Les Notions Fondamentales de la théorie des langages

Samia Mazouz

Département Informatique

FEI-USTHB 2018-2019

Campusvirtuel.usthb.dz

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,…., 9}
- Alphabet des gènes (ADN). X={A, T, C, G}
- Alphabet des expressions arithmétiques X={+, *, (,), Nb} où Nb désigne un nombre quelconque.

MOTS

Définition (Mot)

Un mot sur un alphabet X **une suite finie éventuellement vide d'éléments** X.

Exemples

Alphabet	Mots
{0, 1}	0, 10, 010001, 0011001, 111111
{A, C, G, T}	ATTGCT, TTTGTACGT, GTTTCA
{+, *, (,), Nb}	Nb+Nb, Nb***, +***))), Nb*Nb+Nb

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*={**ε**, a, aa, aaa, aaaa, ...}

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

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

Remarque Les ensembles X* et X+ sont infinis.

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 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:

- **$\varepsilon.W = W.\varepsilon$**
- **La concaténation n'est pas commutative**
- **La concaténation est associative**

LONGUEUR

Définition (Longueur)

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, on a :

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

Exemples

lg(Nb*Nb) **=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 $w=a_1 \dots a_n$ alors $Mir(w) = a_n \dots a_1$.

Formellement, on a :

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

Exemple Le miroir du mot abbaa est aabba.
Le miroir de aba est le mot lui même ie aba, c'est un mot palindrome.

Remarques

$$(w^R)^R = w$$

PUISSANCE

Définition (Puissance d'un mot)

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

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

Exemple

Les puissances du mot abb sont {ε, abb, abbabb, abbabbabb,….}

FACTORISATION

Définition (Factorisation) 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

$$w = u_1.\mathbf{v}.u_2$$

- Le mot **v est facteur propre** du mot w ssi $u_1 \neq \varepsilon$ **et** $u_2 \neq \varepsilon$.
- Le mot v est **facteur gauche** (ou préfixe) de w si $u_1 = \varepsilon$.
- Le mot v est **facteur droit (ou suffixe) de w** si $u_2 = \varepsilon$.

Exemples Soit le mot w= aabbba, nous avons :

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

LANGAGE

Définition (Langage) 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 inférieure strictement à 3**

  = {ε, a, b, aa, ab, ba, bb}

LANGAGE

Remarques

- Un langage **fini** est un langage qui **contient un nombre fini de mots.**

Un langage fini peut être décrit par l'énumération des mots qui le composent.

Dans l'exemple précédent L2 est fini alors que L1 est infini.

- Un langage **vide** est un langage qui ne contient aucun mot et il est noté ∅**.**

- 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
- Intersection
- Complément
- Différence
- Produit

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 (produit)**

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

Remarques

$$\varnothing.L_1=L_1.\varnothing=\varnothing$$

mais $\{\varepsilon\}.L_1=L_1.\{\varepsilon\}=L$

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**

  $L^* = L^0 \cup L^1 \cup \ldots\ldots \cup L^k \cup \ldots\ldots$

  $= \cup_{i \geq 0} L^i$

- **L'étoile propre** (ou ε libre) de L, noté L^+, est défini par :

  $L^+ = \cup_{i \geq 1} L^i$

GRAMMAIRE

Définition (Grammaire)

Une grammaire est un quadruplé 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.

- $S \in N$ est appelé **axiome.**

GRAMMAIRE

Définition (Grammaire) Suite

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

Chaque règle est de la forme $\alpha \rightarrow \beta$ avec $\alpha, \beta \in (T \cup N)^*$ et α contient au moins un non-terminal.

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

la séquence de symboles α peut êre remplacée par la séquence de symboles β.

- α est appelé membre gauche et β membre droit.

Grammaire

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

- $T=\{a\}$
- $N=\{S\}$
- $P=\{S \rightarrow aS,\ S \rightarrow a\}$

Intuitivement, cette grammaire permet de générer les mots $a, a^2, a^3, \ldots$ ie le langage $\{a^n / n \geq 1\}$.

GRAMMAIRE

Notations

Plusieurs règles **ayant même membre gauche :**

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

Exemple Les trois règles suivantes ont le même membre gauche A :

$A \rightarrow Ba$,
$A \rightarrow bA$
$A \rightarrow aA$

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

Grammaire

Définition (Dérivation directe)

Soient G=(T, N, S, P) une grammaire, $w_1 \in (T \cup N)^+$ 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 :

- **$w_1 = u\alpha v$ (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 $w_1 \Rightarrow w_2$

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

w_1 **dérive indirectement (ou produit) directement** w_2

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

w_2 peut êre obtenu par une succession de zéro, une ou plusieurs dérivations directes à partir de w_1.

On écrit alors $w_1 \Rightarrow^* w_2$.

