Analyse syntaxique LR

Alexis Nasr
Franck Dary
Pacôme Perrotin

Compilation – L3 Informatique
Département Informatique et Interactions
Aix Marseille Université

Grammaires hors-contexte

Une grammaire hors-contexte est un 4-uplet $\langle N, \Sigma, P, S \rangle$ où :

- N est un ensemble de symboles non terminaux, appelé l'alphabet non terminal.
- Σ est un ensemble de symboles terminaux, appelé l'alphabet terminal, tel que N et Σ soient disjoints.
- P est un sous ensemble fini de :

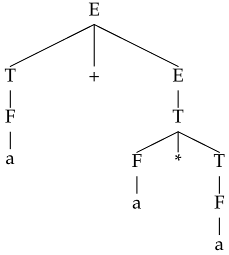
  $$N \times (N \cup \Sigma)^*$$

  un élément (α, β) de P, que l'on note $\alpha \to \beta$ est appelé une règle de production ou règle de réécriture.
  α est appelé partie gauche de la règle
  β est appelé partie droite de la règle
- S est un élément de N appelé l'axiome de la grammaire.

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.

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



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.
  $a + a * a \Leftarrow F + a * a \Leftarrow T + a * a \Leftarrow T + F * a \Leftarrow T + F * F \Leftarrow T + F * T \Leftarrow T + E \Leftarrow E$

Utilisation d’une pile

- Pour l’analyse descendante, comme pour l’analyse ascendante on utilise une pile
- Cette dernière permet de stocker les résultats intermédiaires du processus d’analyse

Analyse Descendante

1. Empiler l'axiome S
2. Remplacer S par la partie droite d'une règle de la forme $S \to \alpha$ 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 $x \to \beta$.

Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\to T + E \mid T \\ T &\to F * T \mid F \\ F &\to (E) \mid a \end{aligned}$$

$$\boxed{E}$$

Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\to T + E \mid T \\ T &\to F * T \mid F \\ F &\to (E) \mid a \end{aligned}$$

$$\begin{array}{|c|} \hline E \\ \hline \end{array} \quad \begin{array}{|c|} \hline T \\ \hline + \\ \hline E \\ \hline \end{array}$$

Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\rightarrow T + E \mid T \\ T &\rightarrow F * T \mid F \\ F &\rightarrow (E) \mid a \end{aligned}$$

$$\begin{array}{ccc} & \boxed{T} & \boxed{F} \\ & \boxed{+} & \boxed{+} \\ \boxed{E} & \boxed{E} & \boxed{E} \end{array}$$

Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\to T + E \mid T \\ T &\to F * T \mid F \\ F &\to (E) \mid a \end{aligned}$$

$$\begin{array}{cccc} & \boxed{T} & \boxed{F} & \boxed{a} \\ & \boxed{+} & \boxed{+} & \boxed{+} \\ \boxed{E} & \boxed{E} & \boxed{E} & \boxed{E} \end{array}$$

Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\rightarrow T + E \mid T \\ T &\rightarrow F * T \mid F \\ F &\rightarrow (E) \mid a \end{aligned}$$

	T	F	a	
	$+$	$+$	$+$	$+$
E	E	E	E	E

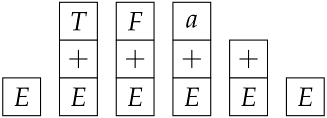
Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\rightarrow T + E \mid T \\ T &\rightarrow F * T \mid F \\ F &\rightarrow (E) \mid a \end{aligned}$$



Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\to T + E \mid T \\ T &\to F * T \mid F \\ F &\to (E) \mid a \end{aligned}$$

	T	F	a			
	$+$	$+$	$+$	$+$		
E	E	E	E	E	E	T

Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\rightarrow T + E \mid T \\ T &\rightarrow F * T \mid F \\ F &\rightarrow (E) \mid a \end{aligned}$$

	T	F	a				F
	$+$	$+$	$+$	$+$			$*$
E	E	E	E	E	E	T	T

Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\to T + E \mid T \\ T &\to F * T \mid F \\ F &\to (E) \mid a \end{aligned}$$

	T	F	a				F	a
	$+$	$+$	$+$	$+$			$*$	$*$
E	E	E	E	E	E	T	T	T

Exemple

Reconnaissance du mot :

$$a + a \textcolor{red}{*} a$$

avec la grammaire :

$$\begin{aligned} E &\rightarrow T + E \mid T \\ T &\rightarrow F * T \mid F \\ F &\rightarrow (E) \mid a \end{aligned}$$

	T	F	a				F	a	
	$+$	$+$	$+$	$+$			$*$	$*$	$*$
E	E	E	E	E	E	T	T	T	T

Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\rightarrow T + E \mid T \\ T &\rightarrow F * T \mid F \\ F &\rightarrow (E) \mid a \end{aligned}$$

	T	F	a				F	a		
	$+$	$+$	$+$	$+$			$*$	$*$	$*$	
E	E	E	E	E	E	T	T	T	T	T

Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\to T + E \mid T \\ T &\to F * T \mid F \\ F &\to (E) \mid a \end{aligned}$$

	T	F	a				F	a			
	$+$	$+$	$+$	$+$			$*$	$*$	$*$		
E	E	E	E	E	E	T	T	T	T	T	F

Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\to T + E \mid T \\ T &\to F * T \mid F \\ F &\to (E) \mid a \end{aligned}$$

	T	F	a				F	a				
	$+$	$+$	$+$	$+$			$*$	$*$	$*$			
E	E	E	E	E	E	T	T	T	T	T	F	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'algorithme est non déterministe.

Analyse Ascendante ou analyse par décalage-réduction

- On empile les terminaux au fur et à mesure qu'ils sont lus.
- L'opération qui consiste à empiler un terminal est appelée décalage.
- lorsque les k symboles au sommet de la pile constituent la partie droite d'une production, ils peuvent être dépilés et remplacés par la partie gauche de la production.
- Cette opération s'appelle réduction.
- La séquence de symboles dépilés s'appelle un manche
- Lorsque la pile ne comporte que l'axiome et que tous les symboles de la chaîne d'entrée ont été lus, l'analyse a réussi.

Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\to T + E \mid T \\ T &\to F * T \mid F \\ F &\to (E) \mid a \end{aligned}$$

$\square$

Exemple

Reconnaissance du mot :

$$a \textcolor{red}{+} a * a$$

avec la grammaire :

$$\begin{aligned} E &\rightarrow T + E \mid T \\ T &\rightarrow F * T \mid F \\ F &\rightarrow (E) \mid a \end{aligned}$$

$\boxed{}$ $\boxed{a}$

Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\rightarrow T + E \mid T \\ T &\rightarrow F * T \mid F \\ F &\rightarrow (E) \mid a \end{aligned}$$

$\boxed{}$ $\boxed{a}$

Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\rightarrow T + E \mid T \\ T &\rightarrow F * T \mid F \\ F &\rightarrow (E) \mid a \end{aligned}$$

$\boxed{}$ $\boxed{a}$ $\boxed{F}$

Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\rightarrow T + E \mid T \\ T &\rightarrow F * T \mid F \\ F &\rightarrow (E) \mid a \end{aligned}$$

$\boxed{} \quad \boxed{a} \quad \boxed{F}$

Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\rightarrow T + E \mid T \\ T &\rightarrow F * T \mid F \\ F &\rightarrow (E) \mid a \end{aligned}$$

$\boxed{}$ $\boxed{a}$ $\boxed{F}$ $\boxed{T}$

Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\to T + E \mid T \\ T &\to F * T \mid F \\ F &\to (E) \mid a \end{aligned}$$

$$\boxed{} \quad \boxed{a} \quad \boxed{F} \quad \boxed{T} \quad \begin{array}{|c|}\hline + \\ \hline T \\ \hline \end{array}$$

Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\rightarrow T + E \mid T \\ T &\rightarrow F * T \mid F \\ F &\rightarrow (E) \mid a \end{aligned}$$

					a
				$+$	$+$
	a	F	T	T	T

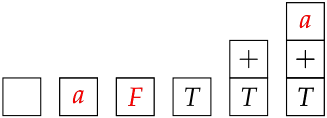
Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\rightarrow T + E \mid T \\ T &\rightarrow F * T \mid F \\ F &\rightarrow (E) \mid a \end{aligned}$$



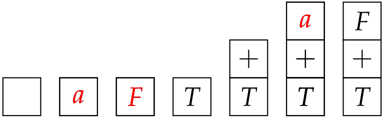
Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\to T + E \mid T \\ T &\to F * T \mid F \\ F &\to (E) \mid a \end{aligned}$$



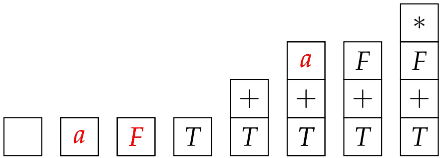
Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\to T + E \mid T \\ T &\to F * T \mid F \\ F &\to (E) \mid a \end{aligned}$$



Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\rightarrow T + E \mid T \\ T &\rightarrow F * T \mid F \\ F &\rightarrow (E) \mid a \end{aligned}$$

								a
							$*$	$*$
					a	F	F	F
				$+$	$+$	$+$	$+$	$+$
	a	F	T	T	T	T	T	T

Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\to T + E \mid T \\ T &\to F * T \mid F \\ F &\to (E) \mid a \end{aligned}$$

								a
							$*$	$*$
					a	F	F	F
				$+$	$+$	$+$	$+$	$+$
	a	F	T	T	T	T	T	T

Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\rightarrow T + E \mid T \\ T &\rightarrow F * T \mid F \\ F &\rightarrow (E) \mid a \end{aligned}$$

								a	F
							$*$	$*$	$*$
					a	F	F	F	F
				$+$	$+$	$+$	$+$	$+$	$+$
	a	F	T	T	T	T	T	T	T

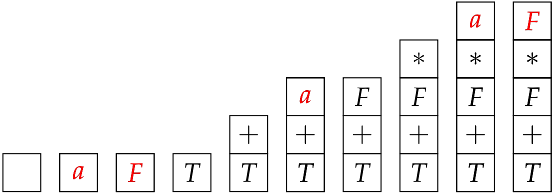
Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\to T + E \mid T \\ T &\to F * T \mid F \\ F &\to (E) \mid a \end{aligned}$$



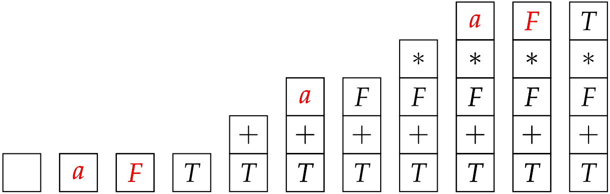
Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\to T + E \mid T \\ T &\to F * T \mid F \\ F &\to (E) \mid a \end{aligned}$$



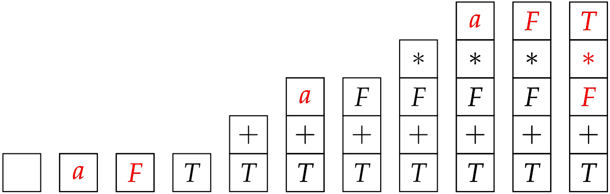
Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\to T + E \mid T \\ T &\to F * T \mid F \\ F &\to (E) \mid a \end{aligned}$$



Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\to T + E \mid T \\ T &\to F * T \mid F \\ F &\to (E) \mid a \end{aligned}$$

								a	F	T	
							$*$	$*$	$*$	$*$	
					a	F	F	F	F	F	T
				$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$
	a	F	T	T	T	T	T	T	T	T	T

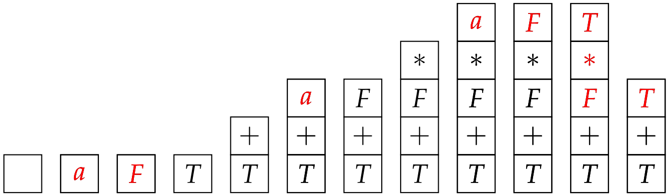
Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\rightarrow T + E \mid T \\ T &\rightarrow F * T \mid F \\ F &\rightarrow (E) \mid a \end{aligned}$$



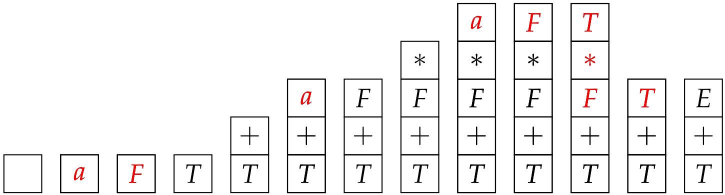
Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\to T + E \mid T \\ T &\to F * T \mid F \\ F &\to (E) \mid a \end{aligned}$$



Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\to T + E \mid T \\ T &\to F * T \mid F \\ F &\to (E) \mid a \end{aligned}$$

								a	F	T		
							$*$	$*$	$*$	$*$		
					a	F	F	F	F	F	T	E
				$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$
	a	F	T	T	T	T	T	T	T	T	T	T

Exemple

Reconnaissance du mot :

$$a + a * a$$

avec la grammaire :

$$\begin{aligned} E &\to T + E \mid T \\ T &\to F * T \mid F \\ F &\to (E) \mid a \end{aligned}$$

								a	F	T			
							$*$	$*$	$*$	$*$			
					a	F	F	F	F	F	T	E	
				$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$	
	a	F	T	T	T	T	T	T	T	T	T	T	E

Non déterminisme

- Si les symboles au sommet de la pile constituent la partie droite de deux productions distinctes alors chacune de ces deux règles peut être utilisée pour effectuer une réduction.
- Lorsque les symboles au sommet de la pile constituent la partie droite d'une ou plusieurs productions, on peut réduire tout de suite ou bien continuer à décaler, afin de permettre ultérieurement une réduction plus juste.

