1. u appartient à $L(G)$ si et seulement si `t[n][0][0]`.
2. `t[0][d][i]` est vrai si et seulement si $i = 0$, $d = 0$ et $S \to \varepsilon$.
3. `t[1][d][i]` est vrai si et seulement si $X_i \to u_d$ est une règle de G.
4. $$t[l][d][i] = \bigvee_{X_i \to X_j X_k} \bigvee_{p=1}^{l-1} t[p][d][j] \wedge t[l-p] \underbrace{[d+p]}_{\text{si } d+p<n} [k]$$
5. On obtient un algorithme de programmation dynamique en complexité $\mathrm{O}(n^3|R|)$.

Algorithme CYK

Entrée : Une grammaire $G = (\Sigma, V, R, S)$ et un mot $u = u_0 ... u_{n-1}$
Sortie : `true` si $u \in L(G)$, `false` sinon

$t \longleftarrow$ tableau de taille $(n+1) \times n \times k$ initialisé à `false`, où $k = |V|$
Si $S \to \varepsilon \in R$ **:**
  $t[0][0][0] \longleftarrow$ `true`
Pour $X_i \to a \in R$ **:**
  Pour $d = 0$ à $n-1$ **:**
    Si $a = u_d$ **:**
      $t[1][d][i] \longleftarrow$ `true`
Pour $l = 2$ à n **:**
  Pour $d = 0$ à $n-l$ **:**
    Pour $X_i \to X_j X_k \in R$ **:**
      Pour $p = 1$ à $l-1$ **:**
        Si $d+p < n$ et $t[p][d][j] \wedge t[l-p][d+p][k]$ **:**
          $t[l][d][i] \longleftarrow$ `true`
Renvoyer $t[n][0][0]$