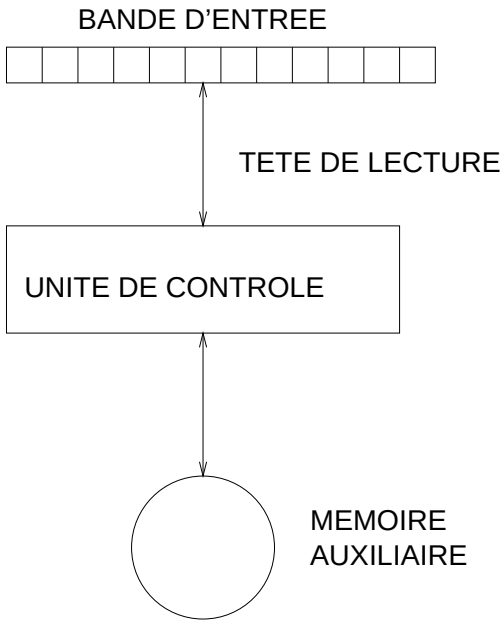
Théorie des langages

Alexis Nasr

Représentation graphique d'un reconnaisseur



Eléments d'un reconnaisseur

Un reconnaisseur est composé de quatre parties :

- 1 - une bande de lecture
  - elle est composée d'une succession de cases.
  - Chaque case pouvant contenir un seul symbole d'un alphabet d'entrée.
  - C'est dans les cases de cette bande de lecture qu'est écrit le mot à reconnaître.
- 2 - une tête de lecture
  - Elle peut lire une case à un instant donné.
  - La case sur laquelle se trouve la tête de lecture à un moment donné s'appelle la case courante.
  - La tête peut être déplacée par le reconnaisseur pour se positionner sur la case immédiatement à gauche ou à droite de la case courante.
- 3 - une mémoire
  - Elle peut prendre des formes différentes.
  - La mémoire permet de stocker des éléments d'un alphabet de mémoire.

Eléments d'un reconnaisseur

- 4 - une unité de contrôle
  - Elle constitue le cœur d'un reconnaisseur.
  - Elle peut être vue comme un progamme qui dicte au reconnaisseur son comportement.
  - Elle est définie par un ensemble fini d'états ainsi que par une fonction de transition qui décrit le passage d'un état à un autre en fonction du contenu de la case courante de la bande de lecture et du contenu de la mémoire.
  - L'unité de contrôle décide aussi de la direction dans laquelle déplacer la tête de lecture et choisit quels symboles stocker dans la mémoire.
  - Parmi les états d'un reconnaisseur, on distingue
    - des états initiaux, qui sont les états dans lesquels doit se trouver le reconnaisseur avant de commencer à reconnaître un mot
    - des états d'acceptation qui sont les états dans lequel doit se trouver le reconnaisseur après avoir reconnu un mot.

Configuration et mouvement

- Configuration d'un reconnaisseur :
  - Etat de l'unité de contrôle
  - Contenu de la bande d'entrée et position de la tête
  - Contenu de la mémoire
- Mouvement : passage d'une configuration à une autre ($C_1 \vdash C_2$)

Configurations

- configuration initiale
  - L'unité de contrôle est dans un état initial
  - La tête est au début de la bande
  - La mémoire contient un élément initial.
- configuration d'acceptation
  - L'unité de contrôle est dans un état d'acceptation
  - La tête de lecture est à la fin de la bande
  - La mémoire se trouve dans un état d'acceptation.

Reconnaissance

- Un mot m est acceptée par un reconnaisseur si, partant de l'état initial, avec m sur la bande d'entrée, le reconnaisseur peut faire une série de mouvements pour se retrouver dans un état d'acceptation.
- Le langage accepté par un reconnaisseur est l'ensemble de tous les mots qu'il accepte.

