Les Notions Fondamentales de la théorie des langages

Samia Mazouz

Département Informatique

USTHB 2010-2011

Introduction

Les modèles mathématiques de la théorie des langages sont :

- **Grammaire** : permet **d'engendrer** les mots du langage, **généralement infini,** en utilisant un ensemble **fini** de **règles.**
- **Automate** : permet de **reconnaître** les mots d'un langage. Etant donné un mot fourni en entrée, l'automate lit les symboles du mot un par un et va d'état en état selon les transitions. Le mot lu est soit accepté par l'automate soit rejeté.

La théorie des langages établit des correspondances entre descriptions **analytiques** et **génératives.**

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

X= {0, 1}

X = {¬, ∧, (,), p, q}

X = {*, +, (,), Nbre}

X= {IDF, Nbre, op, Cste, =}

Mots

Définition (Mot)

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

Exemples

Alphabet	Mots
{0, 1}	0, 10, 0101, 111
{¬, ∧, (,), p, q}	p ∧q, ¬p, (q ∧¬p) , ¬ ∧, (pq¬ ∧p
{*, +, (,), Nbre}	Nbre+Nbre, **+(,** *Nbre, Nbre++Nbre

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***.
- $\mathbf{X}^+$ est l'ensemble des **mots non vides**. On a $X^*=X^+\cup\{\varepsilon\}$.

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 $w_1.w_2$. Formellement, $w_1.w_2$ est définie comme suit :

$\varepsilon.w_2 = w_2$

$w_1.\varepsilon = w_1$

$w_1w_2 = (a_1 \ldots a_n).(b_1 \ldots b_m) = a_1 \ldots a_n\, b_1 \ldots b_m$

Remarque Pour tout mot w de X* on a : $\varepsilon.w = w.\varepsilon = w$.

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 :

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

Miroir

Définition (Miroir) : 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, on a :

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

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

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 (sous-mot)

Définition (Factorisation) Soient v et w deux mots de X*.

- v est **facteur ou sous-mot** 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 ssi **$u_1 \neq \varepsilon$ et $u_2 \neq \varepsilon$** .
- Le mot v est **facteur gauche** de u si **$u_1 = \varepsilon$.**
- Le mot v est **facteur droit** si **$u_2 = \varepsilon$**.

Exemples Soit le mot w= aabbba, nous avons :

Le mot v1= abb est sous-mot de w, c'est un facteur propre

Le mot v2= aab est facteur gauche de w

Le mot v3=ba est facteur gauche 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. 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)

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

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 $T \cap N = \varnothing$, 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.
- $S \in N$ 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 $\alpha \in (T \cup N)^* N (T \cup N)^*$ (ie α contient au moins un non-terminal) et $\beta \in (T \cup N)^*$.

  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→ aS, S→ε }

Intuitivement, cette grammaire permet de générer les mots a, a^2, a^3,… ainsi que le mot vide ε ie le langage $\{a^n / n \geq 0\}$.

Grammaire

Notations

Plusieurs règles ayant même membre gauche seront notées en écrivant à droite du symbole $\rightarrow$ les différents membres droits séparés par /.

Exemple

$A \rightarrow aB$

$A \rightarrow bAa$

$A \rightarrow a$

On notera $A \rightarrow aB / bAa / a$

Grammaire

Définition (Dérivation directe)

Soit G=(T, N, S, P) une grammaire. Un mot $m_1 \in (T \cup N)+$ **dérive (ou produit) directement** un mot $m_2 \in (T \cup N)^*$ si et seulement si il existe **une production $\alpha \rightarrow \beta$** dans P telle que **$m_1 = u\alpha v$** et **$m_2 = u\beta v$** avec $u,v \in (T \cup N)^*$. On écrit alors **$m_1 \Rightarrow^{(1)} m_2$.**

Exemple Soit G=({0, 1}, {S}, S, {S→0S1/ε})

0S1 dérive directement de S : $S \Rightarrow 0S1$

ε dérive directement de S : $S \Rightarrow \varepsilon$

Grammaire

Définition (Dérivation indirecte)

Soit G = (T, N, S, P) une grammaire. Un mot $w_1 \in (T \cup N)+$ **dérive indirectement** (ou simplement dérve) un mot $w_2 \in (T \cup N)^*$ si et seulement si **w_2 peut êre obtenu par une succession de zéro, un ou plusieurs dérivations directes à partir de w_2.**

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

Exemple

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.

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

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

Le langage généré par G contient exactement les mots dérivables à partir de l'axiome et ne contenant que des symboles terminaux.

Grammaire

Définition (Grammaires équivalentes)

Deux grammaires G et G' sont dites équivalentes, noté $G \equiv G'$, si elles engendrent le même langage.

$$G \equiv G' \Leftrightarrow L(G)=L(G')$$

Classification des grammaires

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

Type 3 (Grammaires régulières)

Si toutes les productions dans P sont de la forme : $\mathbf{A \rightarrow wB}$ **ou** $\mathbf{A \rightarrow w}$ avec $A, B \in N$ et $w \in T^*$

Type 2 (Grammaires à contexte libre ou grammaire algébrique)

Si toutes les productions de P sont de la forme : $A \rightarrow \alpha$ avec $A \in N$ et $\alpha \in (T \cup N)^*$

Classification des grammaires

Type 1 (Grammaires à contexte lié)

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

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

et seul l'axiome peut générer le mot vide ε et dans ce cas S n'apparaît pas en partie droite d'une autre règle.

Remarque : Une grammaire monotone est une grammaire dont 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. Les grammaires monotones sont aussi de type 1.

Grammaire

Type 0 (Grammaire sans restriction) :

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

Remarque

Il existe une relation d'inclusion entre ces quatre types de grammaires. Autrement dit, on type 3 $\subseteq$ type 2 $\subseteq$ type 1$\subseteq$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 décidables**. Autrement dit, tous les langages qui peuvent être reconnus en un temps fini par une machine.

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**.

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

Une grammaire de type 1 qui engendre L est :

G = ({a,b},{S, R, T}, S, P) où P est défini par

$S \rightarrow aRbc / abc / \varepsilon$

$R \rightarrow aRTb / aTb$

$Tb \rightarrow bT$

$Tc \rightarrow cc$

Classification des langages

Enfin, à **chaque de langage** est associé u**n type d'automat**e 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 pars des **automates à bornes linéaires**
- Les langages **de type 0** sont reconnus par des **machines de Turing.**