

# Examen TLA

## Grammaire

Grammaire est un quadruplet : (T,V,S,P) :

T : Terminaux (Min)

V : Variables : Non Terminaux (Maj)

S : Start Symbol (axiome)

P : Production Rules

Séquence Alpha générée de Beta après l'application de  $N > 0$  :  $A \Rightarrow + B$

Séquence Alpha générée de Beta après l'application de  $N \geq 0$  :  $A \Rightarrow * B$

Protophrase : Chaîne dérivable à partir de l'axiome S

Une grammaire définit un seul langage

Un même langage peut être engendré par plusieurs grammaires

Dérivation la plus à gauche / la plus à droite

Arbre syntaxique :

- Racine : Axiome
- Feuilles : Terminaux
- Noeuds : Non Terminaux

Grammaire ambiguë  $\Rightarrow$  Existe une chaîne admettant deux arbres de dérivation différents

Grammaire Régulière (type 3) :

- À gauche un seul non terminal
- À droite : (2 symboles)
  - Terminal.NonTerminal
  - Terminal
  - Epsilon

Grammaire Hors Contexte (type 2):

- À gauche un seul non terminal
- À droite : combinaison de terminaux et non terminaux

Grammaire contextuelle (type 1) : La tête est une chaîne

**Type 3 inc Type2 inc Type1(contextuelle) inc Type0(générale)**

Propriétés de GHC :

- Fermeture par union/concatenation/\*/+/image miroir
- Non fermeture par intersection/complémentation

Forme Normale de Chomsky :

- Côté droit des règles :
  - Deux symbols non terminaux
  - Un symbole terminal

Convert to chomsky normal form:

- Remove epsilon productions :  
<https://www.youtube.com/watch?v=mlXYQ8ug2v4&list=PLBlnK6fEyqRgp46KUv4ZY69yXmpwKOlev&index=78> (Si l'axiome a une production epsilon est apprait dans une partie droite on ajoute  $S' \Rightarrow S \mid \epsilon$ )
- Replace every terminal with a variable  $A \rightarrow a \mid B \rightarrow b \dots$
- Make sure every production have exactly 2 variables
- Remove unit productions  $A \rightarrow B$  :  
<https://www.youtube.com/watch?v=B2o75KpzfU4&list=PLBlnK6fEyqRgp46KUv4ZY69yXmpwKOlev&index=77>

Eliminer recursivité à gauche :

- Remove epsilon productions
- Remove cycles  $S \rightarrow A \mid A \rightarrow S$
- On choisit un ordre de non terminaux
  - On les traite 1 à 1 et on regarde les recursivité immédiates

Grammaire Generant  $L(G)^*$  a partir de G:

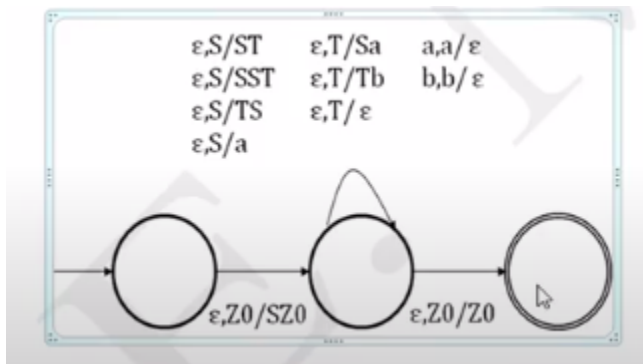
- On ajoute un axiome  $S' \rightarrow SS' \mid \epsilon$  sur G

Grammaire Generant  $L(G1) \cup L(G2)$ :

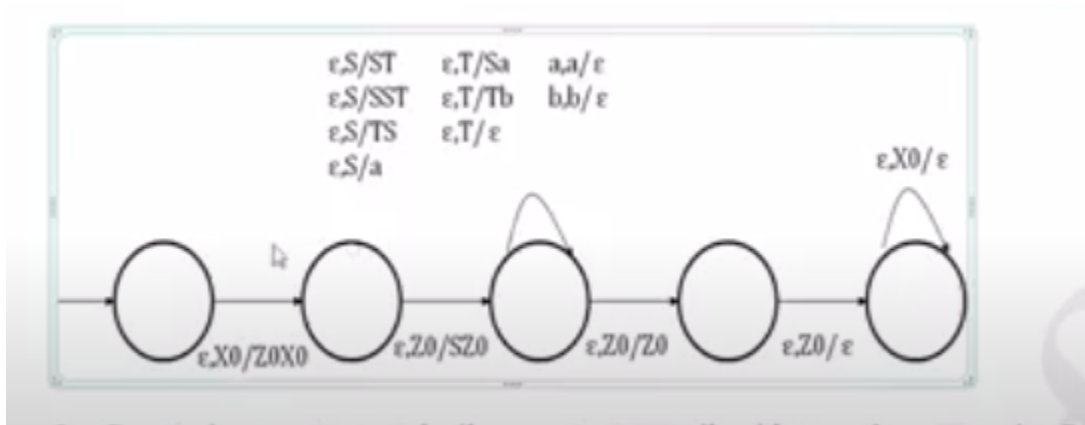
- S'il existe des noms communs, on procède au renommage (dans les productions)
- On ajoute un axiome  $S_{union} \rightarrow S(1) \mid S(2)$