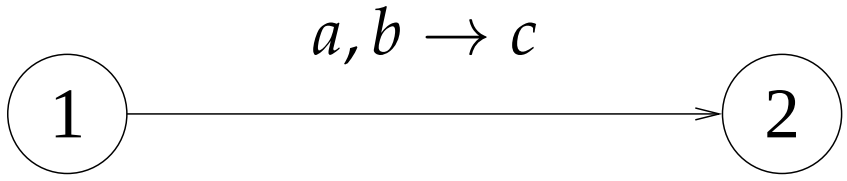
Déterminisme

- L'unité de contrôle est dite déterministe si à toute configuration correspond au plus un mouvement. S'il peut exister plus d'un mouvement, elle est dite non déterministe.
- Le déterminisme est une propriété importante :
  - Un reconnaisseur déterministe reconnaît un mot de longueur n en $O(n)$

Automates à pile

- Forme simple de mémoire : une pile.
- Mode de stockage *Last In First Out*.
- On ne peut accéder qu'à l'élément se trouvant au sommet de la pile.
- Deux opérations possibles :
  - empiler : ajouter un élément au sommet.
  - dépiler : enlever l'élément se trouvant au sommet.
- La pile permet de stocker de l'information sans forcément multiplier le nombre d'états.

Représentation graphique



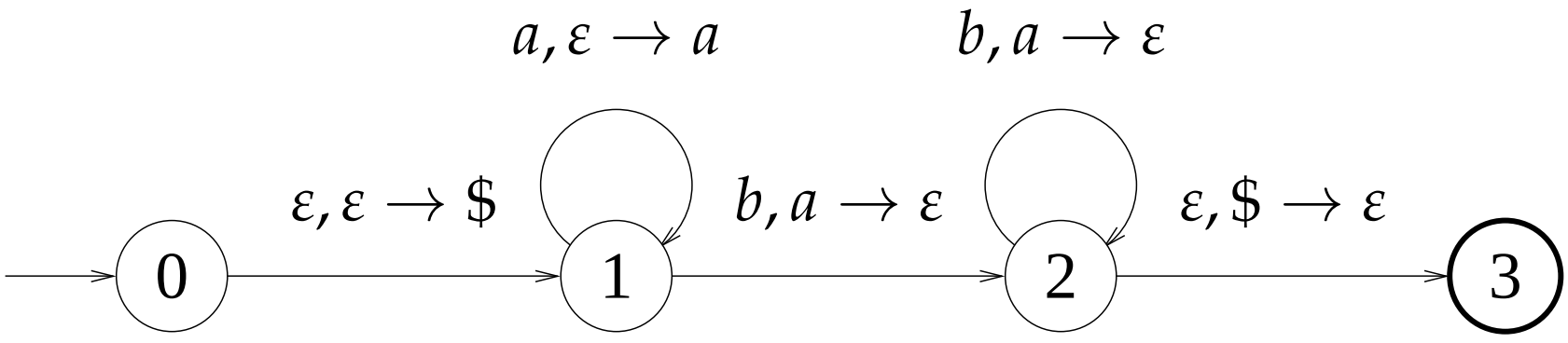
Si l'automate est en 1, et que la tête de lecture est sur a, l'automate :

- décale la tête de lecture d'une case vers la droite
- dépile b (b doit être présent au sommet de la pile)
- empile c
- va en 2

cas particuliers

- si $a = \varepsilon$, l'automate peut franchir cet arc sans lire de symbole.
- si $b = \varepsilon$, l'automate peut franchir cet arc indépendamment du symbole se trouvant en sommet de pile.
- si $c = \varepsilon$, l'automate peut franchir cet arc sans rien empiler.

Exemple


$a, \varepsilon \to a$
$b, a \to \varepsilon$
$\varepsilon, \varepsilon \to \$$
$b, a \to \varepsilon$
$\varepsilon, \$ \to \varepsilon$
0
1
2
3

Définition

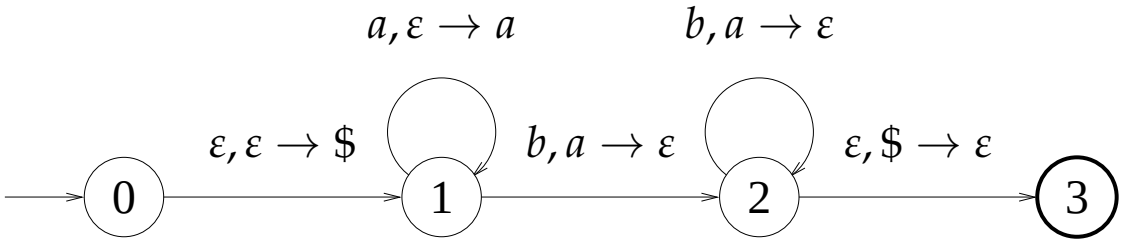
Un automate à pile est un 6-uplet $\langle Q, \Sigma, \Gamma, \delta, q_0, F \rangle$

