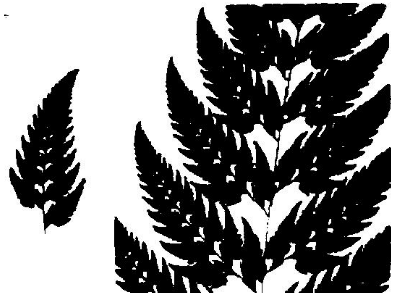


7 – Grammaires hors-contexte (suite)

Récursivité à gauche

- Une grammaire est récursive à gauche dès qu'elle contient une production de la forme :

$$B \to B\,w$$
$$\text{avec } B \in N \text{ et } w \in (N \cup T)^*$$

- la récursivité à gauche provoque des exécutions infinies ... mais elle se retire facilement
- *il peut arriver que du même coup, on supprime l'ambiguïté*

Théorème: Pour toute grammaire hors-contexte récursive à gauche, il existe une grammaire hors-contexte équivalente exempte de récursivité à gauche.

Suppression de la récursivité à gauche

En la supprimant, on la transforme alors en récursivité à droite, non gênante :

- prenons un sous-ensemble de productions :

$$B \to B\,\alpha$$
$$B \to \beta$$

avec $B \in N$, α, $\beta \in (N \cup T)^*$ et β ne commence pas par B

- il engendre le langage :

$$\beta\,\alpha^*$$

- ce langage est aussi engendré par :

$$B \to \beta \mid \beta\,Z$$
$$Z \to \alpha \mid \alpha\,Z$$

avec $Z \notin (N \cup T)^*$

Exemple

- *Reprise* : G = (N,T,P,E) est ambiguë et récursive à gauche
  - N = { E }
  - E axiome
  - T = { +, *, i, (,) }
  - P = { E → E + E
    E → E * E
    E → i
    E → (E) }
- en ôtant la récursivité à gauche de G, on obtient G' = (N',T,P',E) :
  - N' = { E, Z, Z' }
  - T = { +, *, i, (,) }
  - E axiome
  - P = { E → i | i Z
    E → (E) | (E) Z
    Z → + E | + E Z
    E → i Z'
    E → (E) Z'
    Z' → * E | * E Z' }
- G' n'est plus récursive à gauche ... est-elle ambiguë ?

Forme Normale de Greibach

- on peut faire en sorte que chaque production commence par « produire » un terminal, excepté pour l'axiome qui peut éventuellement engendrer ε
- une grammaire hors-contexte G = (N,T,P,S) est sous Forme Normale de Greibach (F.N.G) si toute production est de la forme :

$$A \to a\ \alpha$$

avec

$$A \in N,\ a \in T \text{ et } \alpha \in V^* = (N \cup T)^*$$

- toute grammaire hors-contexte propre et sans production vide peut-être mise sous F.N.G.

Sheila Greibach, née en 1939
Prof. CS, UCLA

Problème de l'appartenance

- il s'agit d'un problème de décision sur les langages hors-contexte (essentiel en compilation, traitement des langues naturelles ...)

  Données :
  - G = (N,T,P,S) une grammaire hors-contexte
  - $\mu \in T^*$ un mot

  Question :
  - est-ce que $\mu \in L(G)$?

- une idée consiste à faire une analyse descendante du mot

  soit G=(N,T,P,S) telle que P ait k productions : on mime depuis l'axiome les dérivations de G. Avec G sous F.N.G, les algorithmes obtenus sont en $k^{|m|}$, $|m|$ étant la longueur du mot à traiter

- une autre idée consiste à faire une analyse ascendante du mot
  Algorithme de Cocke, Younger et Kasami (1965) :
  - il part d'une grammaire sous F.N.C.
  - il utilise l'idée de programmation dynamique vue en Algorithmique.