Generer un automate a pile acceptant **par etat final** un grammaire :

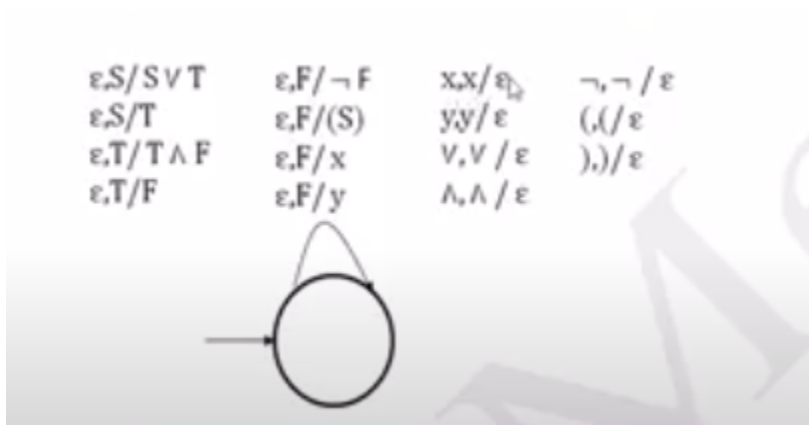
- Etat initial / etat centre / etat final



Generer un automate a pile acceptant **par pile vide** un grammaire à partir de etat final :

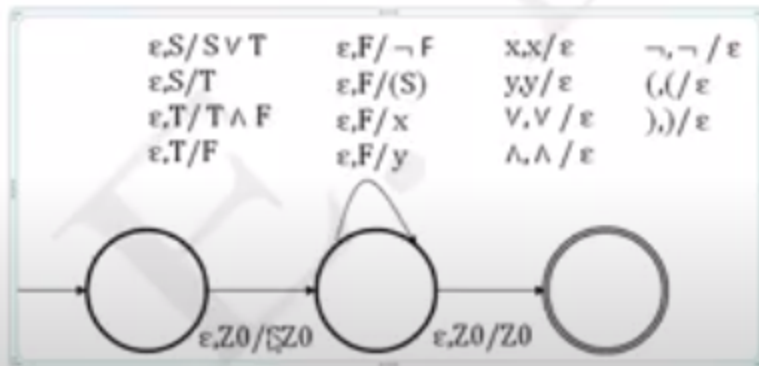


Generer un automate a pile acceptant **par pile vide**:



Le convertir en par etat final :

### Transformation de l'automate pour qu'il accepte par état final



Pour chaque  $L(G)$  tel que  $\epsilon$  n'appartient pas à  $L(G)$  et  $G$  est GHC il existe une grammaire  $C$  sous forme normale de chomsky tel que  $L(G) = L(C)$

Une grammaire est dite propre si elle ne possède pas de règles epsilon ou de règles  $A \rightarrow B$

Mise sous forme normale de chomsky

<https://www.youtube.com/watch?v=FNPSInj3Vt0&list=PLBlnK6fEyqRgp46KUv4ZY69yXmpwKO!ev&index=79>

## Analyse Lexicale

Analyse descendante (Top down) :

- Dérivations les plus à gauches

Analyse SLR et LL(1)

- Elimination de la récursivité et de la factorisation

Montrer que grammaire est LL(1) :

- Non recursive + non ambiguë + factorisé

2. S'il existe une production  $A \rightarrow \alpha \mid \beta$  avec  $(\alpha \neq \beta)$  alors

- i. Il n'existe pas de terminal  $a$  tel que  $\alpha$  et  $\beta$  génèrent tous deux des chaînes qui commencent par  $a$  ( $\text{PREMIER}(\alpha) \cap \text{PREMIER}(\beta) = \emptyset$ )
- ii. Au plus un de  $\alpha$  et  $\beta$  peut générer la chaîne  $\epsilon$  ( $\text{PREMIER}(\alpha) \cap \text{PREMIER}(\beta) = \emptyset$ )
- iii. Si  $\beta \Rightarrow^* \epsilon$  alors  $\alpha$  ne dérive aucune chaîne commençant par un terminal qui est dans  $\text{SUIVANT}(A)$ . ( $\epsilon \in \text{PREMIER}(\beta)$  ET  $\text{PREMIER}(\alpha) \cap \text{SUIVANT}(A) = \emptyset$ )