Automate à pile

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

Grammaires hors-contexte $\Leftrightarrow$ Automate à pile

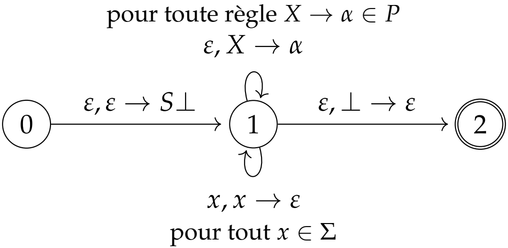
Un langage est hors-contexte si et seulement si il existe un automate à pile qui le reconnaît.

- 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 gauche

- 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 gauche conduisant de S à m.
- Idée clef : écrire dans la pile de A les proto-mots qui constituent la dérivation recherchée.

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



Construction de l'automate gauche

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

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

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

- $\delta(0, \varepsilon, \varepsilon) = \{(1, S\perp)\}$ On empile l'axiome.
- $\delta(1, \varepsilon, X) = \{(1, \alpha)$ pour tout $X \to \alpha \in P\}$
  Si un symbole non terminal X occupe le sommet de la pile, on le remplace par la partie droite α d'une règle $X \to \alpha$.
- $\delta(1, a, a) = \{(1, \varepsilon) \mid$ 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.
- $\delta(1, \varepsilon, \perp) = \{(2, \varepsilon)\}$
  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.

Construction — Exemple

Grammaire :

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

avec :

$$P = \left\{ \begin{array}{lcl} E & \to & T + E \mid T, \\ T & \to & F * T \mid F, \\ F & \to & (E) \mid a \end{array} \right\}$$

Automate :

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

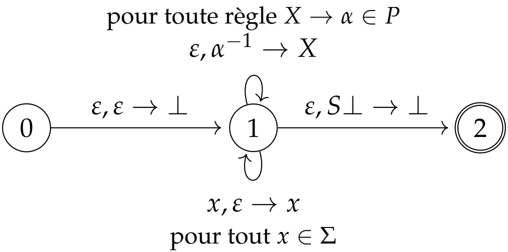
avec :

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

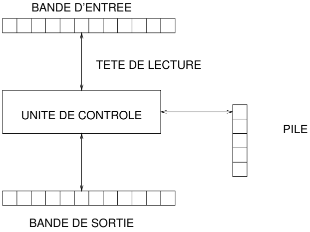
Grammaires hors-contexte $\Rightarrow$ Automate droit

- 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 ($S \stackrel{+}{\Rightarrow} m$).
- A est conçu de telle sorte à déterminer une réduction droite conduisant de m à S.

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



Transducteur 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

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

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

Analyseur gauche

$$\begin{array}{llllllll} 1: & E & \rightarrow & T+E & \quad 2: & E & \rightarrow & T \\ 3: & T & \rightarrow & F*T & \quad 4: & T & \rightarrow & F \\ 5: & F & \rightarrow & (E) & \quad 6: & F & \rightarrow & a \end{array}$$

- Dérivation gauche de $a + a * a$:
  $E \Rightarrow T + E \Rightarrow F + E \Rightarrow a + E \Rightarrow a + T \stackrel{*}{\Rightarrow} a + a * a$
- Elle correspond à l'application des règles suivantes :
  $1, 4, 6, 2, 3, 6, 4, 6$

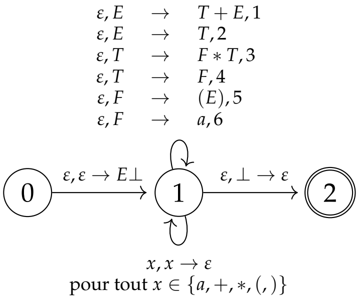
Analyseur gauche

Soit une grammaire hors contexte 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 : $\mathcal{O}(c^{|m|})$

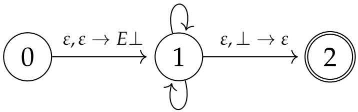
Analyseur gauche : Exemple



Analyseur gauche : Exemple

$$(0, \quad a + a * a, \qquad\qquad \varepsilon, \quad \varepsilon)$$

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

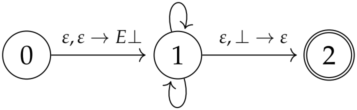


$x, x \to \varepsilon$
pour tout $x \in \{a, +, *, (,)\}$

Analyseur gauche : Exemple

$$
\begin{array}{llrl}
(0, & a + a * a, & \varepsilon, & \varepsilon) \\
(1, & a + a * a, & E\bot, & \varepsilon)
\end{array}
$$

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

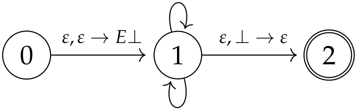


$x, x \to \varepsilon$
pour tout $x \in \{a, +, *, (,)\}$

Analyseur gauche : Exemple

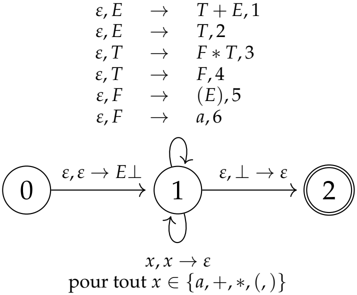
$$
\begin{array}{llll}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & E\bot, & \varepsilon) \\
(1, & a+a*a, & T+E\bot, & 1)
\end{array}
$$

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



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

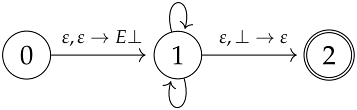
Analyseur gauche : Exemple



$$
\begin{array}{llrl}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & E\perp, & \varepsilon) \\
(1, & a+a*a, & T+E\perp, & 1) \\
(1, & a+a*a, & F+E\perp, & 14)
\end{array}
$$

Analyseur gauche : Exemple

$$
\begin{array}{lcl}
\varepsilon, E & \rightarrow & T+E, 1 \\
\varepsilon, E & \rightarrow & T, 2 \\
\varepsilon, T & \rightarrow & F*T, 3 \\
\varepsilon, T & \rightarrow & F, 4 \\
\varepsilon, F & \rightarrow & (E), 5 \\
\varepsilon, F & \rightarrow & a, 6
\end{array}
$$

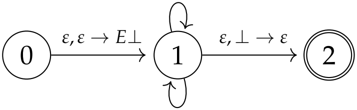


$x, x \rightarrow \varepsilon$
pour tout $x \in \{a, +, *, (,)\}$

$$
\begin{array}{llrl}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & E\bot, & \varepsilon) \\
(1, & a+a*a, & T+E\bot, & 1) \\
(1, & a+a*a, & F+E\bot, & 14) \\
(1, & a+a*a, & a+E\bot, & 146)
\end{array}
$$

Analyseur gauche : Exemple

$$\begin{array}{lcl}
\varepsilon, E & \rightarrow & T+E, 1 \\
\varepsilon, E & \rightarrow & T, 2 \\
\varepsilon, T & \rightarrow & F*T, 3 \\
\varepsilon, T & \rightarrow & F, 4 \\
\varepsilon, F & \rightarrow & (E), 5 \\
\varepsilon, F & \rightarrow & a, 6
\end{array}$$

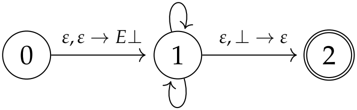


$x, x \rightarrow \varepsilon$
pour tout $x \in \{a, +, *, (,)\}$

$$\begin{array}{llrl}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & E\perp, & \varepsilon) \\
(1, & a+a*a, & T+E\perp, & 1) \\
(1, & a+a*a, & F+E\perp, & 14) \\
(1, & a+a*a, & a+E\perp, & 146) \\
(1, & +a*a, & +E\perp, & 146)
\end{array}$$

Analyseur gauche : Exemple

$$\begin{array}{lcl}
\varepsilon, E & \rightarrow & T+E, 1 \\
\varepsilon, E & \rightarrow & T, 2 \\
\varepsilon, T & \rightarrow & F*T, 3 \\
\varepsilon, T & \rightarrow & F, 4 \\
\varepsilon, F & \rightarrow & (E), 5 \\
\varepsilon, F & \rightarrow & a, 6
\end{array}$$

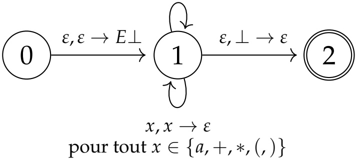


$x, x \rightarrow \varepsilon$
pour tout $x \in \{a, +, *, (,)\}$

$$\begin{array}{llrl}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & E\perp, & \varepsilon) \\
(1, & a+a*a, & T+E\perp, & 1) \\
(1, & a+a*a, & F+E\perp, & 14) \\
(1, & a+a*a, & a+E\perp, & 146) \\
(1, & +a*a, & +E\perp, & 146) \\
(1, & a*a, & E\perp, & 146)
\end{array}$$

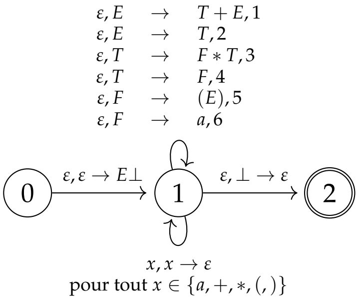
Analyseur gauche : Exemple

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



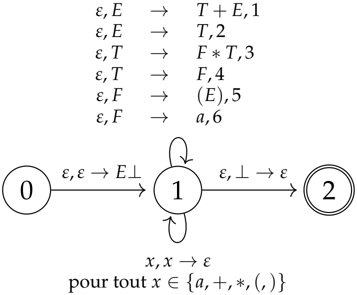
$$\begin{array}{llrl}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & E\perp, & \varepsilon) \\
(1, & a+a*a, & T+E\perp, & 1) \\
(1, & a+a*a, & F+E\perp, & 14) \\
(1, & a+a*a, & a+E\perp, & 146) \\
(1, & +a*a, & +E\perp, & 146) \\
(1, & a*a, & E\perp, & 146) \\
(1, & a*a, & T\perp, & 1462)
\end{array}$$

Analyseur gauche : Exemple



$$
\begin{array}{llrl}
(0, & a + a * a, & \varepsilon, & \varepsilon) \\
(1, & a + a * a, & E\bot, & \varepsilon) \\
(1, & a + a * a, & T + E\bot, & 1) \\
(1, & a + a * a, & F + E\bot, & 14) \\
(1, & a + a * a, & a + E\bot, & 146) \\
(1, & +a * a, & +E\bot, & 146) \\
(1, & a * a, & E\bot, & 146) \\
(1, & a * a, & T\bot, & 1462) \\
(1, & a * a, & F * T\bot, & 14623)
\end{array}
$$

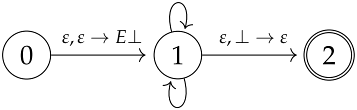
Analyseur gauche : Exemple



$$
\begin{array}{llrl}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & E\bot, & \varepsilon) \\
(1, & a+a*a, & T+E\bot, & 1) \\
(1, & a+a*a, & F+E\bot, & 14) \\
(1, & a+a*a, & a+E\bot, & 146) \\
(1, & +a*a, & +E\bot, & 146) \\
(1, & a*a, & E\bot, & 146) \\
(1, & a*a, & T\bot, & 1462) \\
(1, & a*a, & F*T\bot, & 14623) \\
(1, & a*a, & a*T\bot, & 146236)
\end{array}
$$

Analyseur gauche : Exemple

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

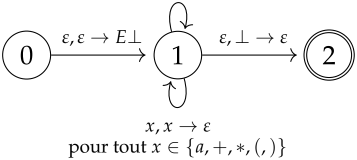


$x, x \to \varepsilon$
pour tout $x \in \{a, +, *, (,)\}$

$$\begin{array}{llrl}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & E\perp, & \varepsilon) \\
(1, & a+a*a, & T+E\perp, & 1) \\
(1, & a+a*a, & F+E\perp, & 14) \\
(1, & a+a*a, & a+E\perp, & 146) \\
(1, & +a*a, & +E\perp, & 146) \\
(1, & a*a, & E\perp, & 146) \\
(1, & a*a, & T\perp, & 1462) \\
(1, & a*a, & F*T\perp, & 14623) \\
(1, & a*a, & a*T\perp, & 146236) \\
(1, & *a, & *T\perp, & 146236)
\end{array}$$