Idée

- Soit G = (N,T,P,S) une grammaire hors-contexte sous F.N.C. pour

$$L = L(G) = L(G) \setminus \{\varepsilon\}$$

- Facteur de μ débutant à la lettre $\mu(i)$ et de longueur j :

$$\mu_{i,j} = \mu(i)\ \mu(i+1)\ \dots\ \mu(i+j-1)$$

- Pour tout couple (i,j), on calcule l'ensemble des non-terminaux :

$$V_{i,j} = \{\ A,\ A \in N \text{ et tel que } A \Rightarrow^* \mu_{i,j}\ \}$$

- Le problème de l'appartenance se formule ainsi :

$$\mu \in L(G)\ ? \Leftrightarrow S \in V_{1,|\mu|}\ ?$$

Algorithme CYK

début

pour i **de** 1 **à** n **faire** *// initialisation 1re ligne*

$V_{i,1} \leftarrow \{A,\ A \in N,\ A \to \mu(i) \in P\}$ *// avec j fixé à 1*

pour j **de** 2 **à** n **faire**

  pour i **de** 1 **à** n-j+1 **faire** *// pour chaque case de la ligne j*

    $V_{i,j} \leftarrow \varnothing$

    pour k **de** 1 **à** j-1 **faire**

$$V_{i,j} \leftarrow V_{i,j} \cup \{\ A \text{ tels que } A \to BC \in P \text{ avec } B \in V_{i,k} \text{ et } C \in V_{i+k,j-k}\ \}$$

si S $\in V_{1,|\mu|}$ **alors** *μ est engendré* par G **sinon** *non*

fin

Exemple

$G = (\{S,A,B,C\}, T=\{a,b\}, S, P)$

$P = \{\ S \rightarrow AB \mid BB$
$\quad A \rightarrow CC \mid AB \mid a$
$\quad B \rightarrow BB \mid CA \mid b$
$\quad C \rightarrow BA \mid AA \mid b\ \}$

A-t-on μ = aabbab $\in L(G)$?

$j = 1$
pour i de 1 à n faire
$V_{i,1} \leftarrow \{A,\ A \in N$ et $A \rightarrow \mu(i) \in P\ \}$

i

a	a	b	b	a	b
A	A	B,C	B,C	A	B,C

j

Exemple

$S \rightarrow AB \mid BB$
$A \rightarrow CC \mid AB \mid a$
$B \rightarrow BB \mid CA \mid b$
$C \rightarrow BA \mid AA \mid b$

nombre de cases par lignes

pour $j = 2$
pour i de 1 à $n-j+1$ faire
$V_{i,j} \leftarrow \varnothing$
pour k de 1 à $j-1$ faire
$V_{i,j} \leftarrow V_{i,j} \cup \{A,\ A \rightarrow BC \in P,$ avec $B \in V_{i,k}$ et $C \in V_{i+k,j-k}\}$

i

a	a	b	b	a	b
A	A	B,C	B,C	A	B,C

j

Exemple

$S \rightarrow AB \mid BB$
$A \rightarrow CC \mid AB \mid a$
$B \rightarrow BB \mid CA \mid b$
$C \rightarrow BA \mid AA \mid b$

pour $j = 2$
pour $i = 1$
$V_{i,j} \leftarrow \varnothing$
pour k de 1 à $j-1$ faire
$V_{i,j} \leftarrow V_{i,j} \cup \{A,\ A \rightarrow BC \in P,$ avec $B \in V_{i,k}$ et $C \in V_{i+k,j-k}\}$

i

a	a	b	b	a	b
A	A	B,C	B,C	A	B,C

j

Exemple

$S \rightarrow AB \mid BB$
$A \rightarrow CC \mid AB \mid a$
$B \rightarrow BB \mid CA \mid b$
$C \rightarrow BA \mid AA \mid b$

$V_{2,1}$

$V_{1,1}$

$V_{1,2}$

pour $j = 2$
pour $i = 1$
$V_{1,2} \leftarrow \varnothing$
pour $k = 1$
$V_{1,2} \leftarrow V_{1,2} \cup \{A,\ A \rightarrow BC \in P,$ $B \in V_{1,1}$ et $C \in V_{2,1}\ \}$

