Les langages algébriques

Samia Mazouz
Département Informatique
USTHB
2013-2014

Plan

I. Introduction
II. Les grammaires algébriques
III. Les automates à pile
IV. Grammaire algébrique et automate à pile

Introduction

Les langages réguliers :

- Sont simples
- Possèdent les bonnes propriétés requises
- Mais pouvoir d'expression faible

Besoin de langages plus expressifs

LANGAGES ALGEBRIQUES

Grammaires algébriques

- Proposées par Chomsky en 1956 pour la description des langages naturels
- Exploitées pour la description des langages de programmation

Définition 1: Une grammaire G=(T, N, S, P) est une grammaire **algébrique** (on dit aussi à contexte libre) si et seulement si toutes ses règles sont de la forme $\mathbf{A \rightarrow \alpha}$ avec $\mathbf{A \in N}$ et $\alpha \in (T \cup N)^*$.

Exemple : La grammaire G=({a, b}, {S}, S, {$S \rightarrow aSb/\varepsilon$} est algébrique.

Grammaires algébriques

Définition 2 : Soit X un alphabet et L un langage sur X^*. Le langage L est dit algébrique s'il existe une grammaire algébrique G telle que L(G)=L.

Exemple : Le langage $L=\{w \in \{a, b\}^* / \ w^R=W\}$ (l'ensemble des mots palindromes sur l'alphabet {a, b}) est algébrique car il est engendré par la grammaire algébrique $G =(\{a, b\}, \{S\} , S, \{S \rightarrow aSa/bSb/a/b/\varepsilon\})$

Remarque : Tout langage régulier est algébrique mais la réciproque est fausse. En effet, le langage $\{a^n b^n / n \geq 0\}$ est algébrique mais non régulier.

Grammaires algébriques

Définition 3 : On dit qu'un mot s'obtient par **dérivation gauche** (respectivement par dérivation droite) s'il est obtenu à partir de l'axiome en **dérivant toujours le non-terminal <u>le plus à gauche</u>** (respectivement le plus à droite).

Exemple : Soit G=({a, b}, {S, A, B}, S, P) avec P :

$S \to bA \,/\, aB \,/\, \varepsilon \quad A \to aS \,/\, bAA \quad B \to bS \,/\, aBB$

Soit le mot = aababb

Une dérivation gauche de ce mot : $S \Rightarrow a\mathbf{B} \Rightarrow aa\mathbf{B}B \Rightarrow aab\mathbf{S}B \Rightarrow aaba\mathbf{B}B \Rightarrow aabab\mathbf{S}B \Rightarrow aabab\mathbf{B} \Rightarrow aababb\mathbf{S} \Rightarrow aababb$

Une dérivation droite de ce mot : $S \Rightarrow a\mathbf{B} \Rightarrow aaB\mathbf{B} \Rightarrow aaBb\mathbf{S} \Rightarrow aa\mathbf{B}b \Rightarrow aab\mathbf{S}b \Rightarrow aaba\mathbf{B}b \Rightarrow aabab\mathbf{S}b \Rightarrow aababb$

Grammaires algébriques

Définition : Soit une grammaire G=(T, N, S, P). Un **arbre de dérivation** d'un mot w appartenant à L(G) est un arbre dont :