Analyseur gauche : Exemple

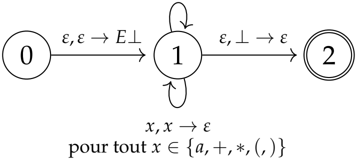
$$\begin{array}{lcl}
\varepsilon, E & \rightarrow & T+E, 1 \\
\varepsilon, E & \rightarrow & T, 2 \\
\varepsilon, T & \rightarrow & F*T, 3 \\
\varepsilon, T & \rightarrow & F, 4 \\
\varepsilon, F & \rightarrow & (E), 5 \\
\varepsilon, F & \rightarrow & a, 6
\end{array}$$



$$\begin{array}{llrl}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & E\bot, & \varepsilon) \\
(1, & a+a*a, & T+E\bot, & 1) \\
(1, & a+a*a, & F+E\bot, & 14) \\
(1, & a+a*a, & a+E\bot, & 146) \\
(1, & +a*a, & +E\bot, & 146) \\
(1, & a*a, & E\bot, & 146) \\
(1, & a*a, & T\bot, & 1462) \\
(1, & a*a, & F*T\bot, & 14623) \\
(1, & a*a, & a*T\bot, & 146236) \\
(1, & *a, & *T\bot, & 146236) \\
(1, & a, & T\bot, & 1462364)
\end{array}$$

Analyseur gauche : Exemple

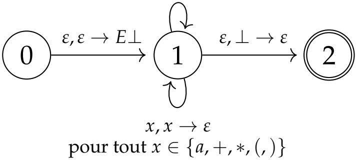
$$
\begin{array}{lcl}
\varepsilon, E & \to & T+E, 1 \\
\varepsilon, E & \to & T, 2 \\
\varepsilon, T & \to & F*T, 3 \\
\varepsilon, T & \to & F, 4 \\
\varepsilon, F & \to & (E), 5 \\
\varepsilon, F & \to & a, 6
\end{array}
$$



$$
\begin{array}{llrl}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & E\perp, & \varepsilon) \\
(1, & a+a*a, & T+E\perp, & 1) \\
(1, & a+a*a, & F+E\perp, & 14) \\
(1, & a+a*a, & a+E\perp, & 146) \\
(1, & +a*a, & +E\perp, & 146) \\
(1, & a*a, & E\perp, & 146) \\
(1, & a*a, & T\perp, & 1462) \\
(1, & a*a, & F*T\perp, & 14623) \\
(1, & a*a, & a*T\perp, & 146236) \\
(1, & *a, & *T\perp, & 146236) \\
(1, & a, & T\perp, & 1462364) \\
(1, & a, & F\perp, & 14623646)
\end{array}
$$

Analyseur gauche : Exemple

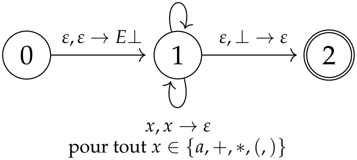
$$
\begin{array}{lcl}
\varepsilon, E & \to & T+E, 1 \\
\varepsilon, E & \to & T, 2 \\
\varepsilon, T & \to & F*T, 3 \\
\varepsilon, T & \to & F, 4 \\
\varepsilon, F & \to & (E), 5 \\
\varepsilon, F & \to & a, 6
\end{array}
$$



$$
\begin{array}{llrl}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & E\bot, & \varepsilon) \\
(1, & a+a*a, & T+E\bot, & 1) \\
(1, & a+a*a, & F+E\bot, & 14) \\
(1, & a+a*a, & a+E\bot, & 146) \\
(1, & +a*a, & +E\bot, & 146) \\
(1, & a*a, & E\bot, & 146) \\
(1, & a*a, & T\bot, & 1462) \\
(1, & a*a, & F*T\bot, & 14623) \\
(1, & a*a, & a*T\bot, & 146236) \\
(1, & *a, & *T\bot, & 146236) \\
(1, & a, & T\bot, & 1462364) \\
(1, & a, & F\bot, & 14623646) \\
(1, & a, & a\bot, & 14623646)
\end{array}
$$

Analyseur gauche : Exemple

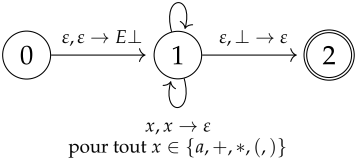
$$
\begin{array}{lcl}
\varepsilon, E & \to & T+E, 1 \\
\varepsilon, E & \to & T, 2 \\
\varepsilon, T & \to & F*T, 3 \\
\varepsilon, T & \to & F, 4 \\
\varepsilon, F & \to & (E), 5 \\
\varepsilon, F & \to & a, 6
\end{array}
$$



$$
\begin{array}{llrl}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & E\bot, & \varepsilon) \\
(1, & a+a*a, & T+E\bot, & 1) \\
(1, & a+a*a, & F+E\bot, & 14) \\
(1, & a+a*a, & a+E\bot, & 146) \\
(1, & +a*a, & +E\bot, & 146) \\
(1, & a*a, & E\bot, & 146) \\
(1, & a*a, & T\bot, & 1462) \\
(1, & a*a, & F*T\bot, & 14623) \\
(1, & a*a, & a*T\bot, & 146236) \\
(1, & *a, & *T\bot, & 146236) \\
(1, & a, & T\bot, & 1462364) \\
(1, & a, & F\bot, & 14623646) \\
(1, & a, & a\bot, & 14623646) \\
(1, & \varepsilon, & \bot, & 14623646)
\end{array}
$$

Analyseur gauche : Exemple

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



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

Analyseur droit

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

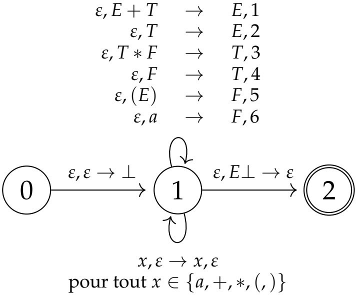
- Réduction droite de $a+a*a$:
  $a+a*a \rightarrow F+a*a \rightarrow T+a*a \rightarrow T+F*a \rightarrow T+F*F \rightarrow T+F*T \rightarrow T+T \rightarrow T+E \rightarrow E$
- Elle correspond à l'application des règles suivantes :
  $6,4,6,6,4,3,2,1$

Analyseur droit

Soit une grammaire hors contexte G dont les règles ont été numérotées de 1 à p. On appelle un analyseur droit de G, un transducteur à pile non déterministe T_G^d qui produit pour un mot $m \in L(G)$, une dérivation droite de m à l'envers.
Performances :

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

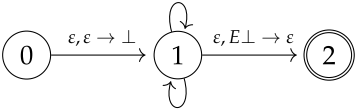
Analyseur droit : Exemple



Analyseur droit : Exemple

$$(0, \quad a + a * a, \qquad \varepsilon, \quad \varepsilon)$$

$$\begin{array}{rcl}
\varepsilon, E + T & \rightarrow & E, 1 \\
\varepsilon, T & \rightarrow & E, 2 \\
\varepsilon, T * F & \rightarrow & T, 3 \\
\varepsilon, F & \rightarrow & T, 4 \\
\varepsilon, (E) & \rightarrow & F, 5 \\
\varepsilon, a & \rightarrow & F, 6
\end{array}$$

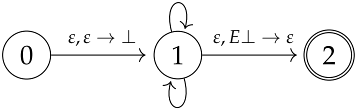


$x, \varepsilon \rightarrow x, \varepsilon$
pour tout $x \in \{a, +, *, (,)\}$

Analyseur droit : Exemple

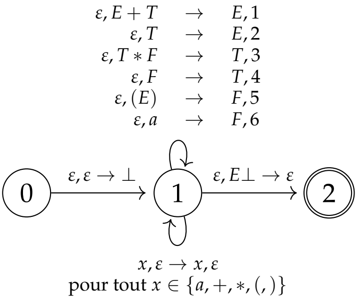
$$\begin{array}{llll} (0, & a+a*a, & \varepsilon, & \varepsilon) \\ (1, & a+a*a, & \perp, & \varepsilon) \end{array}$$

$$\begin{array}{rcl} \varepsilon, E+T & \rightarrow & E, 1 \\ \varepsilon, T & \rightarrow & E, 2 \\ \varepsilon, T*F & \rightarrow & T, 3 \\ \varepsilon, F & \rightarrow & T, 4 \\ \varepsilon, (E) & \rightarrow & F, 5 \\ \varepsilon, a & \rightarrow & F, 6 \end{array}$$



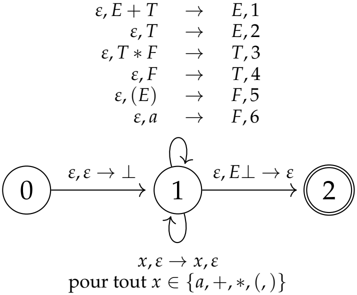
$x, \varepsilon \rightarrow x, \varepsilon$
pour tout $x \in \{a, +, *, (,)\}$

Analyseur droit : Exemple



$$
\begin{array}{llrr}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & \perp, & \varepsilon) \\
(1, & +a*a, & a\perp, & 6)
\end{array}
$$

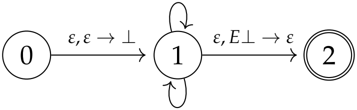
Analyseur droit : Exemple



$$
\begin{array}{llrl}
(0, & a + a * a, & \varepsilon, & \varepsilon) \\
(1, & a + a * a, & \perp, & \varepsilon) \\
(1, & +a * a, & a\perp, & 6) \\
(1, & +a * a, & F\perp, & 64)
\end{array}
$$

Analyseur droit : Exemple

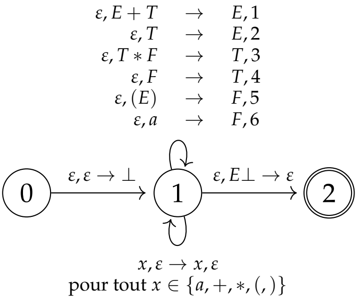
$$
\begin{array}{rcl}
\varepsilon, E+T & \rightarrow & E, 1 \\
\varepsilon, T & \rightarrow & E, 2 \\
\varepsilon, T*F & \rightarrow & T, 3 \\
\varepsilon, F & \rightarrow & T, 4 \\
\varepsilon, (E) & \rightarrow & F, 5 \\
\varepsilon, a & \rightarrow & F, 6
\end{array}
$$



pour tout $x \in \{a, +, *, (,)\}$

$$
\begin{array}{llrl}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & \perp, & \varepsilon) \\
(1, & +a*a, & a\perp, & 6) \\
(1, & +a*a, & F\perp, & 64) \\
(1, & +a*a, & T\perp, & 64)
\end{array}
$$

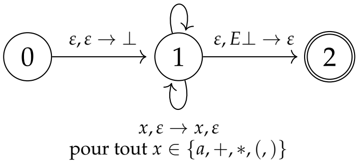
Analyseur droit : Exemple



$$
\begin{array}{llrl}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & \perp, & \varepsilon) \\
(1, & +a*a, & a\perp, & 6) \\
(1, & +a*a, & F\perp, & 64) \\
(1, & +a*a, & T\perp, & 64) \\
(1, & a*a, & +T\perp, & 64)
\end{array}
$$

Analyseur droit : Exemple

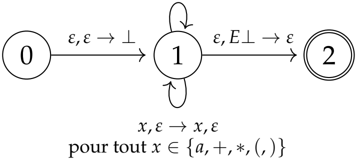
$$
\begin{array}{rcl}
\varepsilon, E+T & \to & E,1 \\
\varepsilon, T & \to & E,2 \\
\varepsilon, T*F & \to & T,3 \\
\varepsilon, F & \to & T,4 \\
\varepsilon, (E) & \to & F,5 \\
\varepsilon, a & \to & F,6
\end{array}
$$



$$
\begin{array}{llrl}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & \perp, & \varepsilon) \\
(1, & +a*a, & a\perp, & 6) \\
(1, & +a*a, & F\perp, & 64) \\
(1, & +a*a, & T\perp, & 64) \\
(1, & a*a, & +T\perp, & 64) \\
(1, & *a, & a+T\perp, & 646)
\end{array}
$$

Analyseur droit : Exemple

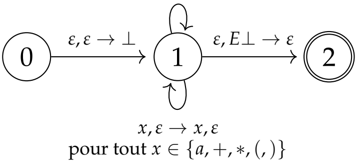
$$
\begin{array}{rcl}
\varepsilon, E+T & \rightarrow & E, 1 \\
\varepsilon, T & \rightarrow & E, 2 \\
\varepsilon, T*F & \rightarrow & T, 3 \\
\varepsilon, F & \rightarrow & T, 4 \\
\varepsilon, (E) & \rightarrow & F, 5 \\
\varepsilon, a & \rightarrow & F, 6
\end{array}
$$



$$
\begin{array}{llrl}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & \perp, & \varepsilon) \\
(1, & +a*a, & a\perp, & 6) \\
(1, & +a*a, & F\perp, & 64) \\
(1, & +a*a, & T\perp, & 64) \\
(1, & a*a, & +T\perp, & 64) \\
(1, & *a, & a+T\perp, & 646) \\
(1, & *a, & F+T\perp, & 646)
\end{array}
$$

