**NOM :** ROUHLING **Prénom :** Damien **Jury :**

**Algèbre** ← Entourez l'épreuve → **Analyse**

910.

**Sujet choisi :** Langages algébriques. Exemples et applications.

**Autre sujet :**


Réf : Sipser — Floyd, Biegel, Le langage des machines
Hopcroft, Ullman, Introduction to automata theory, languages and computation
Autebert, Théorie des langages et des automates
Carton, Langages formels


# I les grammaires hors-contexte

## 1_ Définition

**Déf 1 :** Une grammaire (hors-contexte) est un quadruplet $(V,T,S,P)$ où $V$ est un ensemble de variables, $T$ un ensemble de symboles "terminaux" avec $T \cap V = \emptyset$, $S \in V$ appelée variable de départ et $P$ un ensemble de règles ou productions de la forme $A \to \alpha$ avec $A \in V$ et $\alpha \in (V \cup T)^*$

**Notation 1 :** Si $P$ contient $A \to \alpha_1, \ldots, A \to \alpha_n$, $n \geq 2$, on note le tout comme une seule règle $A \to \alpha_1 | \ldots | \alpha_n$

On se contente en général d'écrire les variables en majuscules et les terminaux en minuscules et de ne donner que l'ensemble des productions

**Ex 1 :** $E \to E+E \mid E*E \mid (E) \mid 1$

**Déf 2 :** Soit $G=(V,T,S,P)$ une grammaire et $(u,v) \in (V \cup T)^*$; $u$ se dérive en $v$, noté $u \to v$, ssi $\exists \alpha, \beta \in (V \cup T)^*$ et $X \to w \in P$ tel que $u = \alpha X \beta$ et $v = \alpha w \beta$

Le langage engendré par $G$ est $L(G) = \{u \in T^*, S \to^* u\}$

**Ex 2 :** pour la grammaire de l'exemple 1 :

$E \to E+E \to E+E*E \to 1+E*E \to 1+1*E \to 1+1*1$

**Déf 3 :** On appelle langage algébrique tout langage engendré par une grammaire

**Déf 4 :** Soit $G=(V,T,S,P)$ une grammaire

Un arbre de dérivation pour $G$ est un arbre étiqueté aux nœuds vérifiant :

(1) les étiquettes appartiennent à $V \cup T \cup \{\varepsilon\}$

(2) La racine est étiquetée par $S$

(3) Si un nœud a pour étiquette $A$ et pour fils des nœuds étiquetés $\alpha_1 \ldots \alpha_k$ dans l'ordre, alors $A \to \alpha_1 \ldots \alpha_k \in P$

Si $a_1 \ldots a_n$ sont les étiquettes des feuilles (de gauche à droite) d'un arbre de dérivation, alors $a = a_1 \ldots a_n$ est le mot produit par l'arbre

**Ex 3 :** Figure 1

**Thm 1 :** Soit $G=(V,T,S,P)$ une grammaire et $u \in T^*$

$S \to^* u$ ssi $u$ est produit par un arbre de dérivation

## 2_ Pourquoi algébrique ?

**Rq 1 :** On peut implicitement définir des langages par une équation algébrique, ie n'utilisant que les opérations d'union et de concaténation

**Ex 4 :** $X = \# \cup aXa \cup bXb$ $(*)$

**Prop 1 :** L'ajout de l'étoile dans le membre droit des équations ne change pas la classe de langages définie

**Rq 2 :** On peut ainsi définir des langages non rationnels, c'est le cas de toute solution de $(*)$.

**Thm 2 :** Si $X = f(X)$ est une équation algébrique, elle admet pour solution minimale unique le plus petit point fixe de $f$

**Cor 1 :** Tout système d'équations algébriques admet une solution minimale unique

**Rq 3 :** On peut transformer tout système d'équations algébriques en grammaire en développant le membre droit et en cassant les unions (ex : Figure 2), et réciproquement.

**Thm 3 :** La solution minimale d'un système d'équations algébriques est l'ensemble des mots admettant un arbre de dérivation pour la grammaire associée.

# II Propriétés des grammaires et des langages algébriques

## 1 _ Existence de langages non algébriques

**Thm 4 (Lemme d'Ogden)** : Soit $G = (V, T, S, P)$ une grammaire. Il existe $N \in \mathbb{N}$ tel que $\forall z \in L(G)$, si l'on marque au moins $N$ lettres de $z$, alors il existe $A \in V$ et $u, v, w, x, y \in T^*$ tels que $z = uvwxy$, il y a au moins $N$ lettres marquées dans $vwx$, $S \to^* uAy$, $A \to^* vAx$ et $A \to^* w$.