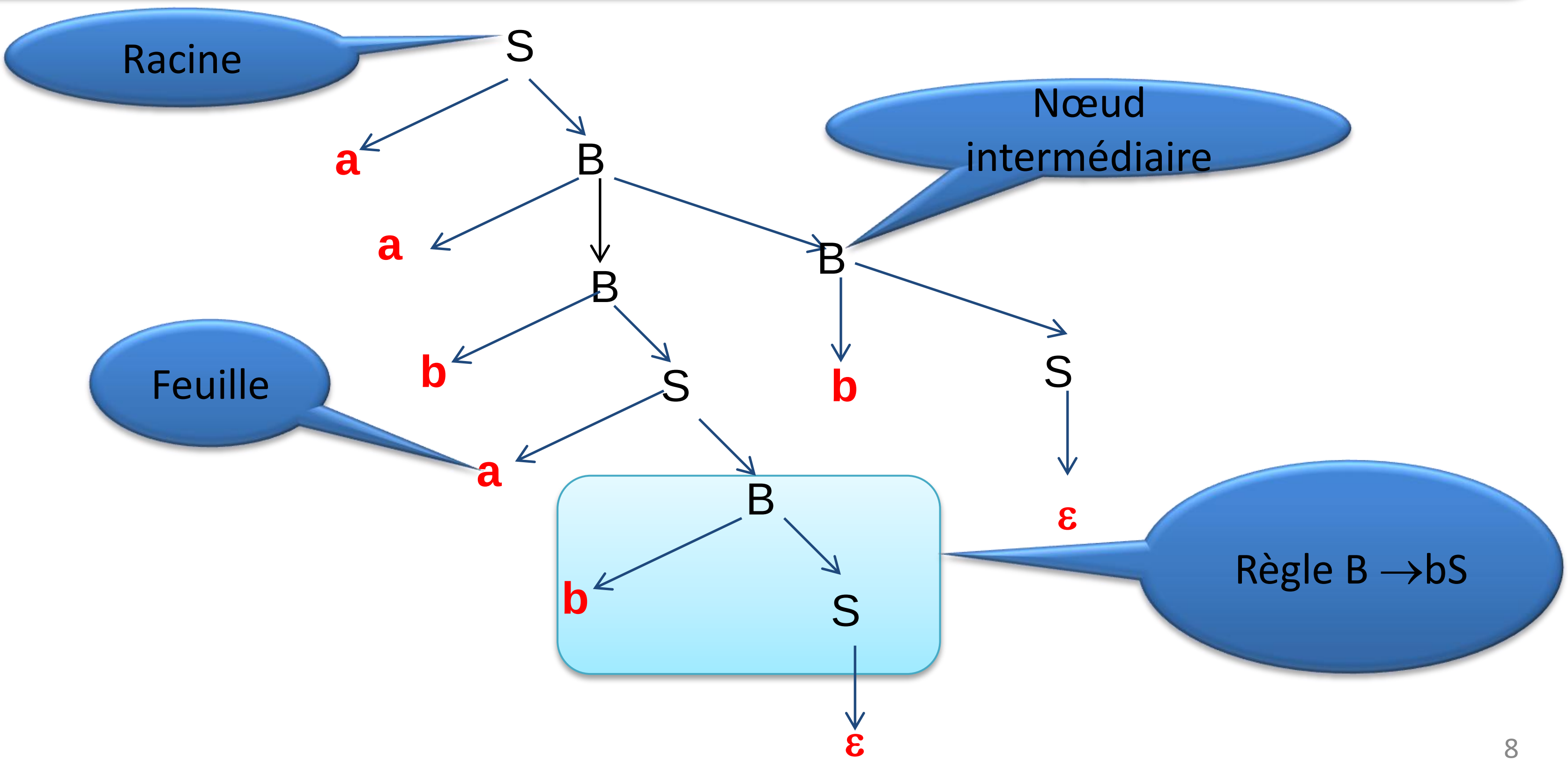
- La racine est étiquetée par l'axiome
- Les feuilles sont étiquetées par des terminaux ou le mot vide
- Les nœuds intermédiaires sont étiquetés par des non-terminaux
- Le passage d'un nœud interne à ses fils correspond à une règle : si le nœud interne est étiqueté par A et ses fils sont étiquetés par X1, …Xn (de gauche vers la droite) alors $A \rightarrow$ X1….Xn est une règle de P
- La lecture des étiquettes des feuilles de gauche à droite donne le mot w.

Grammaires algébriques

Exemple Soit G=({a, b}, {S, A, B}, S, P) avec P :

S→ bA / aB /ε A → aS / bAA B → bS / aBB

Arbre de dérivation du mot =aababb



Grammaires algébriques

Définition :

- Un **mot est ambiguë** s'il est engendré par deux dérivations gauches différentes (ou deux dérivations droites différentes). On peut donc construire deux arbres de dérivations distincts pour ce mot.