Analyseur droit : Exemple

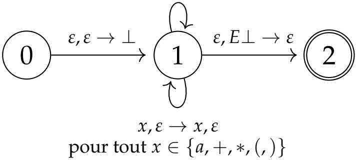
$$
\begin{array}{rcl}
\varepsilon, E+T & \to & E, 1 \\
\varepsilon, T & \to & E, 2 \\
\varepsilon, T*F & \to & T, 3 \\
\varepsilon, F & \to & T, 4 \\
\varepsilon, (E) & \to & F, 5 \\
\varepsilon, a & \to & F, 6
\end{array}
$$



$$
\begin{array}{llrl}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & \perp, & \varepsilon) \\
(1, & +a*a, & a\perp, & 6) \\
(1, & +a*a, & F\perp, & 64) \\
(1, & +a*a, & T\perp, & 64) \\
(1, & a*a, & +T\perp, & 64) \\
(1, & *a, & a+T\perp, & 646) \\
(1, & *a, & F+T\perp, & 646) \\
(1, & a, & *F+T\perp, & 646)
\end{array}
$$

Analyseur droit : Exemple

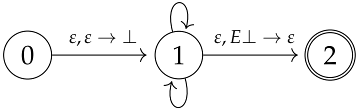
$$\begin{array}{rcl}
\varepsilon, E+T & \to & E, 1 \\
\varepsilon, T & \to & E, 2 \\
\varepsilon, T*F & \to & T, 3 \\
\varepsilon, F & \to & T, 4 \\
\varepsilon, (E) & \to & F, 5 \\
\varepsilon, a & \to & F, 6
\end{array}$$



$$\begin{array}{llrl}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & \perp, & \varepsilon) \\
(1, & +a*a, & a\perp, & 6) \\
(1, & +a*a, & F\perp, & 64) \\
(1, & +a*a, & T\perp, & 64) \\
(1, & a*a, & +T\perp, & 64) \\
(1, & *a, & a+T\perp, & 646) \\
(1, & *a, & F+T\perp, & 646) \\
(1, & a, & *F+T\perp, & 646) \\
(1, & \varepsilon, & a*F+T\perp, & 6466)
\end{array}$$

Analyseur droit : Exemple

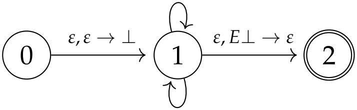
$$
\begin{array}{rcl}
\varepsilon, E+T & \to & E, 1 \\
\varepsilon, T & \to & E, 2 \\
\varepsilon, T*F & \to & T, 3 \\
\varepsilon, F & \to & T, 4 \\
\varepsilon, (E) & \to & F, 5 \\
\varepsilon, a & \to & F, 6
\end{array}
$$



$$
\begin{array}{llrl}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & \perp, & \varepsilon) \\
(1, & +a*a, & a\perp, & 6) \\
(1, & +a*a, & F\perp, & 64) \\
(1, & +a*a, & T\perp, & 64) \\
(1, & a*a, & +T\perp, & 64) \\
(1, & *a, & a+T\perp, & 646) \\
(1, & *a, & F+T\perp, & 646) \\
(1, & a, & *F+T\perp, & 646) \\
(1, & \varepsilon, & a*F+T\perp, & 6466) \\
(1, & \varepsilon, & F*F+T\perp, & 64664)
\end{array}
$$

Analyseur droit : Exemple

$$\begin{array}{rcl}
\varepsilon, E+T & \rightarrow & E,1 \\
\varepsilon, T & \rightarrow & E,2 \\
\varepsilon, T*F & \rightarrow & T,3 \\
\varepsilon, F & \rightarrow & T,4 \\
\varepsilon, (E) & \rightarrow & F,5 \\
\varepsilon, a & \rightarrow & F,6
\end{array}$$

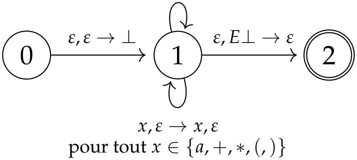


$x, \varepsilon \rightarrow x, \varepsilon$
pour tout $x \in \{a, +, *, (,)\}$

$$\begin{array}{llrl}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & \perp, & \varepsilon) \\
(1, & +a*a, & a\perp, & 6) \\
(1, & +a*a, & F\perp, & 64) \\
(1, & +a*a, & T\perp, & 64) \\
(1, & a*a, & +T\perp, & 64) \\
(1, & *a, & a+T\perp, & 646) \\
(1, & *a, & F+T\perp, & 646) \\
(1, & a, & *F+T\perp, & 646) \\
(1, & \varepsilon, & a*F+T\perp, & 6466) \\
(1, & \varepsilon, & F*F+T\perp, & 64664) \\
(1, & \varepsilon, & T*F+T\perp, & 646643)
\end{array}$$

Analyseur droit : Exemple

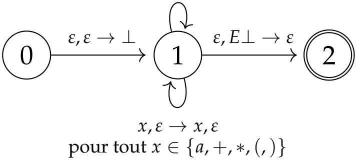
$$\begin{array}{rcl}
\varepsilon, E+T & \to & E,1 \\
\varepsilon, T & \to & E,2 \\
\varepsilon, T*F & \to & T,3 \\
\varepsilon, F & \to & T,4 \\
\varepsilon, (E) & \to & F,5 \\
\varepsilon, a & \to & F,6
\end{array}$$



$(0,$	$a+a*a,$	$\varepsilon,$	$\varepsilon)$
$(1,$	$a+a*a,$	$\perp,$	$\varepsilon)$
$(1,$	$+a*a,$	$a\perp,$	$6)$
$(1,$	$+a*a,$	$F\perp,$	$64)$
$(1,$	$+a*a,$	$T\perp,$	$64)$
$(1,$	$a*a,$	$+T\perp,$	$64)$
$(1,$	$*a,$	$a+T\perp,$	$646)$
$(1,$	$*a,$	$F+T\perp,$	$646)$
$(1,$	$a,$	$*F+T\perp,$	$646)$
$(1,$	$\varepsilon,$	$a*F+T\perp,$	$6466)$
$(1,$	$\varepsilon,$	$F*F+T\perp,$	$64664)$
$(1,$	$\varepsilon,$	$T*F+T\perp,$	$646643)$
$(1,$	$\varepsilon,$	$T+T\perp,$	$6466432)$

Analyseur droit : Exemple

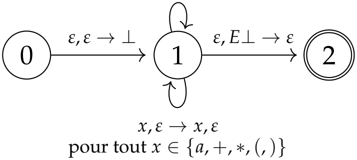
$$
\begin{array}{rcl}
\varepsilon, E+T & \to & E, 1 \\
\varepsilon, T & \to & E, 2 \\
\varepsilon, T*F & \to & T, 3 \\
\varepsilon, F & \to & T, 4 \\
\varepsilon, (E) & \to & F, 5 \\
\varepsilon, a & \to & F, 6
\end{array}
$$



$$
\begin{array}{llrl}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & \perp, & \varepsilon) \\
(1, & +a*a, & a\perp, & 6) \\
(1, & +a*a, & F\perp, & 64) \\
(1, & +a*a, & T\perp, & 64) \\
(1, & a*a, & +T\perp, & 64) \\
(1, & *a, & a+T\perp, & 646) \\
(1, & *a, & F+T\perp, & 646) \\
(1, & a, & *F+T\perp, & 646) \\
(1, & \varepsilon, & a*F+T\perp, & 6466) \\
(1, & \varepsilon, & F*F+T\perp, & 64664) \\
(1, & \varepsilon, & T*F+T\perp, & 646643) \\
(1, & \varepsilon, & T+T\perp, & 6466432) \\
(1, & \varepsilon, & E+T\perp, & 64664321)
\end{array}
$$

Analyseur droit : Exemple

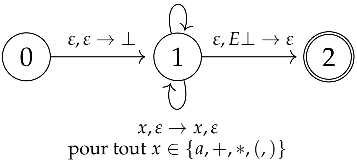
$$
\begin{array}{rcl}
\varepsilon, E+T & \to & E, 1 \\
\varepsilon, T & \to & E, 2 \\
\varepsilon, T*F & \to & T, 3 \\
\varepsilon, F & \to & T, 4 \\
\varepsilon, (E) & \to & F, 5 \\
\varepsilon, a & \to & F, 6
\end{array}
$$



$$
\begin{array}{llrl}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & \perp, & \varepsilon) \\
(1, & +a*a, & a\perp, & 6) \\
(1, & +a*a, & F\perp, & 64) \\
(1, & +a*a, & T\perp, & 64) \\
(1, & a*a, & +T\perp, & 64) \\
(1, & *a, & a+T\perp, & 646) \\
(1, & *a, & F+T\perp, & 646) \\
(1, & a, & *F+T\perp, & 646) \\
(1, & \varepsilon, & a*F+T\perp, & 6466) \\
(1, & \varepsilon, & F*F+T\perp, & 64664) \\
(1, & \varepsilon, & T*F+T\perp, & 646643) \\
(1, & \varepsilon, & T+T\perp, & 6466432) \\
(1, & \varepsilon, & E+T\perp, & 64664321) \\
(1, & \varepsilon, & E\perp, & 64664321)
\end{array}
$$

Analyseur droit : Exemple

$$
\begin{array}{rcl}
\varepsilon, E+T & \rightarrow & E, 1 \\
\varepsilon, T & \rightarrow & E, 2 \\
\varepsilon, T*F & \rightarrow & T, 3 \\
\varepsilon, F & \rightarrow & T, 4 \\
\varepsilon, (E) & \rightarrow & F, 5 \\
\varepsilon, a & \rightarrow & F, 6
\end{array}
$$



$$
\begin{array}{llrl}
(0, & a+a*a, & \varepsilon, & \varepsilon) \\
(1, & a+a*a, & \perp, & \varepsilon) \\
(1, & +a*a, & a\perp, & 6) \\
(1, & +a*a, & F\perp, & 64) \\
(1, & +a*a, & T\perp, & 64) \\
(1, & a*a, & +T\perp, & 64) \\
(1, & *a, & a+T\perp, & 646) \\
(1, & *a, & F+T\perp, & 646) \\
(1, & a, & *F+T\perp, & 646) \\
(1, & \varepsilon, & a*F+T\perp, & 6466) \\
(1, & \varepsilon, & F*F+T\perp, & 64664) \\
(1, & \varepsilon, & T*F+T\perp, & 646643) \\
(1, & \varepsilon, & T+T\perp, & 6466432) \\
(1, & \varepsilon, & E+T\perp, & 64664321) \\
(1, & \varepsilon, & E\perp, & 64664321) \\
(2, & \varepsilon, & \varepsilon, & 64664321)
\end{array}
$$

Analyse déterministe

- L'automate non déterministe n'est pas utilisable
- **Idée générale** : rendre déterministe un analyseur gauche ou un analyseur droit en s'autorisant à regarder les k prochains symboles de la bande d'entrée (*lookahead*)
- La prochaine action à effectuer est indiquée par une table d'analyse
- Etant donné la configuration courante de l'automate et les k prochains symboles, elle indique l'action à effectuer.
- Si l'analyseur gauche correspondant à la grammaire peut être rendu déterministe dans ces conditions, alors on dit que la grammaire est $LL(k)$ (Left to right, Leftmost derivation)
- Si l'analyseur droit correspondant à la grammaire peut être rendu déterministe dans ces conditions, alors on dit que la grammaire est $LR(k)$ (Left to right, Rightmost derivation)

Grammaires $LR(k)$

- Une grammaire est $LR(k)$ s'il est possible d'effectuer une analyse par décalage-réduction déterministe en s'autorisant à lire les k symboles suivant le symbole courant.
- La grammaire suivante n'est pas $LR(1)$ mais elle est $LR(2)$:

$$\begin{array}{llllll}
1: & S & \rightarrow & X & b & c \\
2: & S & \rightarrow & Y & b & d \\
3: & X & \rightarrow & a & & \\
4: & Y & \rightarrow & a & &
\end{array}$$

Table SLR (Simple LR)

Etant donné une grammaire G, on peut construire à partir de G une table SLR.

Lorsque cette dernière ne contient pas de conflits, elle permet de réaliser une analyse déterministe.

état	action					goto	
	0	1	*	+	$	E	B
0	d1	d2				4	3
1	r4	r4	r4	r4	r4		
2	r5	r5	r5	r5	r5		
3	r3	r3	r3	r3	r3		
4			d5	d6	acc		
5	d1	d2					7
6	d1	d2					8
7	r2	r2	r2	r2	r2		
8	r1	r1	r1	r1	r1		

$$\begin{array}{llll}
1: & E & \rightarrow & E * B \\
2: & E & \rightarrow & E + B \\
3: & E & \rightarrow & B
\end{array}
\qquad
\begin{array}{llll}
4: & B & \rightarrow & 0 \\
5: & B & \rightarrow & 1
\end{array}$$

Structure et utilisation de la table SLR

La table d'analyse est composée de deux parties :

Structure et utilisation de la table SLR

La table d'analyse est composée de deux parties :

- Une partie ACTION$[e, a]$ où e est un état et a un terminal (ou le symbole \$).

Structure et utilisation de la table SLR

La table d'analyse est composée de deux parties :

- Une partie ACTION$[e, a]$ où e est un état et a un terminal (ou le symbole \$).
- Une partie GOTO$[e, A]$ où A est un non terminal.