- Q est l'ensemble des états
- Σ est l'alphabet d'entrée
- Γ est l'alphabet de symboles de pile
- δ est la fonction de transition :

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\}) \to \wp(Q \times \Gamma^*)$$

- $q_0 \in Q$ est l'état initial
- $F \subseteq Q$ est l'ensemble des états d'acceptation

Equivalence



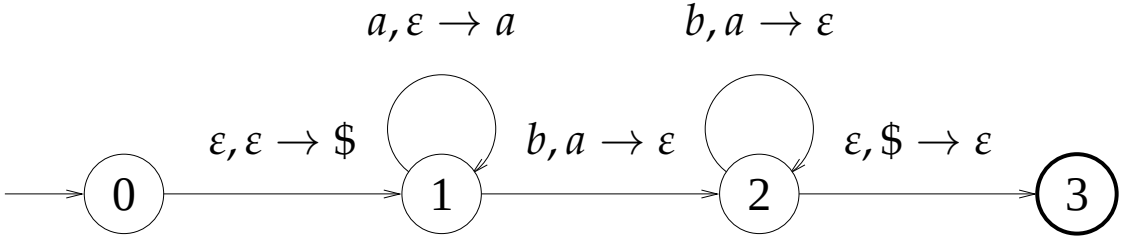
$$
\begin{aligned}
&\langle Q, \Sigma, \Gamma, \delta, q_0, F \rangle \\
Q &= \{0, 1, 2, 3\} \\
\Sigma &= \{a, b\} \\
\Gamma &= \{a, b, \$\} \\
\delta(0, \varepsilon, \varepsilon) &= \{(1, \$)\} \\
\delta(1, a, \varepsilon) &= \{(1, a)\} \\
\delta(1, b, a) &= \{(2, \varepsilon)\} \\
\delta(2, b, a) &= \{(2, \varepsilon)\} \\
\delta(2, \varepsilon, \$) &= \{(3, \varepsilon)\} \\
q_0 &= 0 \\
F &= \{3\}
\end{aligned}
$$

Configurations et mouvement

$A = \langle Q, \Sigma, \Gamma, \delta, q_0, F \rangle$

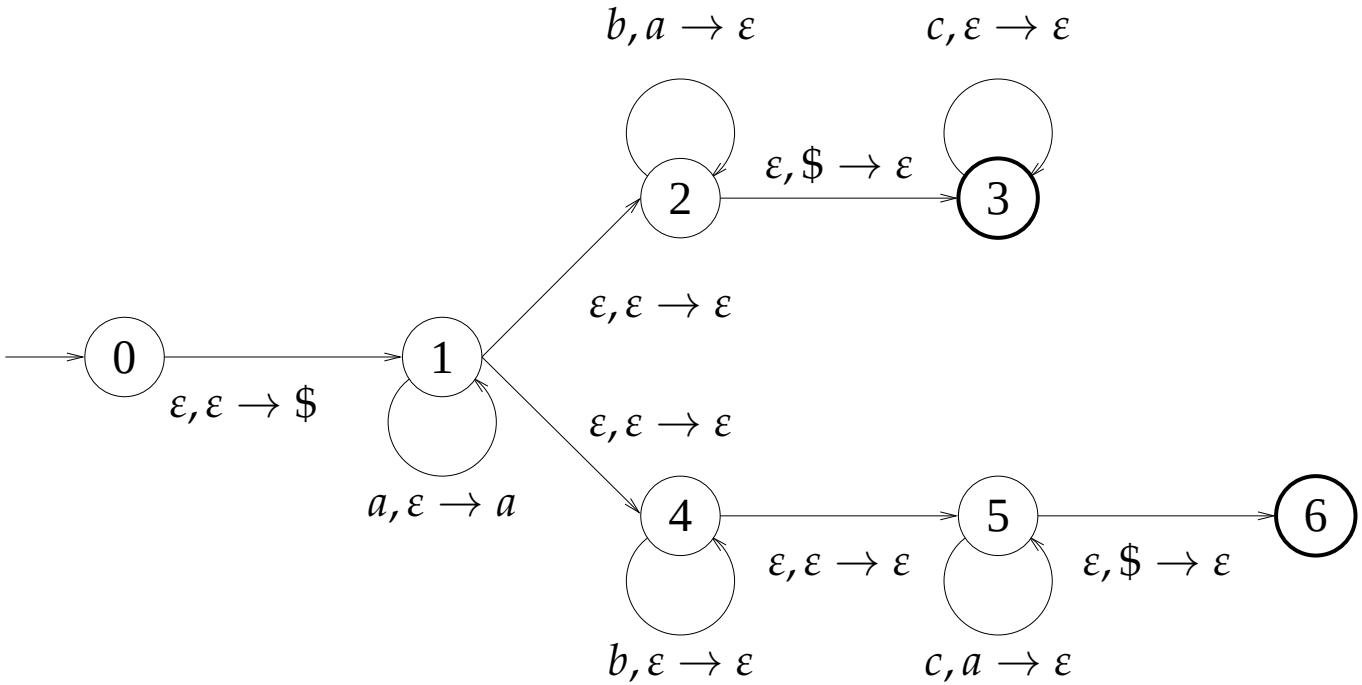
- Configuration : $(q, m, \alpha) \in Q \times \Sigma^* \times \Gamma^*$ où :
  - q représente l'état courant de l'unité de contrôle
  - m est la partie du mot à reconnaître non encore lue. Le premier symbole de m (le plus à gauche) est celui qui se trouve sous la tête de lecture. Si $m = \varepsilon$ alors tout le mot a été lu.
  - α représente le contenu de la pile. Le symbole le plus à gauche est le sommet de la pile. Si $\alpha = \varepsilon$ alors la pile est vide.
- Configuration initiale : (q_0, m, ε) où m est le mot à reconnaître
- Configuration d'acceptation : $(q, \varepsilon, \varepsilon)$ avec $q \in F$
- Mouvement : $(q, aw, Z\alpha) \vdash (q', w, \gamma\alpha)$ si $(q', \gamma) \in \delta(q, a, Z)$.