**Application 1** : $\{a^i b^j c^k,\ i \neq j \text{ et } j = k\}$ n'est pas algébrique

**Rq 4** : Une version affaiblie du théorème, sans marquage de lettres, permet de montrer que $\{a^n b^n c^n, n \geq 0\}$ et $\{a^i b^j c^i d^j,\ i, j \in \mathbb{N}\}$ ne sont pas algébriques

## 2 _ Ambiguïté et forme normale de Chomsky

**Déf 5** : Une grammaire est ambigüe s'il existe un mot admettant plusieurs dérivations
Un langage algébrique est intrinsèquement ambigu si toute grammaire l'engendrant est ambigüe

**Ex 5** : Figure 3

**Thm 5** : $\{a^i b^j c^k,\ i = j \text{ ou } j = k\}$ est algébrique et intrinsèquement ambigu

**Thm 6** : Tout langage algébrique est reconnu par une grammaire sous forme normale de Chomsky, ie n'ayant que des productions de la forme $\begin{cases} A \to BC \text{ pour } A, B, C \in V \\ A \to a \text{ pour } a \in T,\ A \in V \end{cases}$

**Ex 6** :

$S \to G_1 G_2 \mid D_1 D_2$ $\quad A \to A' G_1$ $\quad C \to c$ $\quad C' \to D_2 C$

$G_1 \to AB \mid \varepsilon$ $\quad A' \to a$ $\quad D_1 \to A' D_1 \mid \varepsilon$

$G_2 \to C G_2 \mid \varepsilon$ $\quad B \to b$ $\quad D_2 \to BC' \mid \varepsilon$

## 3 _ Clôture

**Prop 2** : L'ensemble des langages algébriques est clos par union, concaténation et étoile

**Cor 2** : les langages rationnels sont algébriques.

**Prop 3** : L'ensemble des langages algébriques n'est pas clos par intersection, partant, par complémentation.

**Prop 4** : L'intersection d'un langage rationnel avec un langage algébrique est algébrique.

**Déf 6** : Soit $A$ et $B$ deux alphabets
Un morphisme de $A^*$ vers $B^*$ est une application $\varphi$ vérifiant $\begin{cases} \varphi(\varepsilon) = \varepsilon \\ \forall u, v \in A^*\ \varphi(uv) = \varphi(u)\varphi(v) \end{cases}$
Un morphisme $\varphi$ est alphabétique si $\forall a \in A\ |\varphi(a)| \leq 1$
Une substitution est un morphisme de $A^*$ vers $\mathcal{P}(B^*)$
Une substitution $\varphi$ est algébrique si $\forall a \in A\ \varphi(a)$ est algébrique

**Prop 5** : Si $L$ est algébrique et $\varphi$ est une substitution algébrique alors $\varphi(L)$ est algébrique

**Cor 3** : L'ensemble des langages algébriques est clos par image par un morphisme

**Développement 1**

**Thm 7 (Chomsky-Schützenberger)** : $L$ est algébrique ssi $L = \varphi(D_n^* \cap K)$ pour $n \in \mathbb{N}$, $K$ rationnel et $\varphi$ morphisme alphabétique où $D_n^*$ est le langage de Dyck sur $n$ paires de parenthèses

**Application 2 (transduction rationnelle)** : Si $L$ est algébrique, $L = \varphi(\psi^{-1}(D_2^*) \cap K)$ pour $\varphi$ et $\psi$ morphismes et $K$ rationnel

## 4_ Décidabilité

Thm 8 : Si $G$ est en forme normale de Chomsky et $w \in T^*$

Il existe un algorithme polynomial décidant si $w \in L(G)$

D Algorithme CYK (Cocke, Younger, Kasami)

Rq 5 : On peut faire plus efficace avec l'algorithme d'Earley

Thm 9 : Soit $G$ une grammaire

les problèmes suivants sont décidables : $L(G) = \emptyset$, $L(G)$ est fini, $L(G)$ est infini

Thm 10 : PCP est indécidable

Application 3 : les problèmes suivants sont indécidables :

Si $G$ et $G'$ sont deux grammaires : $L(G) \cap L(G') = \emptyset$, $L(G) = L(G')$

Si $G$ est une grammaire : $L(G) = T^*$, $G$ est ambigüe

# III les automates à pile

## 1_ Définition et propriétés

Déf 7 : Un automate à pile est un 6-uplet $(Q, \Sigma, \Gamma, \delta, q_0, F)$ où $Q$ est l'ensemble des états, $\Sigma$ l'alphabet d'entrée, $\Gamma$ l'alphabet de pile, $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\}) \to \mathcal{P}(Q \times (\Gamma \cup \{\varepsilon\}))$ la fonction de transition, $q_0 \in Q$ l'état initial, $F \subseteq Q$ l'ensemble des états finaux

