Université de Rouen	Année en cours
Faculté des sciences	Théorie des langages
Licence Informatique 3ème année	Résumé de cours pour le TD n° 8

Les langages algèbriques et les grammaires algébriques

1 Introduction

Nous allons maintenant étudier une classe particulière de langages : les langages algébriques et un outil permettant de les engendrer, les grammaires algébriques. Les grammaires algébriques sont utilisées pour spécifier la syntaxe des langages de programmation et sont particulièrement adaptées à la description des expressions arithmétiques, des structures de bloc ou des expressions bien parenthèsées. Elles jouent un rôle primordial dans la réalisation de compilateurs.

2 Grammaires algébriques et la dérivation

Une grammaire algébrique est un quadruplet (N, T, S, R) où :
- T est un ensemble fini non vide de symboles appelé ensemble de symboles terminaux ou alphabet terminal ;
- N est un ensemble fini de symboles appelé ensemble de symboles non-terminaux ou alphabet non-terminal qui vérifie $T \cap N = \emptyset$;
- S est un symbole faisant partie de N appelé axiome ;
- R est l'ensemble fini des productions. C'est une partie finie de $N \times (N \cup T)^*$. Il est important de noter ici que la partie droite de chaque règle d'une grammaire algébrique est composée d'un unique non-terminal.

Voici trois jeux de règles qui déterminent chacun une grammaire algébrique :

G_1 : $S \longrightarrow aAba$
$A \longrightarrow \varepsilon$
$A \longrightarrow aA$

G_2 : $S \longrightarrow aSbS \mid \varepsilon$

G_3 : Affectation $\longrightarrow$ Identificateur := Expression
Expression $\longrightarrow$ Expression + Expression | Expression × Expression | (Expression) | Identificateur

Une grammaire algébrique est utilisée pour engendrer les mots d'un langage en appliquant une suite de règles : axiome $S \longrightarrow$ application successive de productions $\longrightarrow$ mots de T^*. L'application successive des règles s'appelle une **dérivation** et les mots obtenus les **mots dérivés**. Formellement, on dit que v se dérive en ν

- **directement** ou en une étape et on note $v \Rightarrow \nu$, $v, \nu \in (N \cup T)^*$ si $\exists \alpha, \beta, \gamma \in (N \cup T)^*$ et $A \in N$ tels que $A \longrightarrow \beta \in R$, $v = \alpha A \gamma$ et $\nu = \alpha\beta\gamma$ c'est à dire que $\alpha A \gamma \Rightarrow \alpha\beta\gamma$ en appliquant la règle $A \longrightarrow \beta$.

- **en** n **étapes** et on note $v \overset{n}{\Rightarrow} \nu$ si $\exists v_0, v_1, \ldots, v_n \in (N \cup T)^*$ tels que $v = v_0 \Rightarrow v_1 \Rightarrow \ldots \Rightarrow v_n = \nu$

De plus, on note

- $v \overset{0}{\Rightarrow} \nu$ si $v = \nu$,
- $v \overset{*}{\Rightarrow} \nu$ si $\exists n \geq 0$ tel que $v \overset{n}{\Rightarrow} \nu$,
- $v \overset{+}{\Rightarrow} \nu$ si $\exists n > 0$ tel que $v \overset{n}{\Rightarrow} \nu$.

Le langage engendré par une grammaire algébrique G relativement à l'axiome S est $L(G) = \{w \in T^* \mid S \overset{*}{\Rightarrow} w\}$.