Reconnaissance

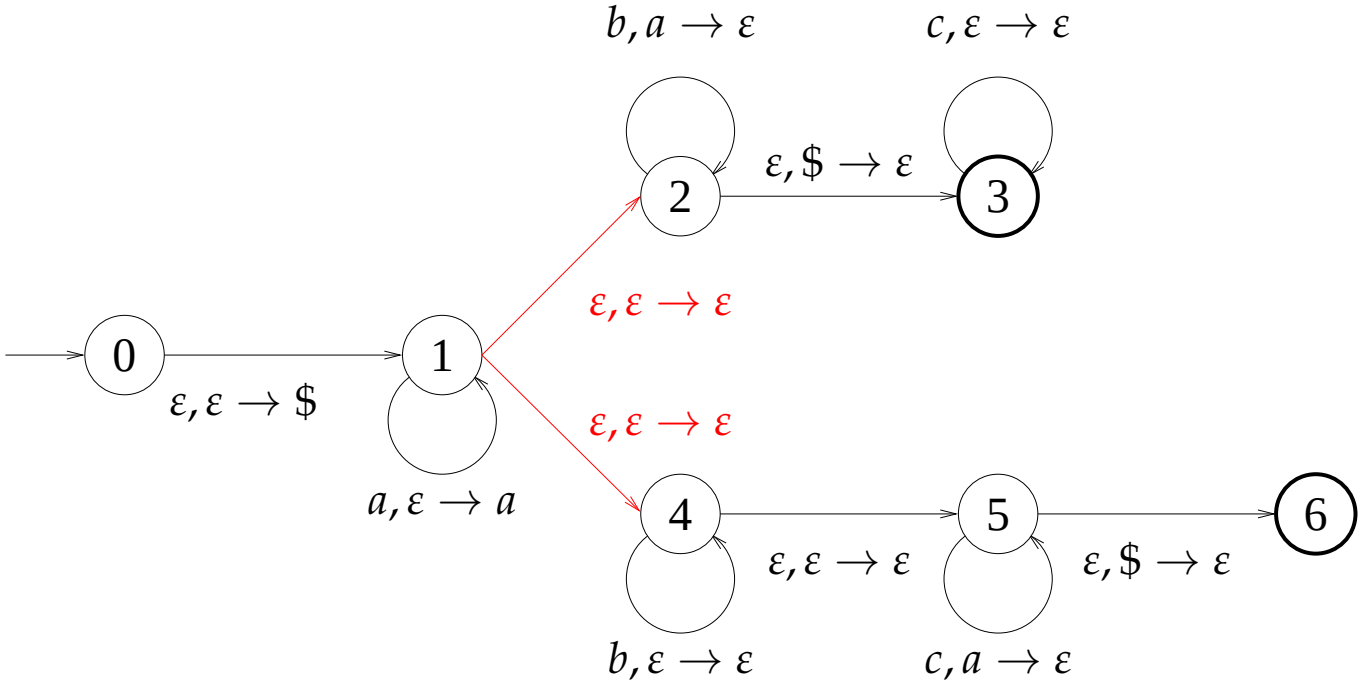


$$
\begin{aligned}
& (0, aaabbb, \varepsilon) \\
\vdash\ & (1, aaabbb, \$) \\
\vdash\ & (1, aabbb, a\$) \\
\vdash\ & (1, abbb, aa\$) \\
\vdash\ & (1, bbb, aaa\$) \\
\vdash\ & (2, bb, aa\$) \\
\vdash\ & (2, b, a\$) \\
\vdash\ & (2, \varepsilon, \$) \\
\vdash\ & (3, \varepsilon, \varepsilon)
\end{aligned}
$$

$$L = \{a^i b^j c^k \mid i = j \text{ ou } i = k\}$$

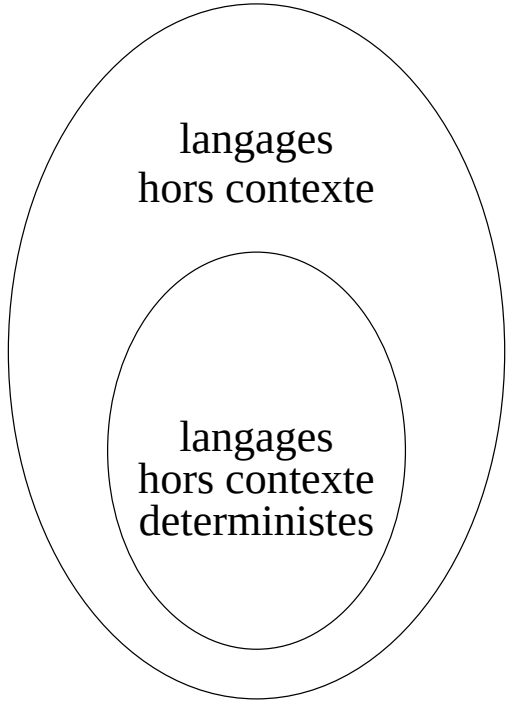


Non déterminisme



$L = \{a^i b^j c^k \mid i = j \text{ ou } i = k\}$ ne peut être reconnu par un automate à pile déterministe !

Langages hors-contexte déterministes



Grammaires hors-contexte $\Leftrightarrow$ Automate à pile

Un langage est hors-contexte si et seulement si il existe un automate à pile qui le reconnaît.

- Si un langage est hors-contexte alors il existe un automate à pile qui le reconnaît.
- Si un langage est reconnu par un automate à pile alors il est hors-contexte.

Grammaires hors-contexte $\Rightarrow$ Automate à pile

- Soit $G = \langle N, \Sigma, P, S \rangle$ une grammaire hors-contexte, on construit un automate à pile A qui accepte un mot m s'il existe une dérivation pour m dans G ($S \overset{+}{\Rightarrow} m$).
- A est conçu de telle sorte à déterminer une dérivation conduisant de S à m.
- Idée clef : écrire dans la pile de A les proto-phrases qui constituent la dérivation recherchée.

Principe

1. Empiler l'axiome S
2. Remplacer S par la partie droite d'une règle de la forme S$\rightarrow \alpha$ de telle sorte que le premier symbole x de α se trouve en sommet de pile.
   - Si x est un terminal alors on le compare avec le caractère se trouvant sous la tête de lecture. S'ils sont égaux alors on dépile.
   - Si x est un non terminal alors on le remplace par la partie droite d'une règle de P de la forme $x \rightarrow \beta$.