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

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

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

GRAMMAIRE

Définition (Langage)

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

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

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

Le langage généré par G contient exactement :

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

Grammaire

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

On distingue deux types deux règles :

- Une règle récursive : $S \rightarrow aSb$

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

Donc, cette règle peut être utilisée de manière récursive comme suit :

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

Donc, $S \Rightarrow^* a^nSb^n$ 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

Grammaire

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

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

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 :

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

$$=\{a^nb^n \;/\; n \geq 1 \}$$

GRAMMAIRE

Définition (Grammaires équivalentes)

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

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

Exemples

Montrer que les deux grammaires G_1 et G_2 sont équivalentes:

G_1= ({a, b}, {S, A, B}, S, {S→AB, A →aA/ε, B →bB/ ε}

G_2= ({a, b}, {S, B}, S, {S→aS/B, B →bB/ ε}

CLASSIFICATION DES GRAMMAIRES

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

Type 3 (Grammaire régulière) Une grammaire G=(T, N, S, P) est de type 3 ssi

elle soit régulière **droite** soit régulière **gauche.**

- **Grammaire régulière droite**

Toutes les productions dans P sont de la forme :

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

CLASSIFICATION DES GRAMMAIRES

- **Grammaire régulière gauche**

Toutes les productions dans P sont de la forme :

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

Remarque

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 (Grammaire à contexte libre ou grammaire algébrique)

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

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

Remarque

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

Classification des grammaires

Type 1 (Grammaire Contextuelle ou Grammaire monotone)

Une grammaire G=(T, N, S, P) est de type 1 ssi

soit G est à contexte liée soit G est monotone.

- **Grammaire à contexte lié**

Si toutes les règles de production de P sont de la forme :

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

et une **contrainte sur le mot vide** (seul l'axiome peut générer le mot vide sous réserve qu'il n'apparaît dans aucun membre droit d'une règle).

La règle $\alpha \mathbf{A} \beta \rightarrow \alpha \mathbf{w} \beta$: le non-terminal A est remplacé par w si son contexte gauche est α et son contexte droit est β.

CLASSIFICATION DES GRAMMAIRES

- **Grammaire monotone** :

  Toutes les règles de production sont de la forme :

  $\alpha \rightarrow \beta$ avec $|\alpha| \leq |\beta|$

et la même restriction sur le mot vide que pour les grammaires à contexte lié.

La caractéristique des grammaires monotones est :

la longueur du mot obtenu après chaque dérivation ne peut jamais décroitre.

Ainsi, si on chercher à dériver un mot de longueur 6

et qu'on a obtenu un mot de longueur 7 ou plus :

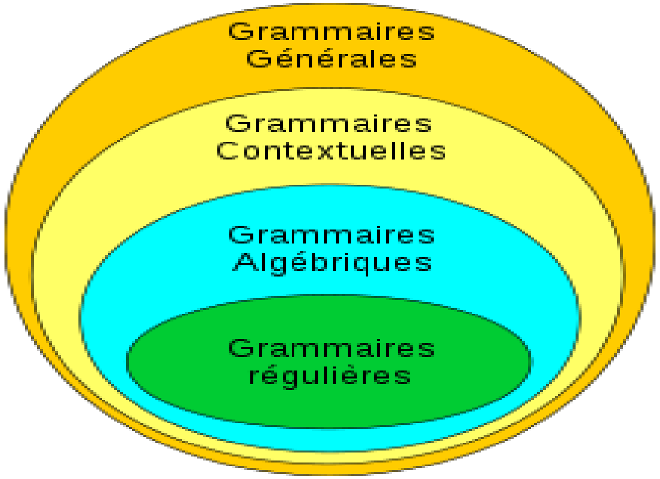
On abandonne alors la dérivation en cours.

Il faut explorer les autres dérivations.

CLASSIFICATION DES GRAMMAIRES

Type 0 (Grammaire sans restriction/ Grammaire Générale) :

Si la forme des règles de production dans P n'est l'objet d'aucune restriction .



On a type 3 $\subseteq$ type 2 $\subseteq$ type 1 $\subseteq$ type 0.

Pour une grammaire G donnée, on chercher à :

Trouver le plus petit type de G au sens de l'inclusion.

Classification des grammaires

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

$A \rightarrow \alpha$ avec **$A \in N$** et $\alpha \in (T \cup N)^*$

Mais elle est aussi de type 3 car elle est régulière droite.

En effet, 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.

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 **à contexte lié.**
- Les grammaires de type 0 permettent de générer 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.**

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.

Remarque

- 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

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

L_1 est généré par la grammaire $G=(\{a, b\}, \{S\}, S, P)$ 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 $\mathbf{A\in N}$ 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

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

Le langage L est généré par la grammaire $G1=(\{a, b\}, \{S, A, B\}, S, \{ S \rightarrow AB, A \rightarrow aaA/\varepsilon, B \rightarrow bB/ \varepsilon\}$

G1 n'est pas de type 3 car la règle $S \rightarrow AB$ n'est ni régulière droite ni régulière gauche.

G1 est de type 2 car toutes les règles sont de la forme

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

L2 est donc de type 2 car il est généré par G1 qui est de type 2.

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

EXEMPLES CLASSIQUES DE LANGAGES

Soit la grammaire G_2=({a, b}, {S, B}, S, P_2) où

P_2= { S→aaS/B, B →bB/ ε}).

G_2 est une grammaire de type 3. En effet, elle est régulière droite. Toutes les règles sont de la forme :

A→wB ou A→w avec **A,B∈N** et w∈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 $L=\{a^n b^m / n, m \geq 0\}$.

Une grammaire de type 3 qui engendre L est :

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

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

Une grammaire de type 2 qui engendre L est :

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

Exemples classiques de langages

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

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 4ème 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.

CLASSIFICATION DES LANGAGES

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

- Les langages **réguliers** sont reconnus par des **automates d'états finis.**
- Les langages algébriques sont reconnus par des **automates à piles.**
- Les 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.**