ACTION$[i, a]$ peut prendre une des quatre formes suivantes :

Structure et utilisation de la table SLR

La table d'analyse est composée de deux parties :

- Une partie ACTION$[e, a]$ où e est un état et a un terminal (ou le symbole \$).
- Une partie GOTO$[e, A]$ où A est un non terminal.

ACTION$[i, a]$ peut prendre une des quatre formes suivantes :

1. de, où e est un état. L'analyseur effectue un décalage : il empile e et consomme une unité lexicale

Structure et utilisation de la table SLR

La table d'analyse est composée de deux parties :

- Une partie ACTION$[e, a]$ où e est un état et a un terminal (ou le symbole \$).
- Une partie GOTO$[e, A]$ où A est un non terminal.

ACTION$[i, a]$ peut prendre une des quatre formes suivantes :

1. de, où e est un état. L'analyseur effectue un décalage : il empile e et consomme une unité lexicale
2. rj, où j est le numéro de la règle $A \to \beta$. L'analyseur effectue une réduction :
   - il dépile $|\beta|$ symboles de la pile
   - l'état l est maintenant au sommet de la pile
   - il empile l'état m, qui correspond à l'entrée GOTO$[l, A]$

Structure et utilisation de la table SLR

La table d'analyse est composée de deux parties :

- Une partie ACTION$[e, a]$ où e est un état et a un terminal (ou le symbole \$).
- Une partie GOTO$[e, A]$ où A est un non terminal.

ACTION$[i, a]$ peut prendre une des quatre formes suivantes :

1. de, où e est un état. L'analyseur effectue un décalage : il empile e et consomme une unité lexicale
2. rj, où j est le numéro de la règle $A \to \beta$. L'analyseur effectue une réduction :
   - il dépile $|\beta|$ symboles de la pile
   - l'état l est maintenant au sommet de la pile
   - il empile l'état m, qui correspond à l'entrée GOTO$[l, A]$
3. `acc` : l'analyseur accepte l'entrée

Structure et utilisation de la table SLR

La table d'analyse est composée de deux parties :

- Une partie ACTION$[e, a]$ où e est un état et a un terminal (ou le symbole \$).
- Une partie GOTO$[e, A]$ où A est un non terminal.

ACTION$[i, a]$ peut prendre une des quatre formes suivantes :

1. de, où e est un état. L'analyseur effectue un décalage : il empile e et consomme une unité lexicale
2. rj, où j est le numéro de la règle $A \to \beta$. L'analyseur effectue une réduction :
   - il dépile $|\beta|$ symboles de la pile
   - l'état l est maintenant au sommet de la pile
   - il empile l'état m, qui correspond à l'entrée GOTO$[l, A]$
3. `acc` : l'analyseur accepte l'entrée
4. `err` : l'analyseur signale une erreur (les cases correspondantes sont en général vides)

Exemple d'utilisation de la table

état	action					goto	
	0	1	*	+	$	E	B
0	d1	d2				4	3
1	r4	r4	r4	r4	r4		
2	r5	r5	r5	r5	r5		
3	r3	r3	r3	r3	r3		
4			d5	d6	acc		
5	d1	d2					7
6	d1	d2					8
7	r2	r2	r2	r2	r2		
8	r1	r1	r1	r1	r1		

On lit le mot :

$$1 + 0\$$$

$$\begin{array}{|c|}\hline 0 \\ \hline \perp \\ \hline \end{array}$$

$$\begin{array}{llllllll} 1: & E & \rightarrow & E + B & 4: & B & \rightarrow & 0 \\ 2: & E & \rightarrow & E * B & 5: & B & \rightarrow & 1 \\ 3: & E & \rightarrow & B & & & & \end{array}$$

Exemple d'utilisation de la table

état	action					goto	
	0	1	*	+	\$	E	B
0	d1	d2				4	3
1	r4	r4	r4	r4	r4		
2	r5	r5	r5	r5	r5		
3	r3	r3	r3	r3	r3		
4			d5	d6	acc		
5	d1	d2					7
6	d1	d2					8
7	r2	r2	r2	r2	r2		
8	r1	r1	r1	r1	r1		

On lit le mot :

$1 + 0\$$

0
$\perp$

$$\begin{array}{llll}
1: & E & \rightarrow & E+B \\
2: & E & \rightarrow & E*B \\
3: & E & \rightarrow & B
\end{array}
\qquad
\begin{array}{llll}
4: & B & \rightarrow & 0 \\
5: & B & \rightarrow & 1
\end{array}$$

Exemple d'utilisation de la table

état	action					goto	
	0	1	*	+	$	E	B
0	d1	d2				4	3
1	r4	r4	r4	r4	r4		
2	r5	r5	r5	r5	r5		
3	r3	r3	r3	r3	r3		
4			d5	d6	acc		
5	d1	d2					7
6	d1	d2					8
7	r2	r2	r2	r2	r2		
8	r1	r1	r1	r1	r1		

On lit le mot :

$1 + 0\$$

2
0
$\perp$

$$
\begin{array}{llll@{\qquad}llll}
1: & E & \to & E + B & 4: & B & \to & 0 \\
2: & E & \to & E * B & 5: & B & \to & 1 \\
3: & E & \to & B & & & &
\end{array}
$$

Exemple d'utilisation de la table

état	action					goto	
	0	1	*	+	$	E	B
0	d1	d2				4	3
1	r4	r4	r4	r4	r4		
2	r5	r5	r5	r5	r5		
3	r3	r3	r3	r3	r3		
4			d5	d6	acc		
5	d1	d2					7
6	d1	d2					8
7	r2	r2	r2	r2	r2		
8	r1	r1	r1	r1	r1		

On lit le mot :

$$1 + 0\$$$

0
$\perp$

$$\begin{array}{llll} 1: & E & \rightarrow & E+B \\ 2: & E & \rightarrow & E*B \\ 3: & E & \rightarrow & B \end{array} \qquad \begin{array}{llll} 4: & B & \rightarrow & 0 \\ 5: & B & \rightarrow & 1 \end{array}$$

Exemple d'utilisation de la table

état	action					goto	
	0	1	*	+	$	E	B
0	d1	d2				4	3
1	r4	r4	r4	r4	r4		
2	r5	r5	r5	r5	r5		
3	r3	r3	r3	r3	r3		
4			d5	d6	acc		
5	d1	d2					7
6	d1	d2					8
7	r2	r2	r2	r2	r2		
8	r1	r1	r1	r1	r1		

On lit le mot :

1 + 0$

3
0
$\perp$

$$
\begin{array}{llll@{\qquad}llll}
1: & E & \to & E + B & 4: & B & \to & 0 \\
2: & E & \to & E * B & 5: & B & \to & 1 \\
3: & E & \to & B & & & &
\end{array}
$$

Exemple d'utilisation de la table

état	action					goto	
	0	1	*	+	$	E	B
0	d1	d2				4	3
1	r4	r4	r4	r4	r4		
2	r5	r5	r5	r5	r5		
3	r3	r3	r3	r3	r3		
4			d5	d6	acc		
5	d1	d2					7
6	d1	d2					8
7	r2	r2	r2	r2	r2		
8	r1	r1	r1	r1	r1		

On lit le mot :

$1 + 0\$$

0
$\perp$

$$
\begin{array}{llll@{\qquad}llll}
1: & E & \to & E+B & 4: & B & \to & 0 \\
2: & E & \to & E*B & 5: & B & \to & 1 \\
3: & E & \to & B & & & &
\end{array}
$$

Exemple d'utilisation de la table

état	action					goto	
	0	1	*	+	$	E	B
0	d1	d2				4	3
1	r4	r4	r4	r4	r4		
2	r5	r5	r5	r5	r5		
3	r3	r3	r3	r3	r3		
4			d5	d6	acc		
5	d1	d2					7
6	d1	d2					8
7	r2	r2	r2	r2	r2		
8	r1	r1	r1	r1	r1		

On lit le mot :

1 + 0$

$$\begin{array}{|c|}\hline 4 \\ \hline 0 \\ \hline \perp \\ \hline \end{array}$$

$$\begin{array}{llll@{\qquad}llll}
1: & E & \to & E+B & 4: & B & \to & 0 \\
2: & E & \to & E*B & 5: & B & \to & 1 \\
3: & E & \to & B & & & &
\end{array}$$

Exemple d'utilisation de la table

état	action					goto	
	0	1	*	+	$	E	B
0	d1	d2				4	3
1	r4	r4	r4	r4	r4		
2	r5	r5	r5	r5	r5		
3	r3	r3	r3	r3	r3		
4			d5	d6	acc		
5	d1	d2					7
6	d1	d2					8
7	r2	r2	r2	r2	r2		
8	r1	r1	r1	r1	r1		

On lit le mot :

$1 + 0\$$

6
4
0
$\perp$

$$
\begin{array}{llll@{\qquad}llll}
1: & E & \rightarrow & E + B & 4: & B & \rightarrow & 0 \\
2: & E & \rightarrow & E * B & 5: & B & \rightarrow & 1 \\
3: & E & \rightarrow & B & & & &
\end{array}
$$

Exemple d'utilisation de la table

état	action					goto	
	0	1	*	+	$	E	B
0	d1	d2				4	3
1	r4	r4	r4	r4	r4		
2	r5	r5	r5	r5	r5		
3	r3	r3	r3	r3	r3		
4			d5	d6	acc		
5	d1	d2					7
6	d1	d2					8
7	r2	r2	r2	r2	r2		
8	r1	r1	r1	r1	r1		

On lit le mot :

$$1 + 0\$$$

1
6
4
0
$\perp$

$$\begin{array}{llll}
1: & E & \rightarrow & E + B \\
2: & E & \rightarrow & E * B \\
3: & E & \rightarrow & B
\end{array}
\qquad
\begin{array}{llll}
4: & B & \rightarrow & 0 \\
5: & B & \rightarrow & 1
\end{array}$$

Exemple d'utilisation de la table

état	action					goto	
	0	1	*	+	$	E	B
0	d1	d2				4	3
1	r4	r4	r4	r4	r4		
2	r5	r5	r5	r5	r5		
3	r3	r3	r3	r3	r3		
4			d5	d6	acc		
5	d1	d2					7
6	d1	d2					8
7	r2	r2	r2	r2	r2		
8	r1	r1	r1	r1	r1		

On lit le mot :

$$1 + 0\$$$

6
4
0
$\perp$

$$
\begin{array}{llll@{\qquad}llll}
1: & E & \to & E + B & 4: & B & \to & 0 \\
2: & E & \to & E * B & 5: & B & \to & 1 \\
3: & E & \to & B & & & &
\end{array}
$$

Exemple d'utilisation de la table

état	action					goto	
	0	1	*	+	$	E	B
0	d1	d2				4	3
1	r4	r4	r4	r4	r4		
2	r5	r5	r5	r5	r5		
3	r3	r3	r3	r3	r3		
4			d5	d6	acc		
5	d1	d2					7
6	d1	d2					8
7	r2	r2	r2	r2	r2		
8	r1	r1	r1	r1	r1		

On lit le mot :

$$1 + 0\$$$

8
6
4
0
$\perp$

$$\begin{array}{llll@{\qquad}llll}
1: & E & \rightarrow & E+B & 4: & B & \rightarrow & 0 \\
2: & E & \rightarrow & E*B & 5: & B & \rightarrow & 1 \\
3: & E & \rightarrow & B & & & &
\end{array}$$

Exemple d'utilisation de la table

état	action					goto	
	0	1	*	+	$	E	B
0	d1	d2				4	3
1	r4	r4	r4	r4	r4		
2	r5	r5	r5	r5	r5		
3	r3	r3	r3	r3	r3		
4			d5	d6	acc		
5	d1	d2					7
6	d1	d2					8
7	r2	r2	r2	r2	r2		
8	r1	r1	r1	r1	r1		

On lit le mot :

$$1 + 0\$$$

$$\begin{array}{|c|}\hline 0 \\ \hline \perp \\ \hline \end{array}$$

$$\begin{array}{llll@{\qquad}llll}
1: & E & \to & E + B & 4: & B & \to & 0 \\
2: & E & \to & E * B & 5: & B & \to & 1 \\
3: & E & \to & B & & & &
\end{array}$$

Exemple d'utilisation de la table

état	action					goto	
	0	1	*	+	\$	E	B
0	d1	d2				4	3
1	r4	r4	r4	r4	r4		
2	r5	r5	r5	r5	r5		
3	r3	r3	r3	r3	r3		
4			d5	d6	acc		
5	d1	d2					7
6	d1	d2					8
7	r2	r2	r2	r2	r2		
8	r1	r1	r1	r1	r1		

On lit le mot :

$1 + 0\$$

$$\begin{array}{|c|}\hline 4 \\ \hline 0 \\ \hline \perp \\ \hline \end{array}$$

$$\begin{array}{llll}
1: & E & \rightarrow & E + B \\
2: & E & \rightarrow & E * B \\
3: & E & \rightarrow & B \\
4: & B & \rightarrow & 0 \\
5: & B & \rightarrow & 1
\end{array}$$

Automate $LR(0)$

- La table SLR est construite à partir d'un automate appelé automate $LR(0)$.
- Le langage reconnu par cet automate est l'ensemble des séquences de symboles qui peuvent apparaître sur la pile d'un analyseur par décalage réduction pour cette grammaire.
- L'automate est utilisé pour construire la table SLR.
- La construction consiste à associer à tout état de l'automate des actions (décalage et réduction) en s'autorisant à regarder le prochain symbole.

