	010 ,10 ,000 ,10011001, 111111, 0, 1
{A, C, G, T}	ATTGCT, TTTGTACGT, GTTTCA, A, G
{+, -, *, ÷, (,), 0, …, 9}	5+3, 7***, +***))), ((4+8) ÷2), 0, 1

MOTS

Notations :

- Le mot vide (suite vide d'éléments) est noté ε .

- L'ensemble des mots formés à partir d'un alphabet X est noté **X***.

  Exemple : Si X={a} alors X* est défini comme suit :

  $X^*=\{\varepsilon , a, aa, aaa, aaaa, ...\}$

- $\mathbf{X}^+$ est l'ensemble des **mots non vides**.

  On a $X^*=X^+ \cup \{\varepsilon\}$.

Remarque : Les ensembles X* et X^+ sont infinis.

Opérations sur les Mots

1) Concaténation

Définition : Soient w_1 et w_2 deux mots de X*. On définit la concaténation comme la juxtaposition de w_1 et w_2 et on la note **$w_1.w_2$** (ou w_1w_2).

Ainsi, **si** $w_1= a_1\ldots a_n$ et $w_{2=} (b_1\ldots b_m)$

alors $w_1.w_2 = a_1\ldots a_n\, b_1\ldots b_m$

Remarques :

- $\varepsilon.w=w.\varepsilon=w$ (ε est :

  l'élément neutre de la concaténation)
- La concaténation n'est pas commutative ($w_1w_2 \neq w_2w_1$)
- La concaténation est associative $(w_1w_2)w_3 = w_1(w_2w_3)$

Opérations sur les Mots

2) Longueur

Définition : On appelle longueur d'un mot w sur un alphabet X la somme des occurrences des différents symboles le constituant. Elle est notée lg(w) (ou | w |).

Formellement :

- $lg(\varepsilon)=0$
- $lg(a)=1 \quad \forall a \in X$
- $lg(a.w) = 1+lg(w), \ \forall a \in X, \ \forall w \in X^*$

Exemples : lg(ab)=2 lg(aba)=3 lg(abb)=3

Notation : $|w|_a$ désigne le nombre de a dans le mot w. $|aba|_a = 2$

Remarque :

La fonction longueur est une application de X* vers N.

Opérations sur les Mots

3) Miroir

Définition: On appelle mot miroir d'un mot w, noté **Mir(w)** ou **w^R** le mot obtenu en inversant les symboles de w.

Ainsi **si** $w=a_1 \ldots a_n$ **alors** $Mir(w)= a_n \ldots a_1$.

Formellement :

- $Mir(\varepsilon)= \varepsilon$
- $Mir(a)=a \qquad \forall a \in X$
- $Mir(a.w) = Mir(w).a \qquad \forall a \in X, \ \forall w \in X^*$

Exemples : Le miroir du mot abbaa est aabba.
Le miroir du aba est le mot lui-même (mot palindrome)

Remarque : $(w^R)^R = w$

Opérations sur les Mots

4) Puissance

Définition : La puissance d'un mot w est définie par récurrence de la manière suivante :

- $w^0 = \varepsilon$
- $w^{n+1} = w^n.w \quad \forall n \geq 1$

Exemple :

Les puissances du mot **abb** sont :

{ε, abb, abbabb, abbabbabb, ...}

Opérations sur les Mots

5) Factorisation

Définition : Soient v et w deux mots de X*.

- v est **facteur ou sousmot** du mot w si et seulement s'il existe deux mots u_1, u_2 appartenant à X* tel que : **w = u_1.v.u_2**
- Le mot v est **facteur propre** du mot w si $\mathbf{u_1 \neq \varepsilon}$ **et** $\mathbf{u_2 \neq \varepsilon}$.
- Le mot v est **facteur gauche (ou préfixe)** de w si $\mathbf{u_1 = \varepsilon}$.
- Le mot v est **facteur droit (ou suffixe)** de w si $\mathbf{u_2 = \varepsilon}$

Exemples : Soit le mot w= aabbba, nous avons :

- Le mot v1 = abb est facteur de w, c'est un facteur propre (aabbba)
- Le mot v2 = aab est facteur gauche de w (aabbba)
- Le mot v3 = ba est facteur droit de w (aabbba)

LANGAGE

Définition : Soit X un alphabet. On appelle langage formel défini sur X tout sous-ensemble de X*.