La fonction de transition fonctionne ainsi : dans l'état $q$, à la lecture de $a \in \Sigma \cup \{\varepsilon\}$, dépiler $\alpha \in \Gamma \cup \{\varepsilon\}$ et pour chaque $(q', \beta) \in \delta(q, a, \alpha)$, on peut passer dans l'état $q'$ en empilant $\beta$, noté $q \xrightarrow{a\alpha \to \beta} q'$

Si à la fin de la lecture du mot en entrée on peut atteindre un état final, alors le mot est accepté

le langage reconnu par l'automate est l'ensemble de ses mots acceptés

Ex 7 : Figure 4

### Développement 2

Thm 11 : Un langage est algébrique ssi il est reconnu par un automate à pile

Thm 12 : Pour tout automate à pile, le langage des contenus de pile est rationnel.

## 2_ Application à l'analyse syntaxique

Principe de l'analyse syntaxique descendante LL(2) :

Etant donné une grammaire $G = (V, T, S, P)$ et un mot $w \in T^*$, on veut déterminer si $w \in L(G)$. Pour ce faire, on construit $w$ de la gauche vers la droite en devinant à chaque instant la production à utiliser dans une dérivation la plus à gauche, avec connaissance des 2 lettres suivantes à construire

Prop 6 : L'analyse LL(1) est réalisable par un automate à pile

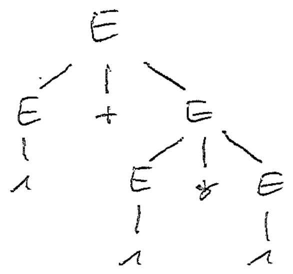
Ex 8 : Pour la grammaire suivante :

$S \to aSbS \mid cT$

$T \to aTbc \mid b$

La Figure 5 donne un automate à pile correspondant et le calcul sur acbbcabbc.

Figure 1 :



produit 1+1*1

Figure 2 :

$L = \varepsilon \cup a(\varepsilon \cup La) \cup b(\varepsilon \cup Lb)$

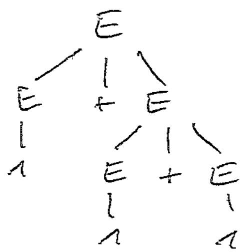
devient

$L = \varepsilon \cup a \cup aLa \cup b \cup bLb$

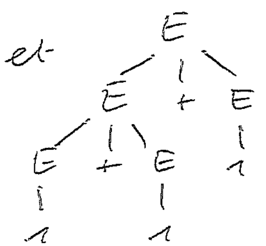
puis

$L \to \varepsilon \mid a \mid b \mid aLa \mid bLb$

Figure 3 :

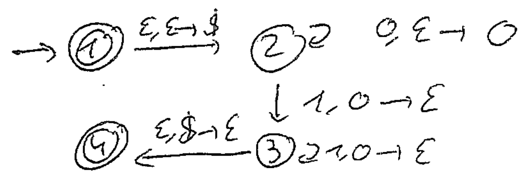


et



produisent 1+1+1

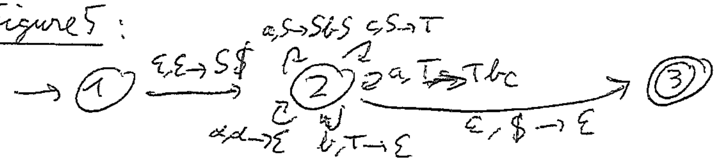
Figure 4 : Automate à pile reconnaissant $\{0^n 1^n, n \in \mathbb{N}\}$



Calcul sur 0011 :

| Position | État | Pile |
|---|---|---|
| .0011 | 1 | ∅ |
| .0011 | 2 | \$ |
| 0.011 | 2 | 0\$ |
| 00.11 | 2 | 00\$ |
| 001.1 | 3 | 0\$ |
| 0011. | 3 | \$ |
| 0011. | 4 | ∅ |

Figure 5 :



Calcul sur acbbcabbc :

| Position | État | Pile |
|---|---|---|
| .acbbcabbc | 1 | ∅ |
| .acbbcabbc | 2 | S\$ |
| a.cbbcabbc | 2 | SbS\$ |
| ac.bbcabbc | 2 | TbS\$ |
| acb.bcabbc | 2 | bS\$ |
| acbb.cabbc | 2 | S\$ |
| acbb.cabbc | 2 | S\$ |
| acbbc.abbc | 2 | T\$ |
| acbbca.bbc | 2 | Tbc\$ |
| acbbcab.bc | 2 | bc\$ |
| acbbcabb.c | 2 | c\$ |
| acbbcabbc. | 2 | \$ |
| acbbcabbc. | 3 | ∅ |