

## Cartouche du document

Année : ING 1 - Matière : Théorie des langages - Activité : Travail dirigé

## Objectifs

- Notions de langage hors contexte ou algébrique
- Normalisation de Chomsky
- Algorithme CKY
- Application au traitement des langues

### Définition :

Un langage hors-contexte est aussi appelé langage algébrique.

Une grammaire hors-contexte (ou algébrique) est un quadruplet  $T, N, S, P$  où :

- $T$  : ensemble des éléments terminaux
- $N$  : ensemble des éléments non terminaux
- $S$  : élément non terminal initial (axiome)
- $P$  : ensemble de règles de la forme :
  - $X \rightarrow a$  où  $a \in T$  et  $X \in N$
  - $X \rightarrow Y$  où  $Y \in (N \cup T)^*$  et  $X \in N$

### Définition :

Une grammaire hors-contexte **qui ne produit pas  $\epsilon$**  est dite sous forme normale de Chomsky si et seulement si toutes les règles sont de la forme :

$A \rightarrow a$  où  $a \in T$

$A \rightarrow BC$  où  $B, C \in N$

**On rappelle que  $\epsilon$  (le mot vide) peut être également noté  $\lambda$  (c'est le cas par défaut en JFLAP).**

## Sommaire des exercices

- 1 - Reconnaissance d'un mot par l'algorithme CKY
- 2 - Des propositions très relatives
- 3 - JFLAP

## Corps des exercices

### 1 - Reconnaissance d'un mot par l'algorithme CKY

#### Énoncé :

Il s'agit dans l'exercice suivant de :

- Transformer une grammaire sous Forme Normale de Chomsky.
- Définir un algorithme qui teste l'appartenance d'un mot à cette grammaire normalisée.

#### Question 1)

##### Énoncé de la question

Soit la grammaire  $G$  basée sur l'alphabet  $\{a, b\}$ .

On a :

$G = \{$

$T = \{a, b\}$

$$N = \{S, A, B\}$$

$$S = S$$

$$P = \{$$

$$(1) S \rightarrow b A$$

$$(2) S \rightarrow a B$$

$$(3) A \rightarrow b A A$$

$$(4) A \rightarrow a S$$

$$(5) A \rightarrow a$$

$$(6) B \rightarrow a B B$$

$$(7) B \rightarrow b S$$

$$(8) B \rightarrow b$$

$$\}$$

$$\}$$

Mettre cette grammaire sous forme normalisée de Chomsky.

Procédure de transformation :

- 1. Remplacer tous les terminaux  $x$  en partie droite des règles par des non terminaux en ajoutant des règles de la forme  $X \rightarrow x$
- 2. Transformer les parties droites des règles comme suit :  $X \rightarrow Y Z W$  par deux règles  
 $X \rightarrow Y V$  et  $V \rightarrow Z W$
- 3. Transformer les parties droites des règles comme suit :  $X \rightarrow Y$  par  $X \rightarrow W Z$  si  $Y \rightarrow W Z$

[Solution de la question](#)

### Etape 1

Pour chaque terminal, on crée une règle avec un nouveau terminal comme membre gauche et le terminal comme membre droit.

$$(9') A_1 \rightarrow a$$

$$(10') B_1 \rightarrow b$$

### Etape 2

Pour chaque règle comportant plus de deux membres à droite ((3) et (6)), on scinde la partie droite en créant de nouveaux non terminaux et de nouvelles règles.

$$(11') A_2 \rightarrow A A$$

$$(12') B_2 \rightarrow B B$$

### Etape 3

On normalise toutes les règles et on obtient la grammaire suivante :

(1')  $S \rightarrow B_1 A$

(2')  $S \rightarrow A_1 B$

(3')  $A \rightarrow B A_2$

(4')  $A \rightarrow A_1 S$

(5')  $A \rightarrow a$

(6')  $B \rightarrow A_1 B_2$

(7')  $B \rightarrow B_1 S$

(8')  $B \rightarrow b$

(9')  $A_1 \rightarrow a$

(10')  $B_1 \rightarrow b$

(11')  $A_2 \rightarrow A A$

(12')  $B_2 \rightarrow B B$

## Question 2)

### Énoncé de la question

Montrer que le mot **aabbab** appartient au langage engendré par la grammaire.