Exemples :

- L_1 = l'ensemble des mots de {a, b}* qui commencent par a

  = {**a**, **a**a, **a**b, **a**aa, **a**ab, **a**ba, **a**bb, …..}

  = {**a**w / w∈{a,b}*}

- L_2 = l'ensemble des mots de {a, b}* de longueur strictement inférieure à 3

  = {ε, a, b, aa, ab, ba, bb}

Remarque : L_1 est un langage infini et L_2 est un langage fini.

LANGAGE

- Un langage **vide** est un langage qui ne contient aucun mot. Il est noté ∅.
- Un langage **fini** est un langage qui **contient un nombre fini de mots.** Dans l'exemple précédent L_2 est fini. Un langage fini peut être décrit par l'énumération des mots qui le composent.
- Un langage **infini** est un langage qui contient une infinité de mots. Dans l'exemple précédent L_1 est infini. De façon générale, un langage infini peut être décrit par :
  1) application d'opérations à des langages plus simples,
  2) un ensemble de règles appelées grammaires.
- Un langage est dit propre s'il ne contient pas le mot vide.
- Le langage ∅ est **différent** du langage {ε}

Opérations sur les langages

Les langages étant des ensembles, on peut effectuer sur eux les opérations définies sur les ensembles :

- Union : $L_1 \cup L_2 = \{w / w \in L_1 \text{ ou } w \in L_2\}$
- Intersection : $L_1 \cap L_2 = \{w / w \in L_1 \text{ et } w \in L_2\}$
- Complément : $\overline{L} = \{w / w \in X^* \text{ et } w \notin L\}$
- Différence : $L_1 - L_2 = \{w / w \in L_1 \text{ et } w \notin L_2\}$
- Produit : $L_1 \times L_2 = \{(w_1, w_2) / w_1 \in L_1 \text{ et } w_2 \in L_2\}$

OPÉRATIONS SUR LES LANGAGES

De plus, les opérations définies sur les mots peuvent être étendues aussi aux langages.

Soient deux langages L_1 et L_2 respectivement définis sur les alphabets X_1 et X_2 et soit L un langage défini sur l'alphabet X.

- **La concaténation de langages**

$$L_1.L_2=\{w_1.w_2 \ / \ w_1 \in L_1 \text{ et } w_2 \in L_2\}$$

Exemple : Si $L_1=\{a, ba\}$ et $L_2=\{b, bb, ab\}$ alors

$$L_1.L_2=\{ab, abb, aab, bab, babb, baab\}$$

Remarques :

$$\{\varepsilon\}.L = L.\{\varepsilon\} = L$$

$$\varnothing.L = L.\varnothing = \varnothing$$

OPÉRATIONS SUR LES LANGAGES

- **Langage miroir** $L^R = \{w^R / w \in L\}$
- **Puissance concaténative** $L^0 = \{\varepsilon\}$ et $L^{n+1} = L^n.L$
- **Fermeture itérative ou Etoile** $L^* = L^0 \cup L^1 \cup L^2 \cup \ldots \cup L^k \cup \ldots = \cup_{i \geq 0} L^i$
- **L'étoile propre (ou ε libre)** $L^+ = \cup_{i \geq 1} L^i$

Remarques : si $\varepsilon \in L$ alors $\varepsilon \in L^*$ et $\varepsilon \in L^+$

si $\varepsilon \notin L$ alors $\varepsilon \in L^*$ et $\varepsilon \notin L^+$

GRAMMAIRE

Un langage peut être décrit comme étant un ensemble de mots satisfaisant un certain nombre de règles…

…appelées grammaire.

Définition (Grammaire)

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 lequel 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 $\mathbf{T \cap N = \varnothing}$, 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, C, ...
- $S \in N$ est appelé **axiome.**

GRAMMAIRE

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.

GRAMMAIRE

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

- T={a}
- N={S}
- P={S→aS, S→a}

Intuitivement, cette grammaire permet de **générer** les mots :

S→a

S→ aS →aa

S→ aS →aaS →aaa

S→ aS →aaS →…….