Construction de l'automate $LR(0)$

- Augmentation de la grammaire
- Construction des ensembles d'items (FERMETURE)
- Construction de la fonction de transition (ALLER_A)
- Construction des SUIVANT() pour la grammaire

Augmentation de la grammaire

Soit la grammaire suivante :

$$\begin{array}{lcl} E & \rightarrow & E * B \mid E + B \mid B \\ B & \rightarrow & 0 \mid 1 \end{array}$$

Augmentation de la grammaire

Soit la grammaire suivante :

$$\begin{array}{lcl} E & \rightarrow & E * B \mid E + B \mid B \\ B & \rightarrow & 0 \mid 1 \end{array}$$

On ajoute un non-terminal de départ $S \rightarrow E$. La grammaire augmentée est donc :

$$\begin{array}{lcl} S & \rightarrow & E \\ E & \rightarrow & E * B \mid E + B \mid B \\ B & \rightarrow & 0 \mid 1 \end{array}$$

Articles

- Un article d'une grammaire G est une règle de G avec un marqueur $\bullet$ à une certaine position de la partie droite.
- En partant de la règle $A \rightarrow XYZ$ on peut créer les quatre articles suivants :
$$\begin{array}{lll} A & \rightarrow & \bullet XYZ \\ A & \rightarrow & X \bullet YZ \\ A & \rightarrow & XY \bullet Z \\ A & \rightarrow & XYZ\bullet \end{array}$$
- A partir de la règle $A \rightarrow \varepsilon$ on ne peut créer que l'article suivant : $A \rightarrow \bullet$
- Un article indique quelle partie d'une règle a déjà été reconnue à un certain point de l'analyse syntaxique (la partie se trouvant à gauche du point) et ce qui reste à reconnaître (à droite du point).

La fonction FERMETURE

Si I est un ensemble d'articles d'une grammaire G, alors, FERMETURE(I) est l'ensembles des articles construits à partir de I grâce aux deux règles suivantes :

1. Ajouter tous les articles de I dans FERMETURE(I)
2. Si $A \rightarrow \alpha \bullet B\beta \in$ FERMETURE(I) et $B \rightarrow \gamma$ est une règle de G, alors ajouter l'article $\beta \rightarrow \bullet\gamma$ à FERMETURE(I) s'il ne s'y trouve pas déjà. Appliquer cette règle jusqu'à ce qu'aucun nouvel article ne puisse être ajouté à FERMETURE(I).

$$\begin{array}{rcl} \textsc{Fermeture}(S \rightarrow \bullet E) & = & \{S \rightarrow \bullet E, \\ & & E \rightarrow \bullet E * B, \\ & & E \rightarrow \bullet E + B, \\ & & E \rightarrow \bullet B, \\ & & B \rightarrow \bullet 0, \\ & & B \rightarrow \bullet 1\} \end{array}$$

La fonction FERMETURE permet de définir les états de l'automate $LR(0)$

La fonction GOTO(I, X)

- Si I est un ensemble d'articles et X est un symbole, alors GOTO(I, X) est la FERMETURE de l'ensemble de tous les articles $A \to \alpha X \bullet \beta$ tels que $A \to \alpha \bullet X\beta$ est dans I
- La fonction GOTO(I, X) est utilisée pour définir les transitions de l'automate $LR(0)$ d'une grammaire.
- Exemple : $I_4 = \{S \to E\bullet, E \to E \bullet *B, E \to E \bullet +B\}$

$$\begin{aligned} \text{GOTO}(I_4, *) &= \text{FERMETURE}\{E \to E * \bullet B\} \\ &= \{E \to E * \bullet B, B \to \bullet 0, B \to \bullet 1\} \end{aligned}$$

Automate $LR(0)$

$$\boxed{\begin{array}{lll} S & \rightarrow & \bullet E \\ \hline & & \\ & & \\ & & \end{array}}$$

Automate $LR(0)$

$$
\boxed{\begin{array}{lcl}
\boldsymbol{S} & \rightarrow & \bullet\boldsymbol{E} \\
\hline
E & \rightarrow & \bullet E * B \\
E & \rightarrow & \bullet E + B \\
E & \rightarrow & \bullet B \\
 & & \\
 & &
\end{array}}
$$

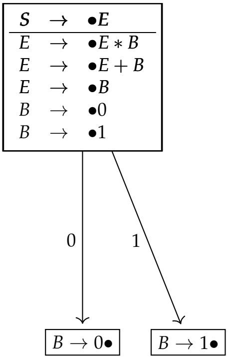
Automate $LR(0)$

$$\boxed{\begin{array}{ccl} S & \to & \bullet E \\ \hline E & \to & \bullet E * B \\ E & \to & \bullet E + B \\ E & \to & \bullet B \\ B & \to & \bullet 0 \\ B & \to & \bullet 1 \end{array}}$$

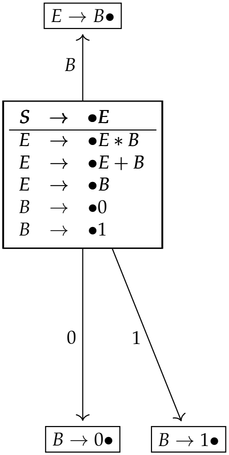
Automate $LR(0)$

$$\begin{tikzcd}
\boxed{\begin{array}{lcl} \boldsymbol{S} & \boldsymbol{\to} & \boldsymbol{\bullet E} \\ \hline E & \to & \bullet E * B \\ E & \to & \bullet E + B \\ E & \to & \bullet B \\ B & \to & \bullet 0 \\ B & \to & \bullet 1 \end{array}} \arrow[d, "0"'] \\
\boxed{B \to 0\bullet}
\end{tikzcd}$$