On utilisera l'algorithme de Cocke, Younger et Kasami (CKY) qui permet de tester si un mot  $w$  est reconnu par une grammaire sous forme normale de Chomsky.

- On note  $n$  la longueur du mot  $w$  :  $n = |w|$ .
- On définit  $v$  une matrice de dimensions  $(n,n)$ .

Le pseudo-code de l'algorithme CKY est le suivant :

Pour  $i = 1$  à  $n$

DEBUT

$v[i,1] = \{A \text{ tel que } A \text{ est le membre gauche d'une règle } A \rightarrow a \text{ et } a \text{ est le } i^{\text{ème}} \text{ symbole du mot } w\}$

FIN

Pour  $j = 2$  à  $n$

DEBUT

Pour  $i = 1$  à  $n - j + 1$

DEBUT

$v[i,j] = \emptyset$

Pour  $k = 1$  à  $j - 1$

DEBUT

$v[i,j] = v[i,j] \cup \{A \text{ tel que } A \text{ est le membre gauche d'une règle } A \rightarrow B C \text{ avec } B \in v[i,k] \text{ et } C \in v[i+k,j-k]\}$

FIN

FIN

FIN

Le mot  $w$  est reconnu par la grammaire  $\Leftrightarrow S \in v[1,n]$ .

### Solution de la question

Le tableau  $v$  avec le mot **aabbab** est le suivant :

a	A, A <sub>1</sub>	A <sub>2</sub>	A	S	A	S
a	A, A <sub>1</sub>	S	B	S	B	
b	B, B <sub>1</sub>	B <sub>2</sub>	B	B <sub>2</sub>		
b	B, B <sub>1</sub>	S	B			
a	A, A <sub>1</sub>	S				
b	B, B <sub>1</sub>					

Nous constatons que  $S \in v[1,6]$ , donc "aabbab" est un élément du langage.

## 2 - Des propositions très relatives

### Énoncé :

On s'intéresse aux constructions de phrases avec des propositions relatives.

On considère la grammaire hors contexte suivante :

$G = \{$

$T = \{ \text{que, qui, regarde, regardent, mange, mangent, dort, dorment, tombe, tombent, une, un, la, le, des, les, pommes, pomme, femme, femmes, Pierre, Marie} \}$

$N = \{ s, sn, reln, rela, sv, proa, pron, vt, vi, det, n, np \}$

$S = s$

$P = \{$

(1)  $s \rightarrow sn \ sv$

(2)  $sn \rightarrow det \ n \ reln \mid det \ n \ rela \mid np \ reln \mid np \ rela$

(3)  $reln \rightarrow pron \ sv$

(4)  $rela \rightarrow proa \ sn \ vt$

(5)  $sn \rightarrow det \ n$

(6)  $sn \rightarrow np$

(7)  $sv \rightarrow vi \mid vt \ sn$

(8)  $proa \rightarrow que$

(9) pron → qui

(10) vt → regarde | regardent | mange | mangent

(11) vi → dort | dorment | tombe | tombent

(12) det → une | un | la | le | des | les

(13) n → pommes | pomme | femme | femmes

(14) np → Pierre | Marie

}

Légende :