- Une **grammaire est ambiguë** dés lors qu'elle possède un mot ambiguë.

Exemple La grammaire donnée précédemment est ambiguë car le mot aababb est ambiguë. Il possède deux dérivations gauches différentes.

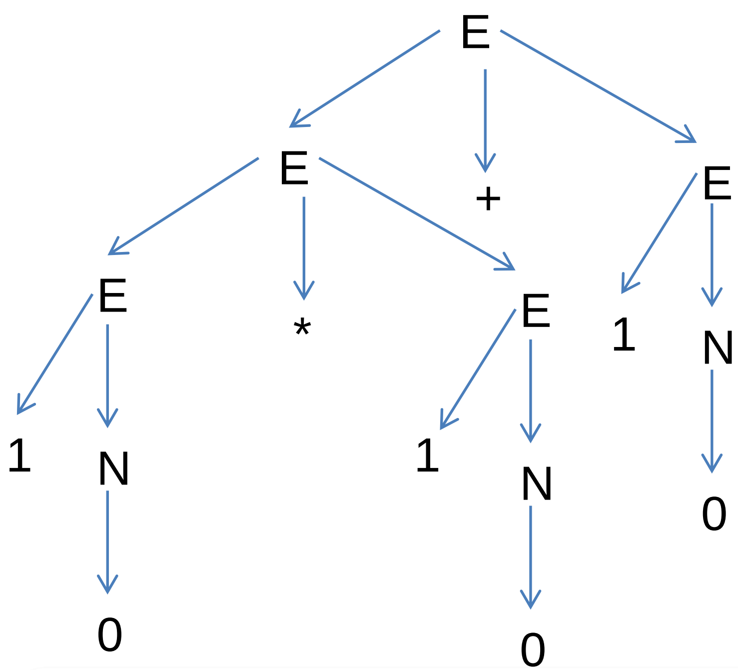
Grammaires algébriques

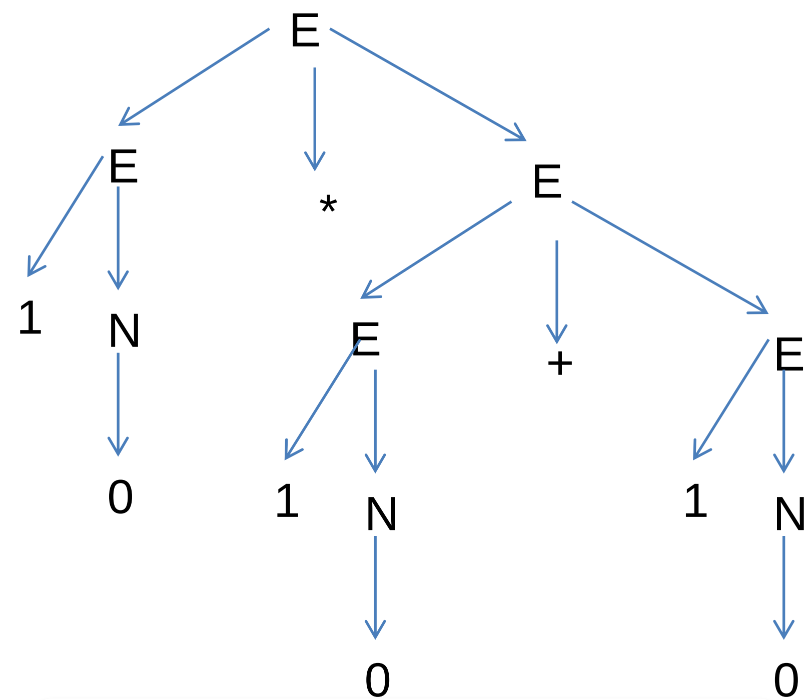
Soit la grammaire G=({+, *, 0, ..,9}, {E, N}, E, P) où P :

$E \rightarrow E + E \,/\, E * E / (E) / N$

$N \rightarrow 0/\ldots./9/\ 0N/\ldots./\ 9N$

Le mot 10*10+10 possède deux arbres de dérivations différents





La valeur calculée est 110

La valeur calculée est 200

Grammaires algébriques

Remarque Pour un même langage, certaines grammaires sont ambiguës, d'autres non.

