ALGORITHME CYK

Floyd Biegel p320.

CYK = Cocke, Younger et Kasami.

Problème

"Savoir si un mot est engendré par une grammaire"

Entrée $\begin{cases} G = (\Sigma, \Gamma, R, S) \text{ sous FNC (ie dont les règles sont de la forme } X \to UV \text{ ou } X \to a) \\ w \text{ un mot de } \Sigma^*\text{, de lettres } w_1 \ldots w_n. \end{cases}$

Sortie 1 OUI si $w \in L_G(S)$, NON sinon

Sortie 2 Si $w \in L_G(S)$, une dérivation de S à w dans G, Sinon NON.

Idée

Mettre en oeuvre le paradygme de la prog. dynamique ayant remarqué la propriété ci dessous.

Pté

- Si $|w| = 1$, disons $w = a \in \Sigma$ alors $X \to^* w$ ssi $(X \to a) \in R$

⇐ clair
⇒ car G sous FNC

- Si $|w| > 1$ alors $X \to^* w$ ssi $\begin{cases} \text{il existe } (U,V) \in \Gamma^2 \text{ et } (\alpha_1, \alpha_2) \in \Sigma^{*2} \text{ tq} \\ (X \to UV) \in R \\ U \to^* \alpha_1 \\ V \to^* \alpha_2 \\ \alpha_1 \alpha_2 = w \end{cases}$

⇐ clair
⇒ G sous FNC + lme fond.

Donc en notant, pour $X \in \Gamma$, $i \in [0..n-1]$, $j \in [1..n]$

$$m_{X,i,j} := \begin{cases} 1 \text{ si } X \to^* w_{i+1} \ldots w_j \\ 0 \text{ sinon} \end{cases}$$

on a la relation de récurrence :

$$\left(m_{X,i,j} = \bigvee_{k=i+1}^{k=j-1} m_{U,i,k} \wedge m_{V,k,j} \right) \text{ si } (X \to UV) \in R \text{ et } j > i+1$$

et $m_{X,i,i+1} = 1$ ssi $(X \to w) \in R$

Algorithme

1. Décision

```
CYK (G, w)
  n ← |w|
  M ← tableau indexé par Γ x [0..n-1] x [1..n] , initialisé à faux
  R ← ______________

  // t = 1
  Pour r allant de 1 à Nr
      X ← Gauche (r)
      α ← Droit (r)
      Si α ∈ Σ
      alors pour i allant de 0 à n-1
               si w_{i+1} = α
               alors M[X, i, i+1] ← vrai
                     R[X, i, i+1] ← (α, -1)

  Pour t allant de 2 à n
      Pour r allant de 1 à Nr
          X ← Gauche (r)
          α ← Droit (r)
          Si α = UV
          alors pour i allant de 0 à n-t
                   pour k allant de i+1 à i+t-1
                       Si M[U, i, k] et M[V, k, i+t]
                       alors M[X, i, i+t] ← vrai
                             R[X, i, i+t] ← (α, k)
  Retourner M[S, 0, n]  ou  ☆
```

NB : le code en bleu n'est pas utile pour fournir la sortie 1.

2. Dérivation

☆ : Si $M[S, 0, n]$ = vrai

alors

AT ← pile vide

AT. ajouter $((S, 0, n))$

γ ← mot vide

δ ← mot vide

imprimer $\gamma S \delta \rightarrow$

Tant que AT n'est pas vide

(X, i, j) = AT. extraire

Match $R[X, i, j]$ with

| $(a, -1)$ → imprimer $\gamma a \delta \rightarrow$
$\gamma \leftarrow \gamma a$
$\delta \leftarrow \delta$ privé de sa 1ère lettre

| (UV, k) → imprimer $\gamma UV \delta \rightarrow$
$\delta \leftarrow V\delta$
AT. ajouter (V, k, j)
AT. ajouter $((U, i, k)$

NB en fait on sait construire la dérivation de X_0 à tout facteur de w $w_{i_0+1} — w_{j_0}$ en remplaçant $S \rightarrow X_0$, $0 \rightarrow i_0$, $n \rightarrow j_0$, pourvu que cette dérivation existe bien, c'est ce que vérifie la condition du Si

ex

G : $S \to AB$ $r=1$
$A \to BC$ $r=2$
$A \to a$ $r=3$
$B \to b$ $r=4$
$B \to BC$ $r=5$
$C \to c$ $r=6$

$w = a\ b\ c\ c$ $|w| = 4$

$n = 4$.

$N_r = 6$.

CYK :

S	1	2	3	4	
0	F	V (AB,1)	V (AB,1)	$\boxed{\text{V (AB,1)}}$	t=4
1		F	F	F	t=3
2			F	F	t=2
3				F	t=1

A	1	2	3	4
0	V (a,-1)	F	F	V (BC,3)
1		F	V (BC,2)	V (BC,3)
2			F	F
3				F

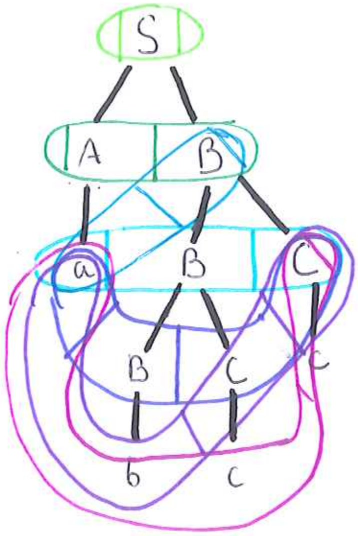
B	1	2	3	4
0	F	F	F	V (BC,3)
1		V (b,-1)	V (BC,2)	V (BC,3)
2			F	F
3				F

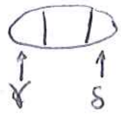
C	1	2	3	4
0	F	F	F	F
1		F	F	F
2			V (c,-1)	F
3				V (c,-1)

AT

☆ :

pile		
S,0,4	S	$\gamma = \varepsilon$, $\delta = \varepsilon$
A,0,1 / B,1,4	AB	$\gamma = \varepsilon$, $\delta = B$
B,1,4	aB	$\gamma = a$, $\delta = \varepsilon$
B,1,3 / C,3,4	aBC	$\gamma = a$, $\delta = C$
B,1,2 / C,2,3 / C,3,4	aBCC	$\gamma = a$, $\delta = CC$
C,2,3 / C,3,4	abCC	$\gamma = ab$, $\delta = C$
C,3,4	abcC	$\gamma = abc$, $\delta = \varepsilon$
(vide)	abcc	





"légende" γ δ. on déduit