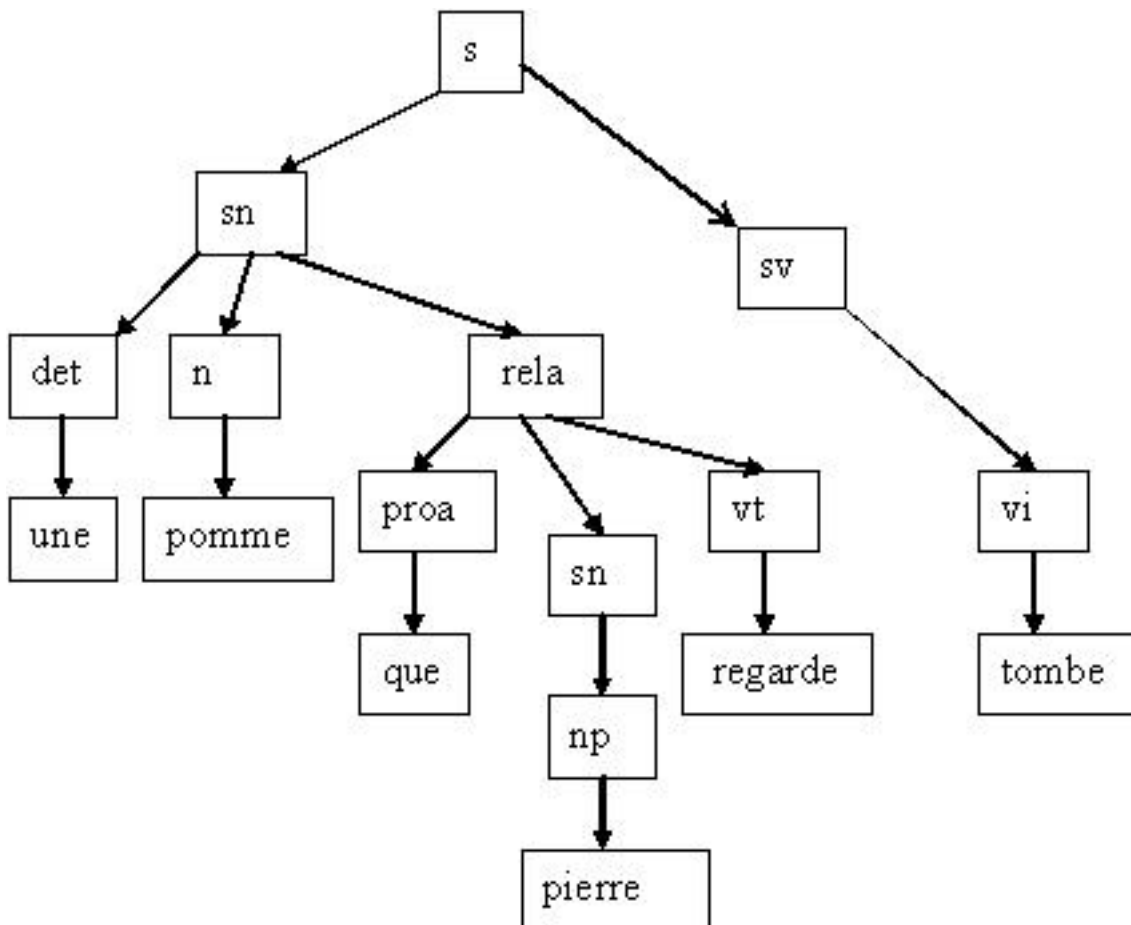
- reln <==> proposition relative nominative
- rela <==> proposition relative accusative
- proa <==> pronom relatif accusatif
- pron <==> pronom relatif nominatif
- vi <==> verbe intransitif
- vt <==> verbe transitif
- det <==> déterminant
- n <==> nom commun
- np <==> nom propre

### Question 1)

#### Énoncé de la question

Vérifier l'appartenance de la phrase **Une pomme que Pierre regarde tombe** au langage reconnu par la grammaire en utilisant un arbre syntaxique.

#### Solution de la question



## Question 2)

### Énoncé de la question

Appliquer l'algorithme CKY pour vérifier :

- l'appartenance de la phrase **Une pomme que Pierre regarde tombe** au langage engendré par la grammaire
- le rejet de la phrase **Une pomme qui Pierre regarde tombe** du langage engendré par la grammaire

### Solution de la question

Pour pouvoir appliquer l'algorithme CKY, il nous faut tout d'abord normaliser la grammaire :

Les règles de production ne respectant pas la FNC sont les suivantes :

(2)  $sn \rightarrow det\ n\ reln \mid det\ n\ rela \mid np\ reln \mid np\ rela$

(4)  $rela \rightarrow proa\ sn\ vt$

(6)  $sn \rightarrow np$

(7)  $sv \rightarrow vi \mid vt\ sn$

On commence par séparer la partie droite des règles contenant plus de 3 non terminaux.

