Algorithme CYK

Réf : Floyd & Beigel : Le langage des machines
Carton : LFCC

Prop : Soit G une grammaire donnée sous forme normale de Chomsky. Alors, étant donné un mot w, on peut tester si $w \in L(G)$ en temps $O(|w|^3)$

Démo
- Soit $w = w_1 \dots w_n$ le mot à tester. On note $w_{ik} = w_i w_{i+1} \dots w_k$.
On va procéder par programmation dynamique en résolvant :
$T[i,k,X] :=$ est ce que le non-terminal X dérive w_{ik}.

- Pour le cas de base où $i = k$, on a $w_{ik} = w_i = c \in \Sigma$, et donc $T[i,i,X]$ est vrai ssi la production $X \to c \in G$
- Pour le cas où $i < k$, $T[i,j,X]$ est vraie ssi on a une production $X \to YZ \in G$ et un indice l tq $T[i,l,Y]$ et $T[l,j,Z]$ soient vrais.

- On remplit donc le tableau à $d = j - i$ croissant, puis pour chaque $i \in [0; n-d]$ et pour chaque $X \in V_N$.
- La complexité est en $O(n^3)$ et la correction est donné par la relation de récurrence

□

Th : Pour toute grammaire G, il existe une grammaire G' sous forme de Chomsky telle que $L_{G'} = L_G \setminus \{\varepsilon\}$

Démo
- La première étape consiste à se ramener à une grammaire où tous les membres de droite sont soit une lettre terminale, soit un mot formé uniquement de non-terminaux, càd de la forme :

$$\begin{cases} S \to a & \text{pour } a \in V_T \\ S \to S_1 S_2 \dots S_n & \text{, où } S_i \in V_N \end{cases}$$

Soit $V' = \{V_a \mid a \in A\}$ un ensemble de nouvelles variables en bijection avec A. Soit G' la grammaire $(A, V \cup V', P')$ où P' est l'ensemble $\{V_a \to a \mid a \in A\} \cup \{S \to \sigma(w) \mid S \to w \in P\}$ et où σ est la substitution définie par $\sigma(S) = S$ pour $S \in V_N$ et $\sigma(a) = V_a$ pour $a \in A$

- Dans la seconde étape, les règles $S \to S_1 S_2 \dots S_n$ avec $n \geq 2$ sont supprimées et remplacées par

$$\begin{cases} S \to S_1 S_2' \\ S_i' \to S_i S_{i+1}' & \text{pour } 2 \leq i < n-1 \\ S_{n-1}' \to S_{n-1} S_n \end{cases}$$

$\square$