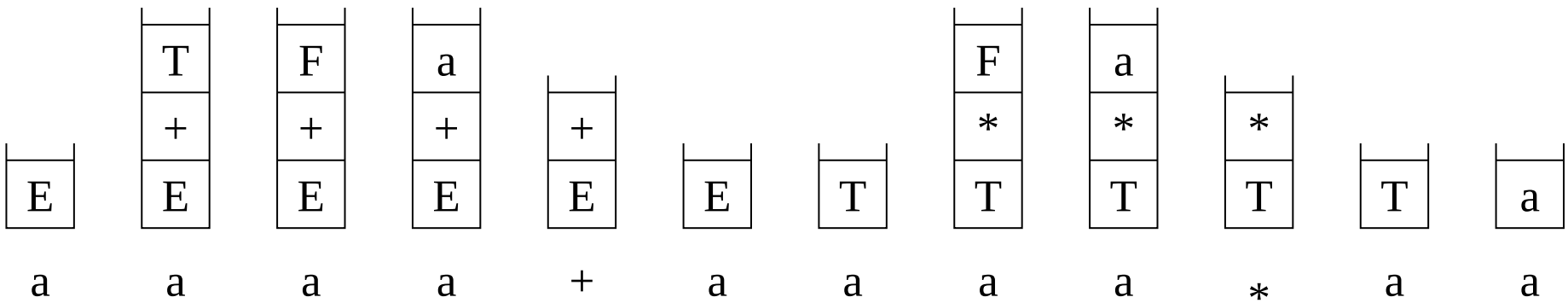
Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

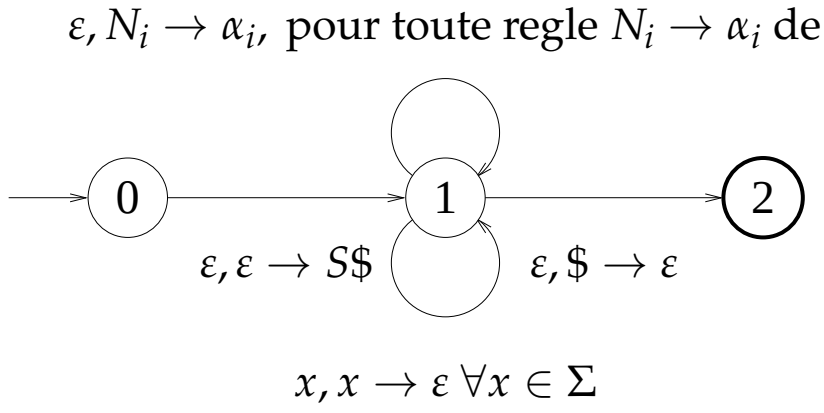
$$E \to T + E \mid T$$
$$T \to F * T \mid F$$
$$F \to (E) \mid a$$



Non déterminisme

- Lorsqu'un non terminal X doit être remplacé au sommet de la pile, il peut l'être par la partie droite d'une règle de la forme $X \to \beta$.
- Plusieurs règles de cette forme peuvent exister dans la grammaire.
- L'automate correspondant est généralement non déterministe.

Automate correspondant à la grammaire $G = \langle N, \Sigma, P, S \rangle$



Construction de l'automate

Automate à pile A correspondant à la grammaire $G = \langle N, \Sigma, P, S \rangle$:

$$A = \langle \{0, 1, 2\}, \Sigma, N \cup \Sigma \cup \{\$\}, \delta, 0, \{2\} \rangle$$

La fonction de transition δ est définie de la façon suivante :

- $\delta(0, \varepsilon, \$) = \{(1, S\$)\}$ On empile l'axiome.
- $\delta(1, \varepsilon, N_i) = \{(1, \alpha_i) \mid \text{avec } N_i \to \alpha_i \in P\}$
  Si un symbole non terminal N_i occupe le sommet de la pile, on le remplace par la partie droite α_i d'une règle $N_i \to \alpha_i$.
- $\delta(1, a, a) = \{(1, \varepsilon) \mid \text{avec } a \in \Sigma\}$
  Si le même symbole terminal occupe le sommet de la pile et la case courante de la bande d'entrée, on dépile.
- $\delta(1, \varepsilon, \$) = \{(2, \$)\}$
  Si le mot en entrée a été reconnu et que la pile ne contient que le symbole de fond de pile, on passe à l'état d'acceptation.