(2a)  $sn \rightarrow DN\ reln \mid DN\ rela \mid np\ reln \mid np\ rela$

(2b)  $DN \rightarrow det\ n$

(4a) rela  $\rightarrow$  proa SNV

(4b) SNV  $\rightarrow$  sn vt

Les règles (6) et (7) contiennent un unique non terminal en partie droite de leur règle, on le remplace par toutes ses décompositions.

(6') sn  $\rightarrow$  Pierre | Marie

(7') sv  $\rightarrow$  vt sn | dort | dorment | tombe | tombent |

On remarque que les règles (11) et (14) deviennent inutiles, on les supprime donc.

On obtient la grammaire suivante :

G = {

T = { que, qui, regarde, regardent, mange, mangent, dort, dorment, tombe, tombent, une, un, la, le, des, les, pommes, pomme, femme, femmes, Pierre, Marie }

N = { s, sn, reln, rela, sv, proa, pron, vt, det, n, DN, SNV }

S = s

P = {

(1) s  $\rightarrow$  sn sv

(2a) sn  $\rightarrow$  DN reln | DN rela | np reln | np rela

(2b) DN  $\rightarrow$  det n

(3) reln  $\rightarrow$  pron sv

(4a) rela  $\rightarrow$  proa SNV

(4b) SNV  $\rightarrow$  sn vt

(5) sn  $\rightarrow$  det n

(6') sn  $\rightarrow$  Pierre | Marie

(7') sv  $\rightarrow$  vt sn | dort | dorment | tombe | tombent |

(8) proa  $\rightarrow$  que

(9) pron  $\rightarrow$  qui

(10) vt  $\rightarrow$  regarde | regardent | mange | mangent

(12) det  $\rightarrow$  une | un | la | le | des | les

(13) n  $\rightarrow$  pommes | pomme | femme | femmes

}

On a maintenant la grammaire sous forme normale de Chomsky, on peut appliquer CKY à nos 2 phrases.

Le tableau  $v$  avec le mot **une pomme que Pierre regarde tombe** est le suivant :

une	det	DN, sn	∅	∅	sn	s
pomme	n	∅	∅	∅	∅	
que	proa	∅	rela	∅		
Pierre	sn	SNV	∅			
regarde	vt	∅				
tombe	sv					

Nous constatons que  $s \in v[1,6]$ , donc **une pomme que Pierre regarde tombe** est un élément du langage.

Le tableau  $v$  avec le mot **une pomme qui Pierre regarde tombe** est le suivant :

une	det	DN, sn	∅	∅	∅	∅
pomme	n	∅	∅	∅	∅	
qui	pron	∅	∅	∅		
Pierre	sn	SNV	∅			
regarde	vt	∅				
tombe	sv					

Nous constatons que  $s \notin v[1,6]$ , donc **une pomme qui Pierre regarde tombe** n'est pas un élément du langage.

### 3 - JFLAP

#### Énoncé :

L'analyse CKY peut être effectuée par le logiciel JFLAP (<http://jflap.org/jflaptmp>).

Pour cela, on se place en mode **grammaire** (*Grammar*). Ensuite, on rentre les différentes règles de la grammaire non normalisée.

On peut alors vérifier qu'il s'agit bien d'une grammaire de type 2 (hors-contexte / *context-free*) grâce à l'item *Test for Grammar Type* de l'onglet *Test*.

On peut ensuite normaliser la grammaire grâce à l'item (*Transform Grammar*) de l'onglet *Convert*.

On peut ensuite réaliser l'analyse CKY grâce à la commande *CYK Parse* de l'onglet *Input*.