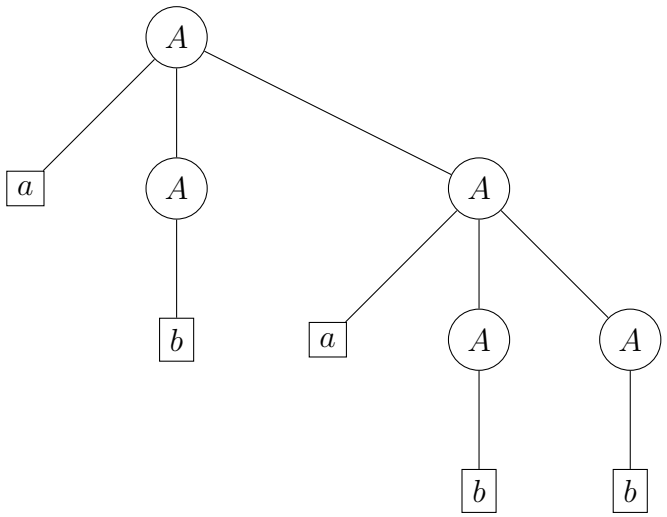
3 Langages algébriques

Un langage engendré par une grammaire algébrique est appelé langage algébrique. C'est l'ensemble des mots de T^* qui dérivent de l'axiome S. On note $\text{Alg}(T^*)$ l'ensemble des langages algébriques sur T, c'est à dire l'ensemble des langages pour lesquels il existe une grammaire qui les engendre.

Exemple 1 : Considérons l'ensemble des expressions arithmétiques en notation préfixe utilisant les opérateurs binaires $+$ et $\cdot$ et un opérande b. Par exemple, $+ \cdot b + bbb$ est l'?expression préfixe correspondant à l'expression $b \cdot (b + b) + b$ en notation infixe. Notons maintenant les opérateurs par la lettre a. L'ensemble des expressions arithmétiques est alors un langage sur l'alphabet $T = \{a, b\}$ appelé **langage de Lukasiewicz**. On a $L = \{b, abb, aabbb, ababb, aababbb, \ldots\}$. Ce langage est un langage algébrique, il peut être obtenu à partir de la grammaire algébrique suivante :

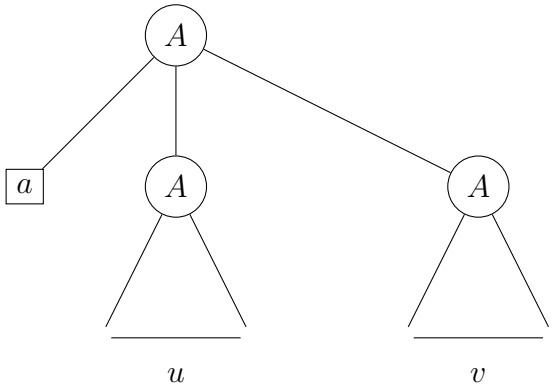
$G : \quad A \longrightarrow b \mid aAA$

Par exemple, le mot $ababb$ est obtenu par la dérivation : $A \Rightarrow aAA \Rightarrow abA \Rightarrow abaAA \Rightarrow ababA \Rightarrow ababb$. On peut aussi représenter cette dérivation sous forme d'un arbre appelé arbre de dérivation :



Les feuilles de l'arbre (ou **frontière** de l'arbre) lues de gauche à droite donnent le résultat de la dérivation soit $ababb$.

Si on considère la première règle appliquée dans une dérivation partant de l'axiome A, ce ne peut être que $A \Rightarrow aAA$ qui correspond à l'arbre de dérivation