Exemple

Grammaire :

$$\langle \{E, T, F\}, \{a, +, -, *(,)\}, P, E \rangle$$

avec :

$$P = \{E \to T + E \mid T \ , \ T \to F * T \mid F \ , \ F \to (E) \mid a\}$$

Automate :

$$A_1 = \langle \{0, 1, 2\}, \{a, +, *, (,)\}, \{a, +, *, (,), E, T, F, \$\}, \delta, 0, \$, \{2\} \rangle$$

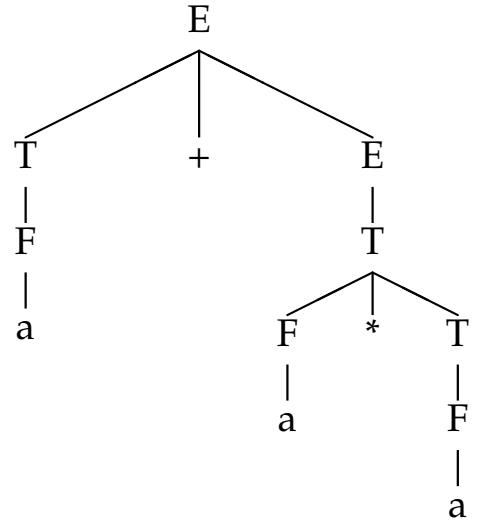
avec :

$$\begin{array}{lcl@{\quad}lcl}
\delta(0, \varepsilon, \$) & = & \{(1, E\$, \varepsilon)\} & \delta(1, +, +) & = & \{(1, \varepsilon)\} \\
\delta(1, \varepsilon, E) & = & \{(1, E + T), \ (1, T)\} & \delta(1, *, *) & = & \{(1, \varepsilon)\} \\
\delta(1, \varepsilon, T) & = & \{(1, T * F), \ (1, F)\} & \delta(1, (, () & = & \{(1, \varepsilon)\} \\
\delta(1, \varepsilon, F) & = & \{(1, (E)), \ (1, a)\} & \delta(1, a, a) & = & \{(1, \varepsilon)\} \\
\delta(1, \varepsilon, \$) & = & \{(2, \$)\} & & &
\end{array}$$

Analyse syntaxique

Etant donné $m \in \Sigma^*$ et $G = \langle \Sigma, N, P, A \rangle$, analyser m consiste à trouver pour m son (et éventuellement ses) arbre de dérivation.

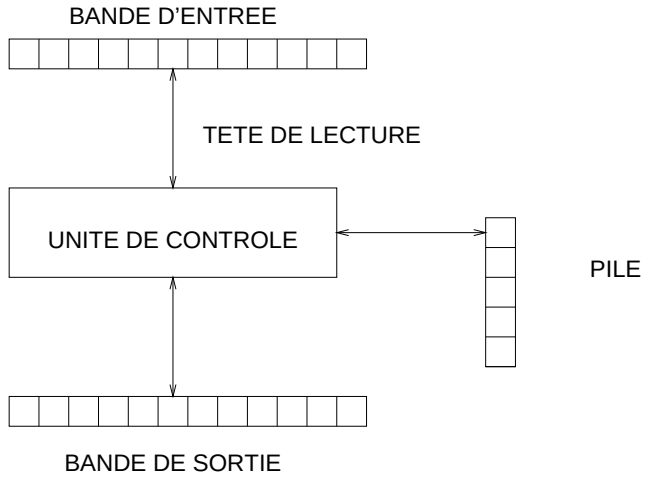
$$
\begin{array}{lcl}
E & \rightarrow & T + E | T \\
T & \rightarrow & F * T | F \\
F & \rightarrow & (E) | a
\end{array}
$$



Sens d'analyse

- Analyse descendante
  L'arbre de dérivation est construit depuis la racine vers les feuilles
  Séquence de dérivations gauches à partir de l'axiome
  $E \Rightarrow T + E \Rightarrow F + E \Rightarrow a + E \Rightarrow a + T \Rightarrow a + F * T \Rightarrow$
  $a + a * T \Rightarrow a + a * F \Rightarrow a + a * a$
- Analyse ascendante
  L'arbre de dérivation est construit des feuilles vers la racine
  Séquence de dérivation telle que la séquence inverse soit une dérivation droite de m.
  $a + a * a \Leftarrow F + a * a \Leftarrow T + a * a \Leftarrow T + F * a \Leftarrow T + F * F \Leftarrow$
  $T + F * T \Leftarrow T + E \Leftarrow E$

Transducteurs à pile



- Un transducteur à pile est un automate à pile qui émet, à chaque déplacement, un suite finie de symboles de sortie.
- Une configuration d'un transducteur à pile est un quadruplet (q, w, α, y) où y est une séquence de symboles de sortie.