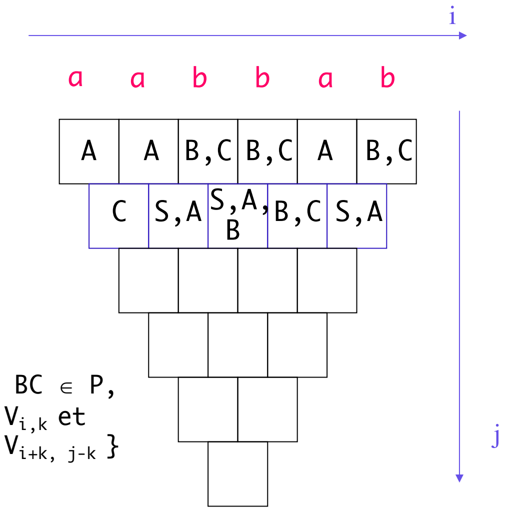
i

a	a	b	b	a	b
A	A	B,C	B,C	A	B,C
C					

j

Exemple

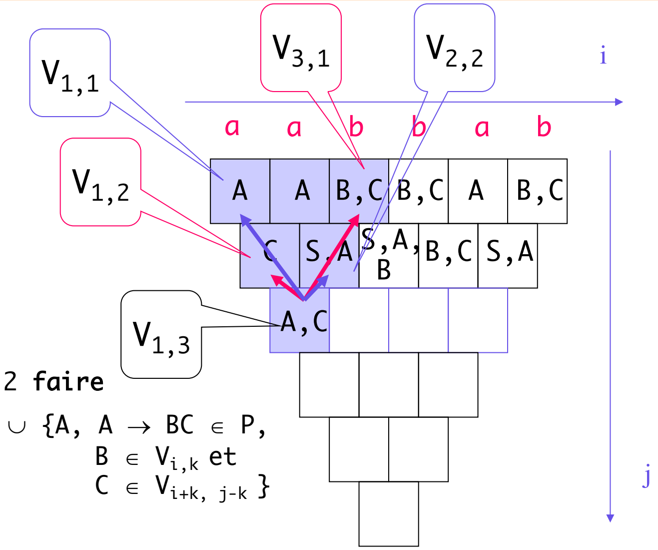
S → AB | BB
A → CC | AB | a
B → BB | CA | b
C → BA | AA | b



pour j = 2
pour i = 2, 3, 4, 5
$V_{i,j} \leftarrow \varnothing$
pour k = 1
$V_{i,j} \leftarrow V_{i,j} \cup \{A,\ A \rightarrow BC \in P,$
$B \in V_{i,k}$ et
$C \in V_{i+k,\ j-k}\}$

Exemple

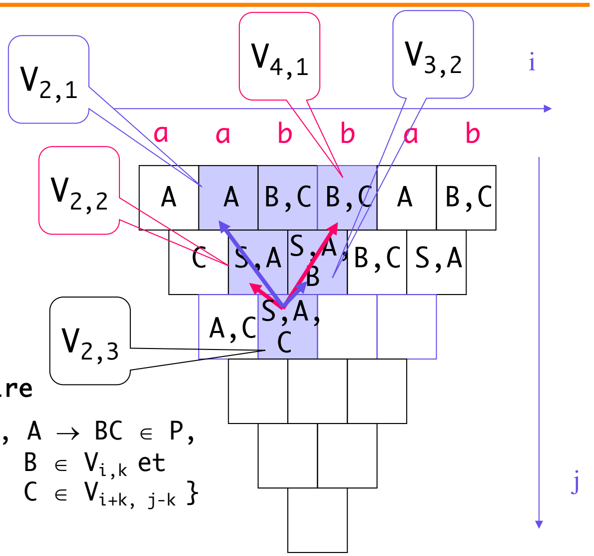
S → AB | BB
A → CC | AB | a
B → BB | CA | b
C → BA | AA | b



pour j = 3
pour i = 1
$V_{i,j} \leftarrow \varnothing$
pour k **de** 1 **à** 2 **faire**
$V_{i,j} \leftarrow V_{i,j} \cup \{A,\ A \rightarrow BC \in P,$
$B \in V_{i,k}$ et
$C \in V_{i+k,\ j-k}\}$