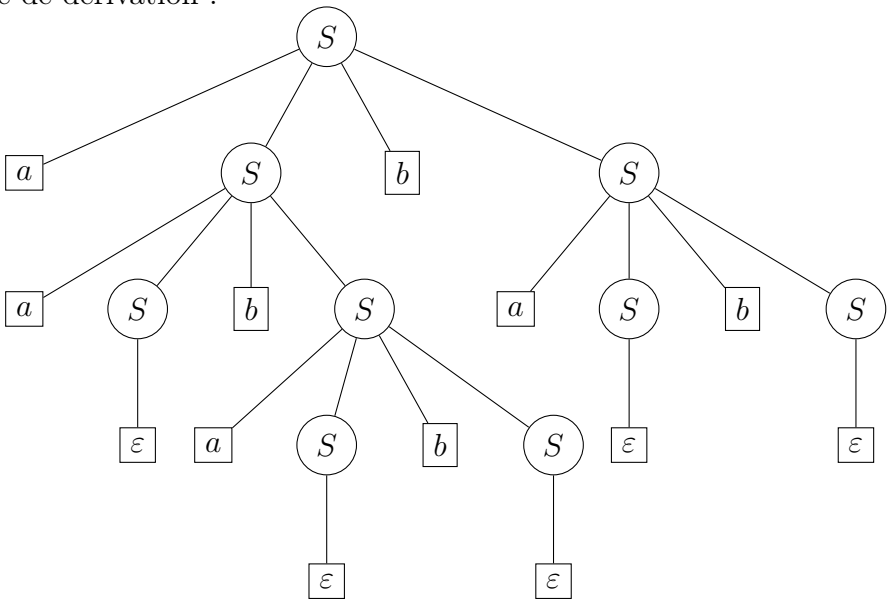


dans lequel u et v, mots de T^*, dérivent de A et donc $u, v \in L(G)$ ou $A \Rightarrow b$ qui correspond à l'arbre de dérivation suivant.

On en déduit que $L(G) = \{w \in \Sigma^* \mid w = b \text{ ou } w = auv, u, v \in L(G)\}$

Exemple 2 : On considère la grammaire définie par les règles : $S \longrightarrow aSbS \mid \varepsilon$. Le langage engendré est appelé **langage de Dyck restreint** et noté D'_1*. C'est l'ensemble des mots obtenus à partir des expressions bien parenthèsées en notant « (»par la lettre a et «) »par la lettre b et en oubliant tout autre caractère. Par exemple le mot $aababbab$ est dans D'_1*. Il est l'image de $(()())()$ et est obtenu par la dérivation :
$S \Rightarrow aSbS \Rightarrow aaSbSbS \Rightarrow aa\varepsilon bSbS = aabSbS \Rightarrow aabaSbSbS \Rightarrow aaba\varepsilon bSbS \Rightarrow aababSbS \Rightarrow aabab\varepsilon bS = aababbS \Rightarrow aababbaSbS \Rightarrow aababba\varepsilon bS = aababbabS \Rightarrow aababbab\varepsilon = aababbab$

$L(G) = \{w \in \Sigma^* \mid w = \varepsilon \text{ ou } w = aubv, u, v \in L(G)\}$.
On a l'arbre de dérivation :



4 Ambiguïté

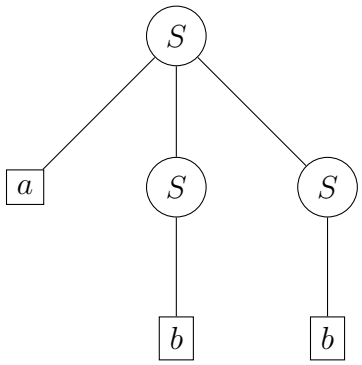
Deux grammaires sont dites équivalentes si elles engendrent le même langage. Nous en verrons des exemples par la suite. Deux dérivations pour une grammaire donnée sont équivalentes si leurs arbres de dérivation sont identiques.

Exemple : Soit la grammaire G définie par les règles :$S \longrightarrow b \mid aSS$. Le mot abb peut s'obtenir par les dérivations suivantes :

$S \Rightarrow aSS \Rightarrow abS \Rightarrow abb$ (1)

$S \Rightarrow aSS \Rightarrow aSb \Rightarrow abb$ (2)

Dans les deux cas l'arbre de dérivation est :



On appelle **dérivation la plus à gauche** une dérivation dans laquelle on dérive systématiquement le non-terminal le plus à gauche. La dérivation (1) est une dérivation la plus à gauche. On appelle **dérivation la plus à droite** une dérivation dans laquelle on dérive systématiquement le non-terminal le plus à droite. La dérivation (2) est une dérivation la plus à droite.

Soit G une grammaire algébrique et $L(G)$ le langage engendré par G. Si on peut construire deux arbres de dérivations distincts pour un mot w de $L(G)$, on dit que la grammaire G est **ambiguë**.