Transducteur à pile : définition

Un transducteur à pile est un 8-uplet
$\langle Q, \Sigma, \Gamma, \Delta, \delta, q_0, F \rangle$

- Q est l'ensemble des états
- Σ est l'alphabet d'entrée
- Γ est l'alphabet de symboles de pile
- Δ est l'alphabet de sortie
- δ est la fonction de transition

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to \wp(Q \times \Gamma^* \times \Delta^*)$$

- $q_0 \in Q$ est l'état initial
- $F \subseteq Q$ est l'ensemble des états d'acceptation

Analyseur gauche

$$\begin{array}{ll} 1\ E \to T + E & 2\ E \to T \\ 3\ T \to F * T & 4\ T \to F \\ 5\ F \to (E) & 6\ F \to a \end{array}$$

- Dérivation gauche de $a + a * a$:
  $E \overset{1}{\Rightarrow} T + E \overset{4}{\Rightarrow} F + E \overset{6}{\Rightarrow} a + E \overset{2}{\Rightarrow} a + T \overset{*}{\Rightarrow} a + a * a$
- Analyse gauche : 14623646

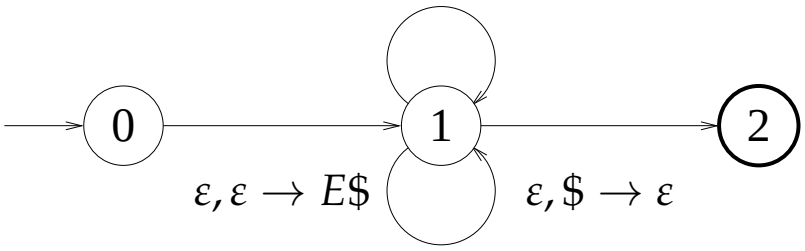
Analyseur gauche

Soit une CFG G dont les règles ont été numérotées de 1 à p. On appelle un analyseur gauche de G, un transducteur à pile non déterministe T_G^g qui produit pour un mot $m \in L(G)$, une dérivation gauche de m.
Performances :

- Espace : $\mathcal{O}(|m|)$
- Temps : $\mathcal{O}(c^{|m|})$

Analyseur gauche : Exemple

$\varepsilon, E \to T, 2$ $\quad$ $\varepsilon, T \to F, 4$
$\varepsilon, E \to T + E, 1$ $\quad$ $\varepsilon, F \to (E), 5$
$\varepsilon, T \to F * T, 3$ $\quad$ $\varepsilon, F \to a$



$x, x \to \varepsilon, \varepsilon$ avec $x \in \{a, +, *, (,)\}$

$$
\begin{aligned}
& (0, a + a * a, \$) \\
\vdash \quad & (1, a + a * a, E\$) \\
\vdash \quad & (1, a + a * a, T + E\$, 1) \\
\vdash \quad & (1, a + a * a, F + E\$, 14) \\
\vdash \quad & (1, a + a * a, a + E\$, 146) \\
\vdash \quad & (1, +a * a, +E\$, 146) \\
\vdash \quad & (1, a * a, E\$, 146) \\
\vdash \quad & (1, a * a, T\$, 1462) \\
\vdash \quad & (1, a * a, F * T\$, 14623) \\
\vdash \quad & (1, a * a, a * T\$, 146236) \\
\vdash \quad & (1, *a, *T\$, 146236) \\
\vdash \quad & (1, a, T\$, 1462364) \\
\vdash \quad & (1, a, F\$, 14623646) \\
\vdash \quad & (1, a, a\$, 14623646) \\
\vdash \quad & (1, \varepsilon, \$, 14623646) \\
\vdash \quad & (2, \varepsilon, \varepsilon, 14623646)
\end{aligned}
$$

Limites des automates à pile

- Certains langages ne peuvent être reconnus par les automates à pile (ne peuvent être générés par une grammaire hors-contexte).
- Exemple : le langage $m\sharp m$ avec $m \in \{0,1\}^*$:
  1. L'automate lit le premier m et le stocke dans la pile.
  2. Il lit le premier symbole du second m.
  3. Comment vérifier qu'il est identique au symbole se trouvant au fond de la pile ?