Exemple

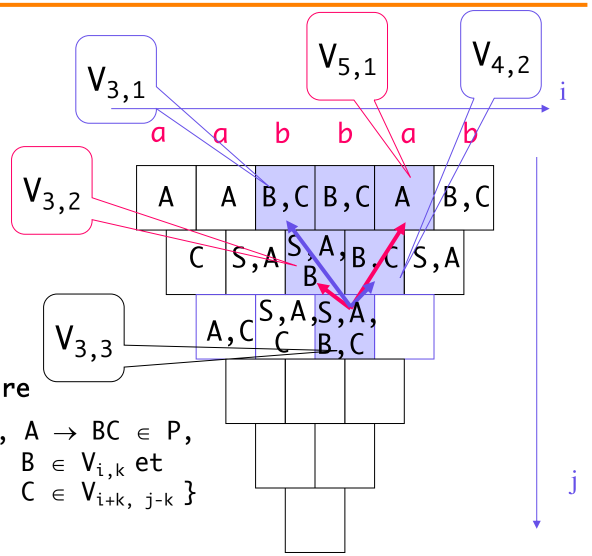
S → AB | BB
A → CC | AB | a
B → BB | CA | b
C → BA | AA | b



pour j = 3
pour i = 2
$V_{i,j} \leftarrow \varnothing$
pour k **de** 1 **à** 2 **faire**
$V_{i,j} \leftarrow V_{i,j} \cup \{A,\ A \rightarrow BC \in P,$
$B \in V_{i,k}$ et
$C \in V_{i+k,\ j-k}\}$

Exemple

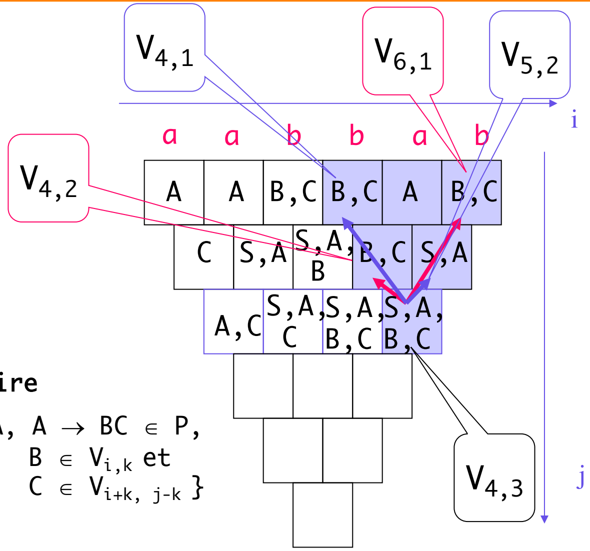
S → AB | BB
A → CC | AB | a
B → BB | CA | b
C → BA | AA | b



pour j = 3
pour i = 3
$V_{i,j} \leftarrow \varnothing$
pour k **de** 1 **à** 2 **faire**
$V_{i,j} \leftarrow V_{i,j} \cup \{A,\ A \rightarrow BC \in P,$
$B \in V_{i,k}$ et
$C \in V_{i+k,\ j-k}\}$

Exemple

S → AB | BB
A → CC | AB | a
B → BB | CA | b
C → BA | AA | b

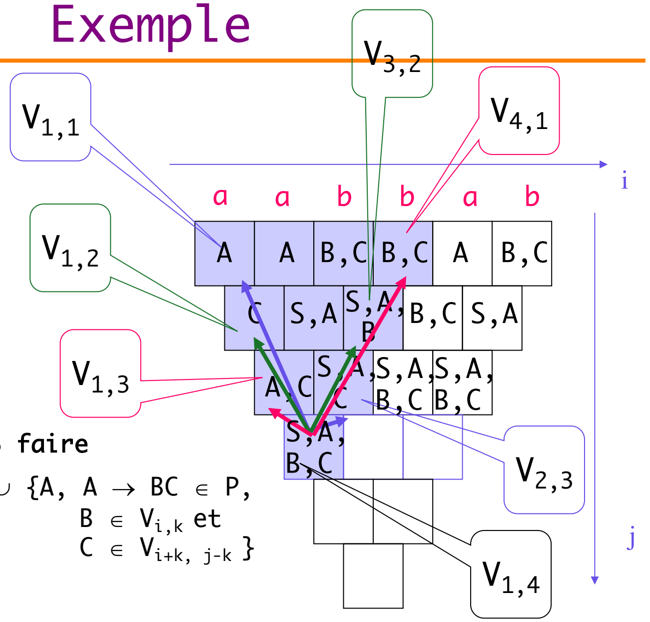


pour j = 3
pour i = 4

$V_{i,j} \leftarrow \varnothing$
pour k de 1 à 2 faire
$V_{i,j} \leftarrow V_{i,j} \cup \{A,\ A \to BC \in P,$
$B \in V_{i,k}$ et
$C \in V_{i+k,\ j-k}\}$