Donc les mots : a, a^2, a^3,… ainsi le langage généré (ou engendré) par la grammaire G est : $\{a^n / n \geq 1\}$

GRAMMAIRE

Notations :

Plusieurs règles **ayant même membre gauche :**

- Seront regroupées en écrivant une seule fois le membre gauche
- A droite du symbole → 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 comme suit : $A \rightarrow Ba / bA / aA$

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

GRAMMAIRE

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 :

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

En d'autres termes : $u\alpha v \rightarrow u\beta v$

(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 $w_1 \Rightarrow w_2$

GRAMMAIRE

Exemples :

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)

GRAMMAIRE

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 $\mathbf{w_1 \Rightarrow^* w_2}$.

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 : $w_1 \Rightarrow^{(n)} w_2$ exp : $w_1 \Rightarrow^{(3)} w_2$

GRAMMAIRE

Exemples: En considérant la grammaire précédente

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

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

Remarque : $000111 = 0^3 1^3 \neq (01)^3 = 010101$

GRAMMAIRE

Définition (Langage)

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.

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

Le langage généré par G contient exactement :

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

GRAMMAIRE

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

On distingue deux types deux règles :

- **Une règle récursive :** S → 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 :

  $$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow \ldots\ldots \Rightarrow a^nSb^n$$

  Donc, $\mathbf{S \Rightarrow^* a^nSb^n}$ avec n≥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→ ab

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

GRAMMAIRE

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

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

Donc, $\mathbf{S \Rightarrow^* a^{n+1}b^{n+1}}$ 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 :

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

GRAMMAIRE

Définition (Grammaires équivalentes)

Deux grammaires G_1 et G_2 sont dites équivalentes, notées $G_1 \equiv G_2$, si elles engendrent (génèrent) 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:

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

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

Dans G_1: $S \rightarrow AB$ et $A \rightarrow aA \rightarrow aaA \rightarrow aaaA\ldots$ donc $A \Rightarrow^* a^n$ (pareil: $B \Rightarrow^* b^m$)

Donc $L(G_1)=$ **$\{a^n b^m / n\geq 0$ et $m\geq 0\}$**

Dans G_2 : $S \rightarrow aS \rightarrow aaS \rightarrow \ldots \rightarrow a^nS \rightarrow a^nB \rightarrow a^nbB \rightarrow a^nbbB \rightarrow \ldots \rightarrow a^nb^mB \rightarrow a^nb^m$

Donc $L(G_2)=$ **$\{a^n b^m / n\geq 0$ et $m\geq 0\}$**

Ces deux grammaires génèrent le même langage : **$\{a^n b^m / n\geq 0$ et $m\geq 0\}$**

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

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.

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 :**

Si toutes les productions dans P sont de la forme :

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

Exemples : $A \rightarrow aabB$ $A \rightarrow B$ $A \rightarrow aa$ $A \rightarrow \varepsilon$: sont de type 3 (RD)

$A \rightarrow aBB$ $A \rightarrow aBa$ $A \rightarrow AB$: ne sont pas de type 3

CLASSIFICATION DES GRAMMAIRES

- **Grammaire régulière gauche :**

Si toutes les productions dans P sont de la forme :

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

Exemples : $A \rightarrow Baab$ $A \rightarrow B$ $A \rightarrow aa$ $A \rightarrow \varepsilon$: sont de type 3 (RG)

Remarques :

- Les trois dernières règles sont droites et gauches en même temps.
- Une grammaire de type 3 ne doit pas contenir en même temps :

  - des règles régulières droites $(A \rightarrow wB)$ et
  - des règles régulières gauches $(A \rightarrow Bw)$.

(Pour qu'une grammaire soit régulière : il faut que toutes ses règles soient régulières droites ou toutes régulières gauches)

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 :

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

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

sont de type 2 (algébriques)

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.