Définition : On dit qu'un langage algébrique a une ambiguïté inhérente si toute grammaire qui l'engendre est ambiguë.

Remarque Il n'existe pas d'algorithme qui permet de répondre à la question : est-ce que G est ambiguë?

Néanmoins, il est possible de répondre à cette question pour certaines types de grammaires

Automates à pile

Les automates à pile sont les reconnaisseurs des langages algébriques.

Un automate à pile est un automate fini :

- autorisant les ε-transitions,
- avec la particularité de disposer d'une mémoire auxiliaire comme unité de stockage à savoir une pile.

Automates à pile

Définition : Un automate à pile est un septuplé $P=(\Sigma, \Gamma, Z_0, Q, q_0, F, \delta)$ où :

- Σ est un ensemble fini de l'alphabet d'entrée
- Γ est un ensemble fini de l'alphabet auxiliaire. $Z_0 \in \Gamma$ est le symbole initial de la pile. $\Sigma \cup \Gamma$ constitue l'alphabet du langage de la pile.
- Q est un ensemble fini d'états et $q_0 \in Q$ est l'état initial
- $F \subseteq Q$ est l'ensemble des états finaux
- δ est une fonction de transition,

  qui prend comme argument un triplet (u, q, a) où :
  - ✓ u est le sommet de la pile
  - ✓ q est un état de Q
  - ✓ a est un symbole d'entrée de Σ ou le mot vide.

  qui retourne une paire (α, p) où α est un mot sur $(\Sigma \cup \Gamma)^*$ et q le nouvel état tq:
  - ✓ Si $\alpha=v_1 \ldots v_n$ où $v_i \in \Sigma \cup \Gamma$ alors le sommet de pile est dépilé ensuite on empile v_1, …. et v_n dans cet ordre. Le sommet de pile contiendra v_n.
  - ✓ Si $\alpha=$ u alors la pile est inchangée
  - ✓ Si $\alpha=\varepsilon$ alors on dépile le sommet de pile

Notation : La transition $\delta(u, q, a)= (\alpha, p)$ sera notée $uqa \rightarrow \alpha q$

Automates à pile

Exemple :

Un automate à pile $P=(\Sigma, \Gamma, Z_0, Q, q_0, F, \delta)$ reconnaissant $\{a^n b^n / n \geq 0\}$ est tel que :

$\Sigma=\{a, b\}$, $\Gamma=\{Z_0\}$, $Q=\{q_0, q_1, q_2\}$, $F=\{q_2\}$ et δ est définie par :

N°	Instruction	Commentaire
1	$Z_0q_0a \rightarrow Z_0aq_0$	empiler le 1er a
2	$aq_0a \rightarrow aaq_0$	empiler tous les a suivants
3	$aq_0b \rightarrow q_1$	dépiler le dernier a empilé à la rencontre du 1er b et on change d'état pour ne plus autoriser la lecture de a
4	$aq_1b \rightarrow q_1$	dépiler un a à chaque lecture d'un b
5	$Z_0q_1 \rightarrow q_2$	le nombre de a qui a été empilé dans l'état q_0 est égal au nombre de b lu dans l'état q_1 et on change d'état
6	$Z_0q_0 \rightarrow Z_0q_2$	la reconnaissance du mot vide

Automates à pile

Les étapes de la reconnaissance du mot aaabbb sont données dans la tableau suivant :

Pile	État	Sous-mot d'entrée	N° Transition
Z_0	q_0	aaabbb	1
Z_0a	q_0	aabbb	2
Z_0aa	q_0	abbb	2
Z_0aaa	q_0	bbb	3
Z_0aa	q_1	bb	4
Z_0a	q_1	b	4
Z_0	q_1		5
Z_0	q_2		

Automates à pile

Définition : Si un automate à pile P est dans l'état q, le contenu de sa pile est $\alpha \in (\Gamma \cup \Sigma)^*$ et il lui reste à lire le sous-mot w ($w \in \Sigma^*$), on dira que cet automate se trouve dans la configuration (α, q, w).