Exemple

S → AB | BB
A → CC | AB | a
B → BB | CA | b
C → BA | AA | b

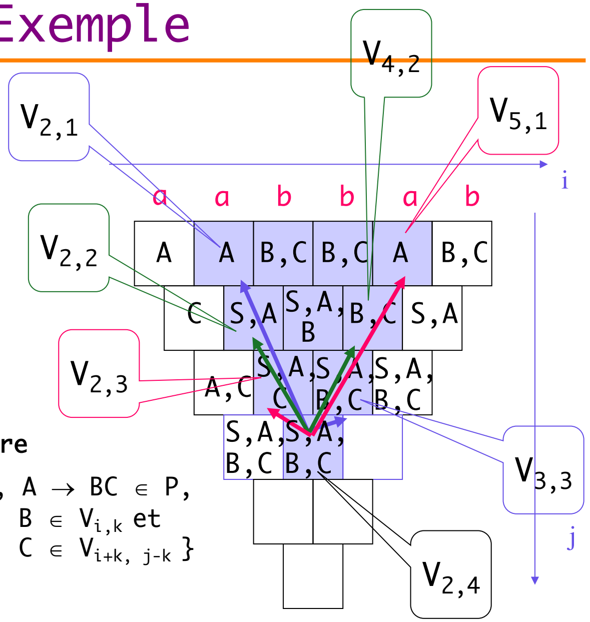


pour j = 4
pour i = 1

$V_{i,j} \leftarrow \varnothing$
pour k de 1 à 3 faire
$V_{i,j} \leftarrow V_{i,j} \cup \{A,\ A \to BC \in P,$
$B \in V_{i,k}$ et
$C \in V_{i+k,\ j-k}\}$

Exemple

S → AB | BB
A → CC | AB | a
B → BB | CA | b
C → BA | AA | b

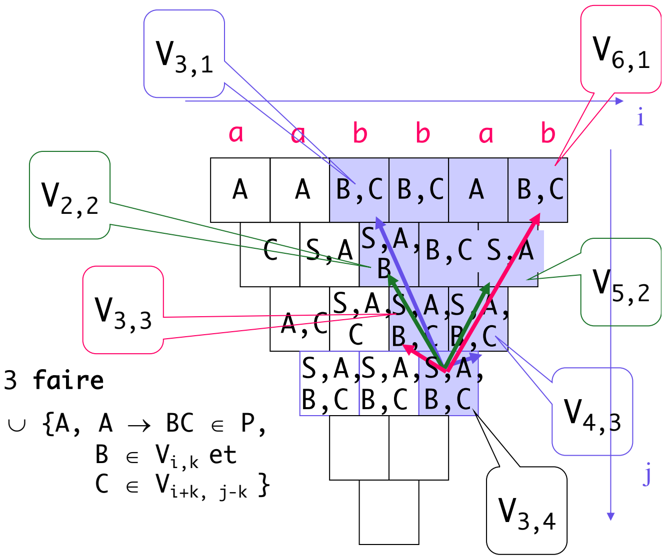


pour j = 4
pour i = 2

$V_{i,j} \leftarrow \varnothing$
pour k de 1 à 3 faire
$V_{i,j} \leftarrow V_{i,j} \cup \{A,\ A \to BC \in P,$
$B \in V_{i,k}$ et
$C \in V_{i+k,\ j-k}\}$

Exemple

S → AB | BB
A → CC | AB | a
B → BB | CA | b
C → BA | AA | b



pour j = 4
pour i = 3

$V_{i,j} \leftarrow \varnothing$
pour k de 1 à 3 faire
$V_{i,j} \leftarrow V_{i,j} \cup \{A,\ A \to BC \in P,$
$B \in V_{i,k}$ et
$C \in V_{i+k,\ j-k}\}$