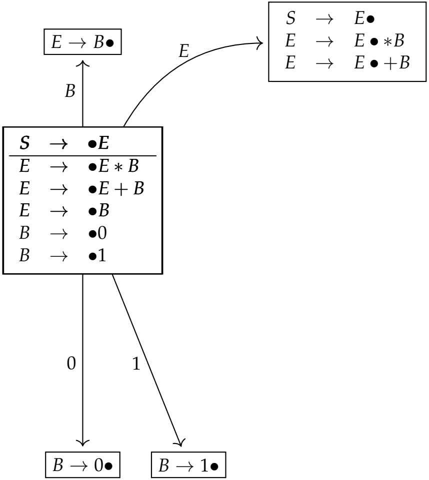
Automate $LR(0)$



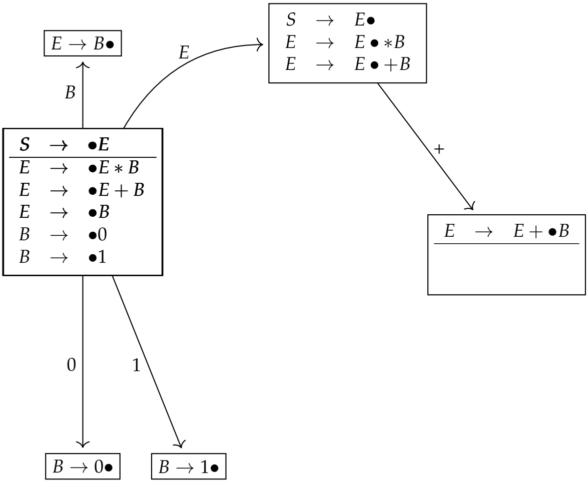
Automate $LR(0)$



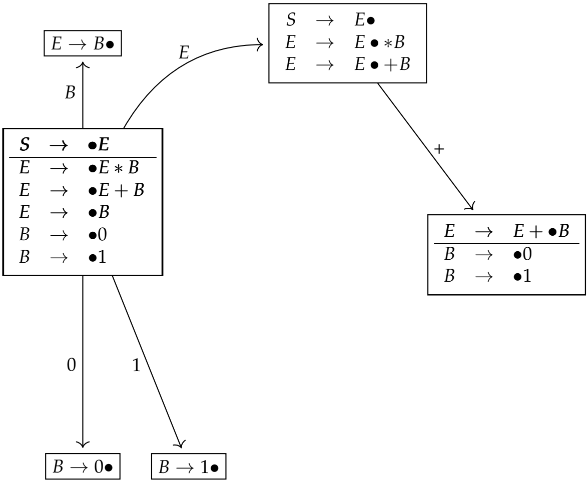
Automate $LR(0)$



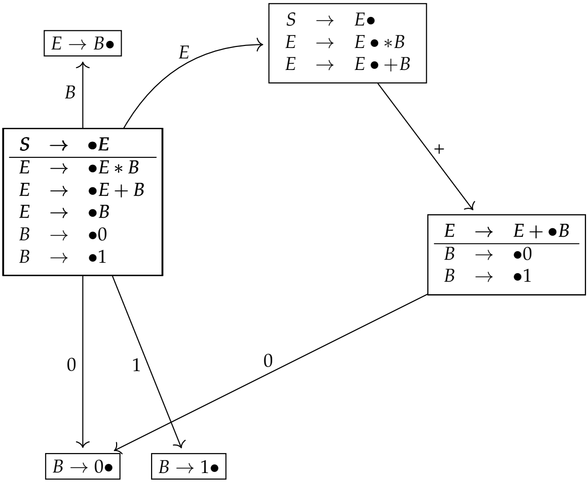
Automate $LR(0)$



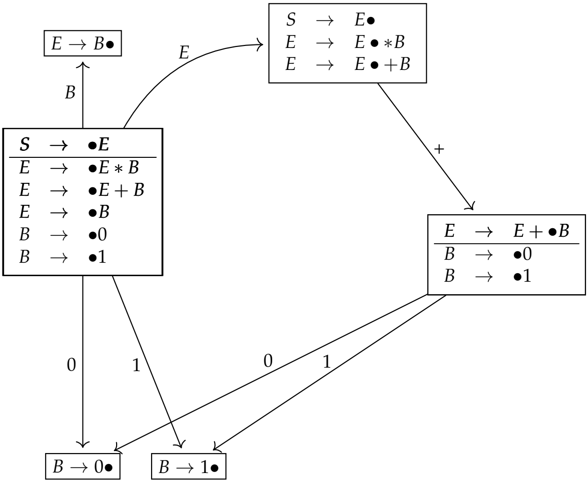
Automate $LR(0)$



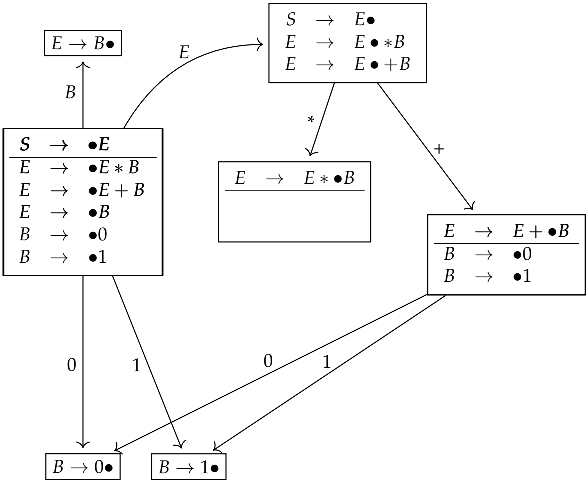
Automate $LR(0)$



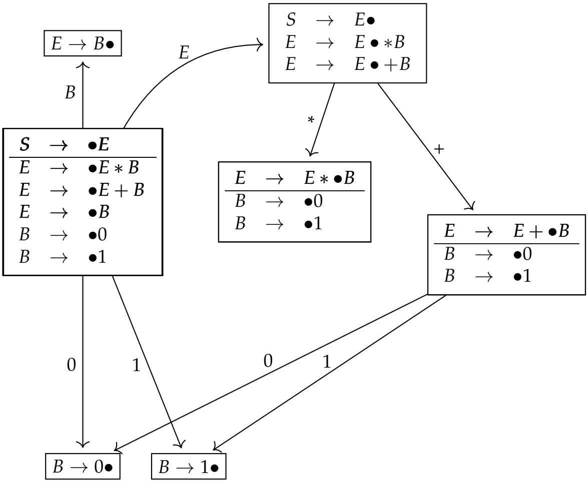
Automate $LR(0)$



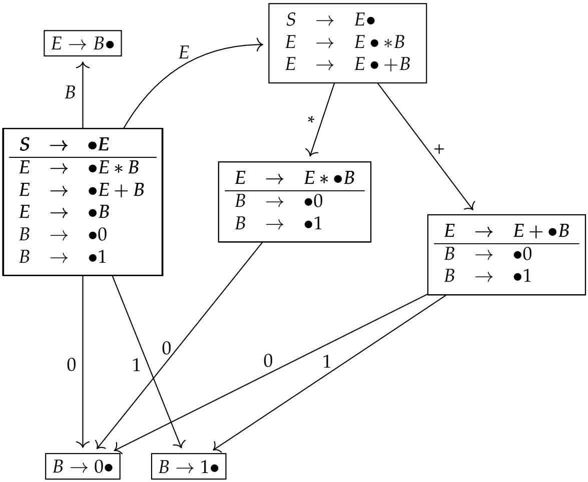
Automate $LR(0)$



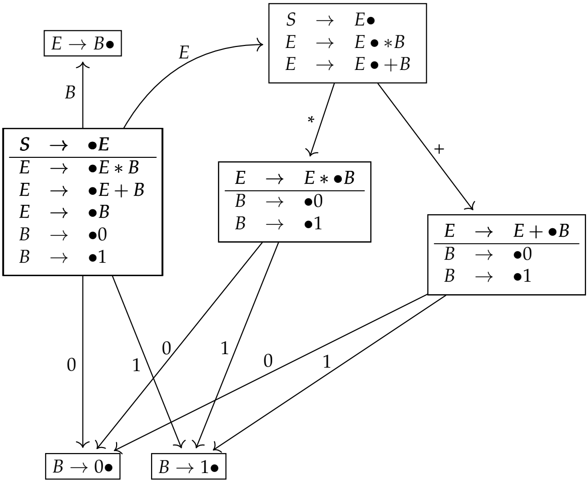
Automate $LR(0)$



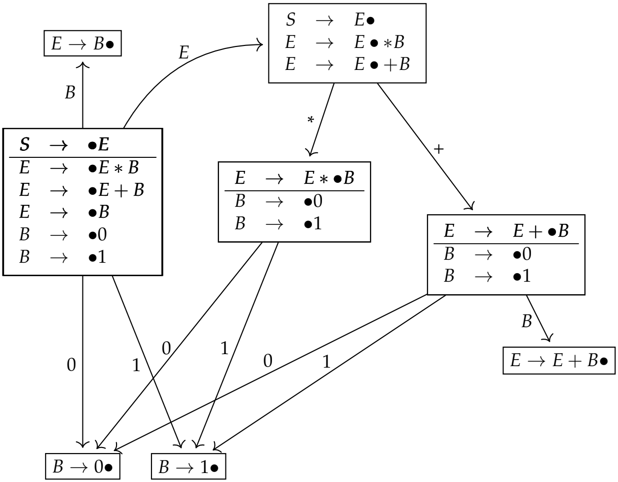
Automate $LR(0)$



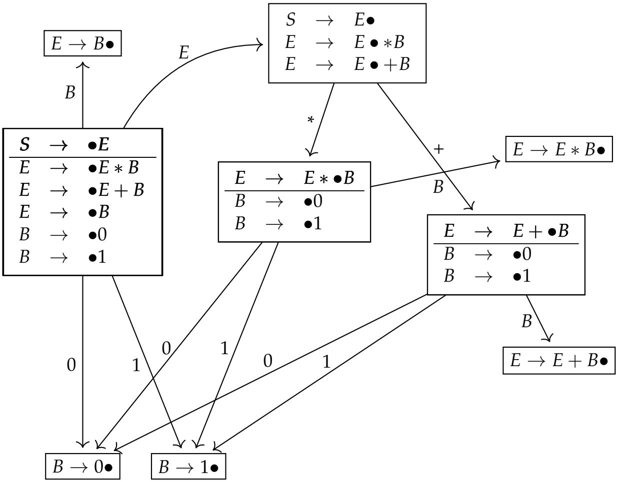
Automate $LR(0)$



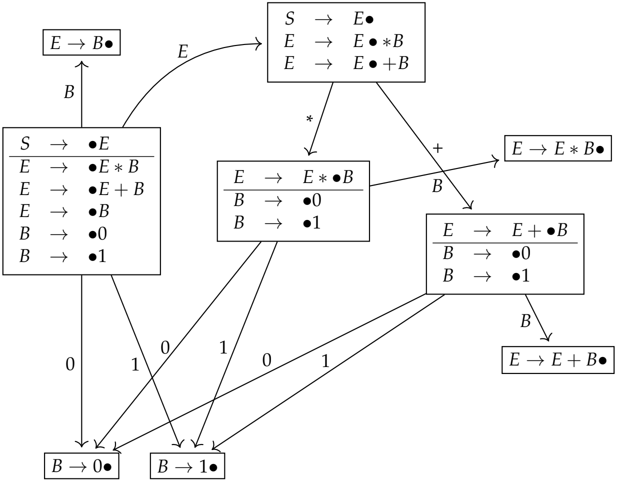
Automate $LR(0)$



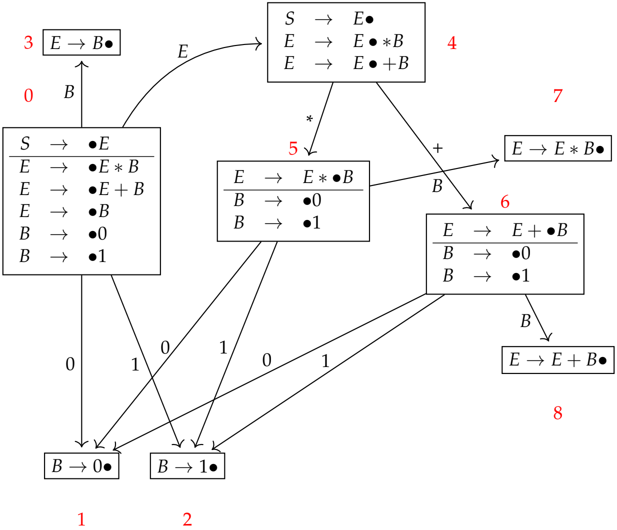
Automate $LR(0)$



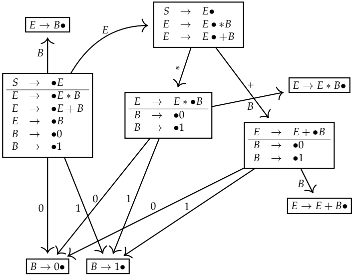
Automate $LR(0)$



Automate $LR(0)$

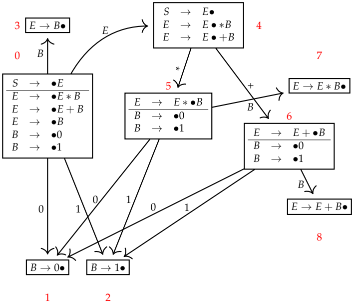


De l'automate $LR(0)$ à la table SLR



De l'automate $LR(0)$ à la table SLR

d = *décalage*, r = *réduction*, acc = *acceptation*



état	action					goto	
	0	1	*	+	$	E	B
0	d1	d2				4	3
1	r4	r4	r4	r4	r4		
2	r5	r5	r5	r5	r5		
3	r3	r3	r3	r3	r3		
4			d5	d6	acc		
5	d1	d2					7
6	d1	d2					8
7	r2	r2	r2	r2	r2		
8	r1	r1	r1	r1	r1		

$$
\begin{array}{llllllll}
1: & E & \to & E + B & 4: & B & \to & 0 \\
2: & E & \to & E * B & 5: & B & \to & 1 \\
3: & E & \to & B & & & &
\end{array}
$$

Construction de la table SLR

Entrée : Une grammaire augmentée G'

Construction de la table SLR

Entrée : Une grammaire augmentée G'

1. Construire l'automate $LR(0)$, ses états correspondent à la collection d'ensemble d'articles $\{I_0, I_1, \ldots, I_n\}$.

Construction de la table SLR

Entrée : Une grammaire augmentée G'

1. Construire l'automate $LR(0)$, ses états correspondent à la collection d'ensemble d'articles $\{I_0, I_1, \ldots, I_n\}$.
2. L'état i correspond à l'ensemble I_i. Les actions d'analyse syntaxique pour l'état i sont déterminées comme suit :

Construction de la table SLR

Entrée : Une grammaire augmentée G'

1. Construire l'automate $LR(0)$, ses états correspondent à la collection d'ensemble d'articles $\{I_0, I_1, \ldots, I_n\}$.
2. L'état i correspond à l'ensemble I_i. Les actions d'analyse syntaxique pour l'état i sont déterminées comme suit :
   1. Si $(A \to \alpha \bullet a\beta \in I_i)$ et $(a \in \Sigma)$ et $(\textsc{goto}(I_i, a) = I_j)$ alors $\textsc{action}[i, a] = dj$.

Construction de la table SLR

Entrée : Une grammaire augmentée G'

1. Construire l'automate $LR(0)$, ses états correspondent à la collection d'ensemble d'articles $\{I_0, I_1, \ldots, I_n\}$.
2. L'état i correspond à l'ensemble I_i. Les actions d'analyse syntaxique pour l'état i sont déterminées comme suit :
   1. Si $(A \to \alpha \bullet a\beta \in I_i)$ et $(a \in \Sigma)$ et $(\text{GOTO}(I_i, a) = I_j)$ alors $\text{ACTION}[i, a] = dj$.
   2. Si $(A \to \alpha \bullet \in I_i)$ et $(A \neq S')$ et $(a \in \text{SUIVANT}(A))$ alors $\text{ACTION}[i, a] = rj$ où j est le numéro de la règle $A \to \alpha$

Construction de la table SLR

Entrée : Une grammaire augmentée G'

1. Construire l'automate $LR(0)$, ses états correspondent à la collection d'ensemble d'articles $\{I_0, I_1, \ldots, I_n\}$.
2. L'état i correspond à l'ensemble I_i. Les actions d'analyse syntaxique pour l'état i sont déterminées comme suit :
   1. Si $(A \to \alpha \bullet a\beta \in I_i)$ et $(a \in \Sigma)$ et $(\text{GOTO}(I_i, a) = I_j)$ alors $\text{ACTION}[i, a] = dj$.
   2. Si $(A \to \alpha\bullet \in I_i)$ et $(A \neq S')$ et $(a \in \text{SUIVANT}(A))$ alors $\text{ACTION}[i, a] = rj$ où j est le numéro de la règle $A \to \alpha$
   3. Si $S' \to S\bullet \in I_i$, alors $\text{ACTION}[i, \$] = acc$

Construction de la table SLR

Entrée : Une grammaire augmentée G'

1. Construire l'automate $LR(0)$, ses états correspondent à la collection d'ensemble d'articles $\{I_0, I_1, \ldots, I_n\}$.
2. L'état i correspond à l'ensemble I_i. Les actions d'analyse syntaxique pour l'état i sont déterminées comme suit :
    1. Si $(A \rightarrow \alpha \bullet a\beta \in I_i)$ et $(a \in \Sigma)$ et $(\text{GOTO}(I_i, a) = I_j)$ alors $\text{ACTION}[i, a] = dj$.
    2. Si $(A \rightarrow \alpha \bullet \in I_i)$ et $(A \neq S')$ et $(a \in \text{SUIVANT}(A))$ alors $\text{ACTION}[i, a] = rj$ où j est le numéro de la règle $A \rightarrow \alpha$
    3. Si $S' \rightarrow S\bullet \in I_i$, alors $\text{ACTION}[i, \$] = acc$
3. Les transitions de transfert pour l'état i sont construites pour tout non terminal A à l'aide de la règle suivante : si $\text{GOTO}(I_i, A) = I_j$ alors $\text{GOTO}[i, A] = j$

Construction de la table SLR

Entrée : Une grammaire augmentée G'

1. Construire l'automate $LR(0)$, ses états correspondent à la collection d'ensemble d'articles $\{I_0, I_1, \ldots, I_n\}$.
2. L'état i correspond à l'ensemble I_i. Les actions d'analyse syntaxique pour l'état i sont déterminées comme suit :
    1. Si $(A \to \alpha \bullet a\beta \in I_i)$ et $(a \in \Sigma)$ et $(\text{GOTO}(I_i, a) = I_j)$ alors $\text{ACTION}[i, a] = dj$.
    2. Si $(A \to \alpha \bullet \in I_i)$ et $(A \neq S')$ et $(a \in \text{SUIVANT}(A))$ alors $\text{ACTION}[i, a] = rj$ où j est le numéro de la règle $A \to \alpha$
    3. Si $S' \to S\bullet \in I_i$, alors $\text{ACTION}[i, \$] = acc$
3. Les transitions de transfert pour l'état i sont construites pour tout non terminal A à l'aide de la règle suivante : si $\text{GOTO}(I_i, A) = I_j$ alors $\text{GOTO}[i, A] = j$
4. Toutes les entrées non remplies sont positionnées à *err*

Construction de la table SLR

Entrée : Une grammaire augmentée G'

1. Construire l'automate $LR(0)$, ses états correspondent à la collection d'ensemble d'articles $\{I_0, I_1, \ldots, I_n\}$.
2. L'état i correspond à l'ensemble I_i. Les actions d'analyse syntaxique pour l'état i sont déterminées comme suit :
   1. Si $(A \rightarrow \alpha \bullet a\beta \in I_i)$ et $(a \in \Sigma)$ et $(\text{GOTO}(I_i, a) = I_j)$ alors $\text{ACTION}[i, a] = dj$.
   2. Si $(A \rightarrow \alpha \bullet \in I_i)$ et $(A \neq S')$ et $(a \in \text{SUIVANT}(A))$ alors $\text{ACTION}[i, a] = rj$ où j est le numéro de la règle $A \rightarrow \alpha$
   3. Si $S' \rightarrow S \bullet \in I_i$, alors $\text{ACTION}[i, \$] = acc$
3. Les transitions de transfert pour l'état i sont construites pour tout non terminal A à l'aide de la règle suivante : si $\text{GOTO}(I_i, A) = I_j$ alors $\text{GOTO}[i, A] = j$
4. Toutes les entrées non remplies sont positionnées à *err*
5. L'état initial de l'analyseur est celui construit à partir de l'ensemble d'articles contenant $S' \rightarrow \bullet S$

SUIVANT(X)

- Permet de savoir quels symboles terminaux peuvent suivre le symbole X dans les proto-mots de la grammaire :

$$\textsc{suivant}(X) = \{a \in \Sigma \mid S \overset{*}{\Rightarrow} \alpha X a \beta\}$$

SUIVANT(X)

- Permet de savoir quels symboles terminaux peuvent suivre le symbole X dans les proto-mots de la grammaire :

$$\text{SUIVANT}(X) = \{a \in \Sigma \mid S \overset{*}{\Rightarrow} \alpha X a \beta\}$$

- Exemple 1 : soit la règle $X \to Ya$, elle nous dit qu'un Y peut être directement suivi par un a.