Si le mot à reconnaître est w alors la configuration initiale est (Z_0, q_0, w)

Définition : Une configuration (αu, q, aw) ($u \in \Sigma \cup \Gamma$, $a \in \Sigma \cup \{\varepsilon\}$) dérive directement ($\alpha\beta$, p, w), notée ($\alpha$u, q, aw) $\Rightarrow$ ($\alpha\beta$, p, w) ,

si upa$\rightarrow\beta$q est une transition de l'automate.

La fermeture réflexive et transitive de la relation $\Rightarrow$ est notée $\Rightarrow^*$. On parle de dérivation indirecte, qui correspond à zéro ou une ou plusieurs dérivations directes consécutives.

Automates à pile

Deux modes de reconnaissances :

- par état final
- par pile vide

Reconnaissance par état final

Définition : Soit $P=(\Sigma, \Gamma, Z_0, Q, q_0, F, \delta)$ un automate à pile. Le langage accepté par P par état final , noté L(P), est : $\{ w \mathbin{/} \exists \alpha \in (\Sigma \cup \Gamma)^* \ \exists\, q_F \in F, (Z_0, q_0, w) \Rightarrow^* (\alpha, q_F, \varepsilon) \}$. **Le contenu de la pile n'a pas d'importance.**

Exemple Soit l'automate à pile P1=$(\{a, b\}, \{Z_0\}, Z_0, \{q_0, q_1\}, q_0,, \{q_1\}, \delta_1)$:

$\delta_1=\{Z_0q_0a \to Z_0aq_0,\ aq_0a \to aaq_0,\ aq_0b \to q_1,\ aq_1b \to q_1\}$

Cet automate reconnaît par état final le langage $\{a^nb^m / n \geq m\}$

Automates à pile

Reconnaissance par pile vide

Définition : Soit $P=(\Sigma, \Gamma, Z_0, Q, q_0, F, \delta)$ un automate à pile. Le langage accepté par P par pile vide, noté $L_\varepsilon(P)$, est : $\{ w / \exists\, q, (Z_0, q_0, w) \Rightarrow^* (\varepsilon, q, \varepsilon)\}$.

Remarque Un mot est accepté si après lecture du mot la pile ne contient aucun symbole y compris Z_0.

Exemple

Soit l'automate à pile $P2=(\{a, b\}, Z_0,\{Z_0\}, \{q_0, q_1\}, q_0, \emptyset, \delta)$:

$\delta_2 =\{Z_0q_0a \rightarrow Z_0aq_0,\ aq_0a \rightarrow aaq_0,\ aq_0c \rightarrow aq_1,\ aq_1b \rightarrow q_1,\ Z_0q_1 \rightarrow q_1,\ Z_0q_0c \rightarrow q_0,\}$

Cet automate reconnaît par pile vide le langage $\{a^ncb^n/ n \geq 0\}$.

Si on considère que l'état q_1 est final, le langage reconnu par état final est alors $\{a^ncb^m/ n>m \geq 0\}$

Automates à pile

Equivalence des deux modes de reconnaissance :

Les deux modes de reconnaissance permettent de reconnaître les mêmes langages.

tout langage reconnu par un automate à pile par pile vide est également reconnu par un automate à pile par état final et vice-versa.

Grammaire algébrique et automate à pile

- Les langages reconnus par les automates à pile correspondent exactement à la classe des langages algébriques.

- La classe des langages algébriques correspond à la classe des langages reconnus par les automates à pile.