Exemple

$S \rightarrow AB \mid BB$
$A \rightarrow CC \mid AB \mid a$
$B \rightarrow BB \mid CA \mid b$
$C \rightarrow BA \mid AA \mid b$

pour j = 5
pour i = 1
$V_{i,j} \leftarrow \varnothing$
pour k **de** 1 **à** 4 **faire**
$V_{i,j} \leftarrow V_{i,j} \cup \{A,\ A \rightarrow BC \in P,$
$B \in V_{i,k}$ et
$C \in V_{i+k,\ j-k}\ \}$

a	a	b	b	a	b
A	A	B,C	B,C	A	B,C
C	S,A	S,A,B	B,C	S,A	
A,C	S,A,C	S,A,B,C	S,A,B,C		
S,A,B,C	S,A,B,C	S,A,B,C			
S,A,B,C					

i
j
$V_{1,5}$

Exemple

$S \rightarrow AB \mid BB$
$A \rightarrow CC \mid AB \mid a$
$B \rightarrow BB \mid CA \mid b$
$C \rightarrow BA \mid AA \mid b$

pour j = 5
pour i = 1
$V_{i,j} \leftarrow \varnothing$
pour k **de** 1 **à** 4 **faire**
$V_{i,j} \leftarrow V_{i,j} \cup \{A,\ A \rightarrow BC \in P,$
$B \in V_{i,k}$ et
$C \in V_{i+k,\ j-k}\ \}$

a	a	b	b	a	b
A	A	B,C	B,C	A	B,C
C	S,A	S,A,B	B,C	S,A	
A,C	S,A,C	S,A,B,C	S,A,B,C		
S,A,B,C	S,A,B,C	S,A,B,C			
S,A,B,C	S,A,B,C				

i
j
$V_{2,5}$

Exemple

$S \rightarrow AB \mid BB$
$A \rightarrow CC \mid AB \mid a$
$B \rightarrow BB \mid CA \mid b$
$C \rightarrow BA \mid AA \mid b$

pour j = 6
pour i = 1
$V_{i,j} \leftarrow \varnothing$
pour k **de** 1 **à** 5 **faire**
$V_{i,j} \leftarrow V_{i,j} \cup \{A,\ A \rightarrow BC \in P,$
$B \in V_{i,k}$ et
$C \in V_{i+k,\ j-k}\ \}$

a	a	b	b	a	b
A	A	B,C	B,C	A	B,C
C	S,A	S,A,B	B,C	S,A	
A,C	S,A,C	S,A,B,C	S,A,B,C		
S,A,B,C	S,A,B,C	S,A,B,C			
S,A,B,C	S,A,B,C				
S,A,B,C					

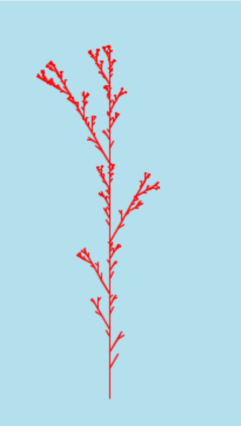
i
j
$V_{1,6}$

↙ $S \in V_{1,6}$ donc $\mu \in L(G)$

Systèmes de Lindenmayer

- Les systèmes de Lindenmayer* sont des systèmes de réécriture inventés en 1968 par le biologiste A. Lindenmayer
- ils modélisent entre autres la croissance des plantes
- ce sont des grammaires de Type I ou II
- parmi eux, les D0L-systems sont hors-contexte

```
def plante(n,T) :     # un D0L-system
    if n == 0 : forward(T)
    else :
        r = randint(0,5)
        if r<3:
            plante(n-1,2*T/3)
            right(30)
            plante(n-1,T/3)
            back(T/3)
            left(30)
            plante(n-1,T/3)
        else :
            plante(n-1,2*T/3)
            left(30)
            plante(n-1,T/3)
            back(T/3)
            right(30)
            plante(n-1,T/3)
```



* L-systems en anglais