Exemples : $aAb \rightarrow aBBb$ $aA \rightarrow aBBa$ $Ab \rightarrow BBb$ $A \rightarrow AB/a$ sont de type 1

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

CLASSIFICATION DES GRAMMAIRES

Type 0 (Grammaire sans restriction ou Grammaire Générale)

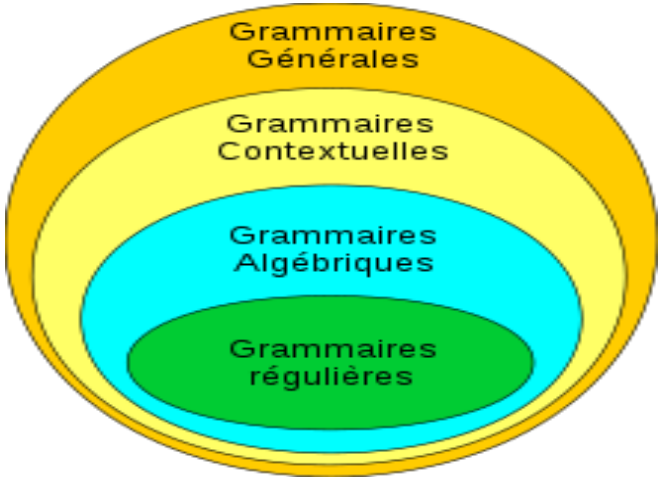
Une grammaire G=(T, N, S, P) est de type 0 si la forme des règles de production dans P n'est l'objet d'aucune restriction

Donc, Type 3 : **$A \rightarrow wB$** (**ou** $A \rightarrow Bw$) et $A \rightarrow w$ $A \in N$ et $w \in T^*$

Type 2 : **$A \rightarrow w$** $A \in N$ et $w \in (T \cup N)^*$

Type 1 : **$\alpha A \beta \rightarrow \alpha w \beta$**

Type 0 : aucune contrainte



Ainsi, nous avons la hiérarchie de Shomsky :

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→ aaS / A, A → bbA / bb }.

G est une grammaire de **type 2** car toutes les règles sont de la forme :

$\mathbf{A \rightarrow \alpha}$ 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 : $\mathbf{A \rightarrow wB}$ **ou** $\mathbf{A \rightarrow w}$ avec $A,B \in N$ et $w \in T^*$

Dans ce cas, on dira qu'elle est de type 3.

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

Etant donnée 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.

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 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 $L_1 = \{ ww^R / w \in \{a, b\}^* \}$

$L_1 = \{\varepsilon, aa, bb, abba, aaaa, bbbb, abbbba, abaaba, \ldots \}$

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 :
$\mathbf{A \rightarrow \alpha}$ 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 $L_2 = \{a^{2n}b^m / n, m \geq 0\}$

$L_2 = \{\varepsilon, aa, b, aab, aaaa, bbb, aabbbb, aaaabbbbbbbb, \ldots\}$

L_2 est généré par la grammaire $G_2=(\{a, b\}, \{S, A, B\}, S, P)$ où

$$P=\{ S \rightarrow AB,\quad A \rightarrow aaA/\varepsilon,\quad B \rightarrow bB/\varepsilon\}$$

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

Par contre, G_2 est de type 2 car toutes les règles sont de la forme

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

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

Toujours pour le langage $L_2 = \{a^{2n}b^m / n, m \geq 0\}$

$L_2 = \{\varepsilon, aa, b, aab, aaaa, bbb, aabbbb, aaaabbbbbbbb, \ldots\}$

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 :

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

La grammaire G_3 génère le langage L_2.

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

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={ε, a, b, aa, bbb, aaabb, abbb, …., a---ab-------b, ….}

Une grammaire de type 3 qui engendre L est :

G= ({a,b}, {S,B}, S, {S → aS / B ; B → bB / ε }.

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

L={ε, ab, aabb, aaabbb, aaaabbbb, …., a---ab---b, …}

Une grammaire de type 2 qui engendre L est :

G =({a,b},{S},S, {S → aSb / ε}

LANGAGES ET AUTOMATES

Enfin, à **chaque type de langage** est associé u**n type d'automat**e 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.**
- Les langages de **Type 2** appelés aussi langages algébriques sont reconnus par des **automates à piles.**
- Les langages de **Type 1** appelés aussi langages contextuels sont reconnus par des **automates à bornes linéaires.**
- Les langages de **Type 0** appelés aussi langages récursivement énumérables sont reconnus par des **machines de Turing**.