SUIVANT(X)

- Permet de savoir quels symboles terminaux peuvent suivre le symbole X dans les proto-mots de la grammaire :

$$\textsc{suivant}(X) = \{a \in \Sigma \mid S \overset{*}{\Rightarrow} \alpha X a \beta\}$$

- Exemple 1 : soit la règle $X \rightarrow Ya$, elle nous dit qu'un Y peut être directement suivi par un a.
- Exemple 2 : soit la règle $X \rightarrow YZ$, elle ne nous dit pas ce qui peut suivre directement un Y, il faut pour cela savoir quels terminaux Z permet de générer.
- Ces symboles sont déterminés par la fonction PREMIER(Z).

PREMIER

Si α est un proto-mot de G, $\textsc{premier}(\alpha)$ est l'ensemble des terminaux qui commencent les chaînes se dérivant de α :

$$\textsc{premier}(\alpha) = \{a \in \Sigma \mid \alpha \stackrel{*}{\Rightarrow} au\}$$

Si $\alpha \stackrel{*}{\Rightarrow} \varepsilon$ alors ε appartient aussi à $\textsc{premier}(\alpha)$.

Exemple

$$
\begin{aligned}
A &\rightarrow BC|a \\
B &\rightarrow b|\varepsilon \\
C &\rightarrow c|\varepsilon
\end{aligned}
$$

$$
\begin{array}{l|l}
A \Rightarrow a & a \in \text{PREMIER}(A) \\
A \Rightarrow BC \Rightarrow bC & b \in \text{PREMIER}(A) \\
A \Rightarrow BC \Rightarrow C \Rightarrow c & c \in \text{PREMIER}(A) \\
A \Rightarrow BC \Rightarrow C \Rightarrow \varepsilon & \varepsilon \in \text{PREMIER}(A)
\end{array}
$$

Exemple

$$A \rightarrow BC|a$$
$$B \rightarrow b|\varepsilon$$
$$C \rightarrow c|\varepsilon$$

$$\begin{array}{l|l} A \Rightarrow a & a \in \textsc{premier}(A) \\ A \Rightarrow BC \Rightarrow bC & b \in \textsc{premier}(A) \\ A \Rightarrow BC \Rightarrow C \Rightarrow c & c \in \textsc{premier}(A) \\ A \Rightarrow BC \Rightarrow C \Rightarrow \varepsilon & \varepsilon \in \textsc{premier}(A) \end{array}$$

plus généralement :
$\textsc{premier}(B) \subseteq \textsc{premier}(A)$
si $\varepsilon \in \textsc{premier}(B)$ alors $\textsc{premier}(C) \subseteq \textsc{premier}(A)$
si $\varepsilon \in \textsc{premier}(B)$ et $\varepsilon \in \textsc{premier}(C)$ alors $\varepsilon \in \textsc{premier}(A)$

Calcul de PREMIER(X)

Pour calculer PREMIER(X) avec $X \in N \cup \Sigma$, on applique les règles suivantes jusqu'à ce qu'aucun terminal ni ε ne puisse être ajouté aux ensembles PREMIER.

Calcul de PREMIER(X)

Pour calculer PREMIER(X) avec $X \in N \cup \Sigma$, on applique les règles suivantes jusqu'à ce qu'aucun terminal ni ε ne puisse être ajouté aux ensembles PREMIER.

1. Si $X \in \Sigma$ (X terminal), PREMIER$(X) = \{X\}$.

Calcul de PREMIER(X)

Pour calculer PREMIER(X) avec $X \in N \cup \Sigma$, on applique les règles suivantes jusqu'à ce qu'aucun terminal ni ε ne puisse être ajouté aux ensembles PREMIER.

1. Si $X \in \Sigma$ (X terminal), PREMIER$(X) = \{X\}$.
2. Si $X \to \varepsilon \in$ productions de la grammaire, on ajoute ε à PREMIER(X).

Calcul de PREMIER(X)

Pour calculer PREMIER(X) avec $X \in N \cup \Sigma$, on applique les règles suivantes jusqu'à ce qu'aucun terminal ni ε ne puisse être ajouté aux ensembles PREMIER.

1. Si $X \in \Sigma$ (X terminal), PREMIER$(X) = \{X\}$.
2. Si $X \to \varepsilon \in$ productions de la grammaire, on ajoute ε à PREMIER(X).
3. Si $X \in N$ (X non terminal) et $X \to Y_1 \ldots Y_k \in P$, mettre a dans PREMIER(X) s'il existe i tel que a est dans PREMIER(Y_i) et que ε est dans tous les PREMIER(Y_1) … PREMIER(Y_{i-1}).
   Si $\varepsilon \in$ PREMIER$(Y_j) \forall j$, $1 \leq j \leq k$, on ajoute ε à PREMIER(X).

PREMIER$(X_1 \ldots X_n)$

On calcule PREMIER$(X_1 \ldots X_n)$ de la façon suivante :

1. Ajouter à PREMIER$(X_1 \ldots X_n)$ tous les symboles de PREMIER(X_1) différents de ε.
2. Si $\varepsilon \in$ PREMIER(X_1), ajouter également les symboles de PREMIER(X_2) différents de ε.
   Si $\varepsilon \in$ PREMIER(X_2), ajouter également les symboles de PREMIER(X_3) différents de ε, etc.
3. Finalement, si ε appartient à PREMIER(X_j) pour tous les $j = 1, 2, \ldots n$, on ajoute ε à PREMIER$(X_1 \ldots X_n)$.

SUIVANT(X)

Si $X \in N$, $\textsc{suivant}(X)$ est l'ensemble des symboles $a \in \Sigma$ qui peuvent apparaître immédiatement à droite de X dans un proto-mot :

$$\textsc{suivant}(X) = \{a \in \Sigma \mid S \stackrel{*}{\Rightarrow} \alpha X a \beta\}$$

Si X peut être le symbole le plus à droite d'un proto-mot alors $\perp$ est dans $\textsc{suivant}(X)$.

Exemple

$$
\begin{aligned}
S &\rightarrow Aa \\
A &\rightarrow BC \\
C &\rightarrow c|\varepsilon
\end{aligned}
$$

$$
\begin{array}{l|l}
S \Rightarrow Aa \Rightarrow BCa \Rightarrow Bca & c \in \textsc{suivant}(B) \\
S \Rightarrow Aa \Rightarrow BCa \Rightarrow Ba & a \in \textsc{suivant}(B)
\end{array}
$$

Exemple

$S \rightarrow Aa$
$A \rightarrow BC$
$C \rightarrow c|\varepsilon$

$$\begin{array}{l|l} S \Rightarrow Aa \Rightarrow BCa \Rightarrow Bca & c \in \textsc{suivant}(B) \\ S \Rightarrow Aa \Rightarrow BCa \Rightarrow Ba & a \in \textsc{suivant}(B) \end{array}$$

plus généralement :
$\textsc{premier}(C) \subseteq \textsc{suivant}(B)$
si $\varepsilon \in \textsc{premier}(C)$ alors $\textsc{suivant}(A) \subseteq \textsc{suivant}(B)$

SUIVANT(X)

Pour calculer SUIVANT(X) pour tous symbole non terminal X, on applique les règles suivantes jusqu'à ce qu'aucun symbole terminal ne puisse être ajouté aux ensembles SUIVANT :

1. Mettre $\perp$ dans SUIVANT(S).
2. si $X \to \alpha B \beta$, le contenu de PREMIER(β), excepté ε, est ajouté à SUIVANT(B).
3. s'il existe une règle $X \to \alpha B$ ou une règle $X \to \alpha B \beta$ telle que $\varepsilon \in$ PREMIER(β) (c'est à dire $\beta \overset{*}{\Rightarrow} \varepsilon$), les éléments de SUIVANT(X) sont ajoutés à SUIVANT(B).

Exemple

Soit la grammaire $G = \langle \{E, E', T, T', F\}, \{a, +, *, (,), a\}, P, E\rangle$ non récursive à gauche où P est composé des règles suivantes :

$$\begin{array}{ll} 1\ E \to TE' & 2\ E' \to +TE' \\ 3\ E' \to \varepsilon & 4\ T \to FT' \\ 5\ T' \to *FT' & 6\ T' \to \varepsilon \\ 7\ F \to (E) & 8\ F \to a \end{array}$$

Alors :

$\textsc{premier}(E) = \textsc{premier}(T) = \textsc{premier}(F) = \{(, a\}$
$\textsc{premier}(E') = \{+, \varepsilon\}$
$\textsc{premier}(T') = \{*, \varepsilon\}$
$\textsc{suivant}(E) = \{), \bot\}$
$\textsc{suivant}(E') = \textsc{suivant}(E) = \{), \bot\}$
$\textsc{suivant}(T) = \{\textsc{premier}(E') - \{\varepsilon\}\} \cup \textsc{suivant}(E) = \{+,), \bot\}$
$\textsc{suivant}(T') = \textsc{suivant}(T) = \{+,), \bot\}$
$\textsc{suivant}(F) = \{\textsc{premier}(T') - \{\varepsilon\}\} \cup \textsc{suivant}(T) = \{+, *,), \bot\}$

Conflits

- Si, à l'issue de la construction de la table, une case possède plusieurs actions, alors la grammaire n'est pas $SLR(1)$.
- Si une case possède deux réductions différentes, on dit qu'il y a un conflit réduction/réduction.
- Si une case possède un décalage et une réduction, on qu'il y a conflit décalage/réduction.
- Dans ce cas, soit la grammaire est ambiguë, soit il faut augmenter le regard en avant (la valeur de k).
- La majorité des langages de programmation admettent une grammaire qui est $SLR(1)$. En particulier le langage L.

Conflit de réduction / réduction

$$\begin{array}{rccc} 1: & S & \to & X \\ 2: & S & \to & Y \\ 3: & X & \to & a \\ 4: & Y & \to & a \end{array}$$

- On ne sait pas s'il faut réduire par 3 ou 4.
- La grammaire est ambiguë !

Conflit de réduction / réduction (2)

$$\begin{array}{lcccc} 1: & S & \rightarrow & X & b \\ 2: & S & \rightarrow & Y & c \\ 3: & X & \rightarrow & a & \\ 4: & Y & \rightarrow & a & \end{array}$$

- Si on sait que a est suivi de b (ou de c), il n'y a pas conflit !
- La grammaire est $LR(1)$

Conflit de décalage / réduction

$$
\begin{array}{llll}
1: & S & \rightarrow & X \\
2: & S & \rightarrow & Y \quad b \\
3: & X & \rightarrow & a \quad b \\
4: & Y & \rightarrow & a
\end{array}
$$

- Après un décalage d'un a, on ne sait pas s'il faut décaler ou réduire.
- La grammaire est ambiguë.

Conflit de décalage / réduction

$$\begin{array}{lllll} 1: & S & \rightarrow & X & \\ 2: & S & \rightarrow & Y & b \\ 3: & X & \rightarrow & a & c \\ 4: & Y & \rightarrow & a & \end{array}$$

- Si on sait que a est suivi de b (ou de c), il n'y a pas conflit !
- La grammaire est $LR(1)$.

Grammaires $LR(2)$

$$
\begin{array}{lllll}
1: & S & \rightarrow & X & b & c \\
2: & S & \rightarrow & Y & b & d \\
3: & X & \rightarrow & a & & \\
4: & Y & \rightarrow & a & &
\end{array}
$$

Sources

- Michael Sipser,
  Introduction to the Theory of Computation.
  PWS Publishing Company, 1997.
- John Hopcroft, Rajeev Motwani, Jeffrey Ullman,
  Introduction to Automata Theory, Languages and Computation.
  2ème édition, Pearson Education International, 2001.
- John Aho, Jeffrey Ullman, *The Theory of Parsing, Translation and Compiling, Vol I : Parsing.*
  Prentice-Hall, 1972