TD5 - LAAS

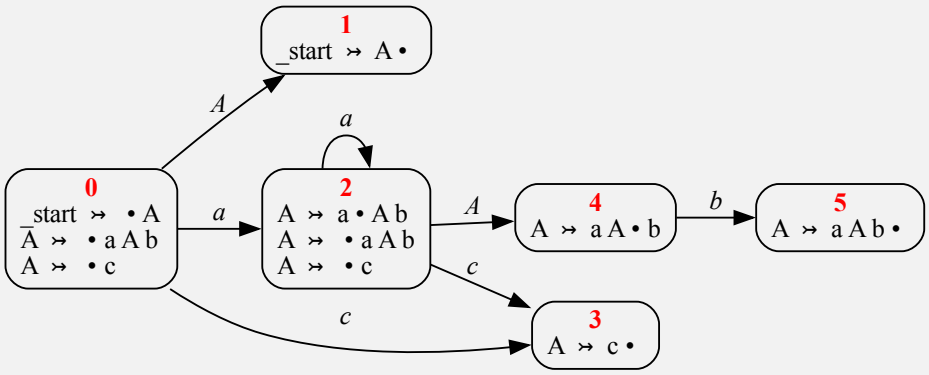
avril 2023

Exercice 1 :

Q 1 . Construisez l'automate LR(0) de la grammaire G_1 :

$$\underline{A} \rightarrow aAb \mid c$$

La grammaire est-elle LR(0) ?
(construisez l'automate LR(0) puis la table « Action »).



La grammaire est LR(0)

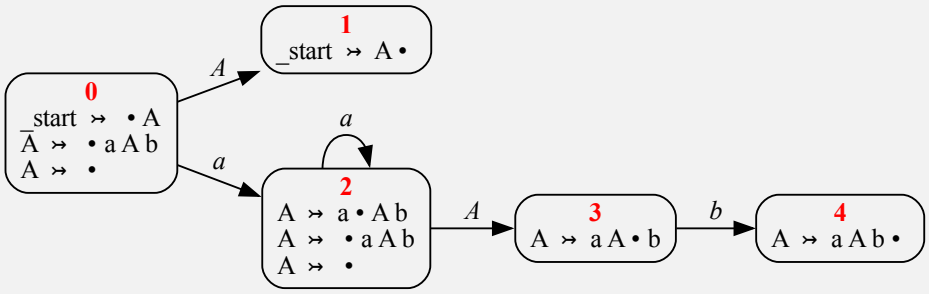
LR0	0	1	2	3	4	5
a	[S] 2		[S] 2	[R] $A \rightarrow c$		[R] $A \rightarrow a\ A\ b$
b				[R] $A \rightarrow c$	[S] 5	[R] $A \rightarrow a\ A\ b$
c	[S] 3		[S] 3	[R] $A \rightarrow c$		[R] $A \rightarrow a\ A\ b$
#		[A]		[R] $A \rightarrow c$		[R] $A \rightarrow a\ A\ b$
A	1		4			

On peut comparer à la table SLR(1), après avoir calculé que $Suiv(A) = \{\#, b\}$:

SLR1	0	1	2	3	4	5
a	[S] 2		[S] 2			
b				[R] $A \rightarrow c$	[S] 5	[R] $A \rightarrow a\ A\ b$
c	[S] 3		[S] 3			
#		[A]		[R] $A \rightarrow c$		[R] $A \rightarrow a\ A\ b$
A	1		4			

Q 2 . Mêmes questions pour G_2 :

$$\underline{A} \rightarrow aAb \mid \varepsilon$$



Confilts shift/reduce en 0 (pour Action[0,a]) ⇒ La grammaire n'est pas LR(0)
(remarque : même conflit pour Action[2,a])

Q 3 .

1. Calculez $Suiv(A)$ (grammaire G_2)
2. Pour construire la table action LR(0), quand un état i contient une règle $V \to w\bullet$ on ajoute une action Reduce pour **toutes** les lettres terminales et pour #.

   Nous allons procéder autrement en ajoutant une action Reduce **uniquement** pour les lettres de $Suiv(V)$.

   En quoi cela change-t-il la table Action de la question précédente ? NB : cette façon de procéder s'appelle l'analyse SLR(1).
3. Avec cette nouvelle table Action, retracez le déroulement de l'algorithme d'analyse ascendante du mot *aabb*.

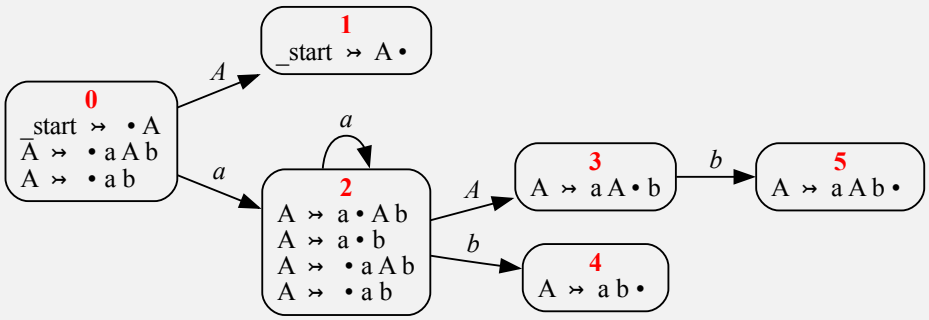
$Suiv(A) = \{b, \#\}$.
En appliquant le principe SLR(1) les conflits précédents disparaissent.

SLR1	0	1	2	3	4
a	[S] 2		[S] 2		
b	[R] $A \to \varepsilon$		[R] $A \to \varepsilon$	[S] 4	[R] $A \to a\ A\ b$
#	[R] $A \to \varepsilon$	[A]	[R] $A \to \varepsilon$		[R] $A \to a\ A\ b$
A	1		3		

état initial	⇝	0	aaba#
décalage	⇝	0 2	abb#
décalage	⇝	0 2 2	bb#
réduction	⇝	0 2 2 3	bb#
décalage	⇝	0 2 2 3 4	b#
réduction	⇝	0 2 3	b#
décalage	⇝	0 2 3 4	#
réduction	⇝	0 1	#
Mot accepté			

Q 4 . Nettoyez la grammaire G_2. On appellera G_3 la grammaire obtenue. Notez que le langage engendré par G_3 ne contient plus le mot vide.
G_3 est-elle LR(0) ?

$\underline{A} \rightarrow aAb \mid ab$



La grammaire est LR(0)

LR0	0	1	2	3	4	5
a	[S] 2		[S] 2	[R] $A \rightarrow c$		[R] $A \rightarrow a\ A\ b$
b				[R] $A \rightarrow c$	[S] 5	[R] $A \rightarrow a\ A\ b$
c	[S] 3		[S] 3	[R] $A \rightarrow c$		[R] $A \rightarrow a\ A\ b$
#		[A]		[R] $A \rightarrow c$		[R] $A \rightarrow a\ A\ b$
A	1		4			

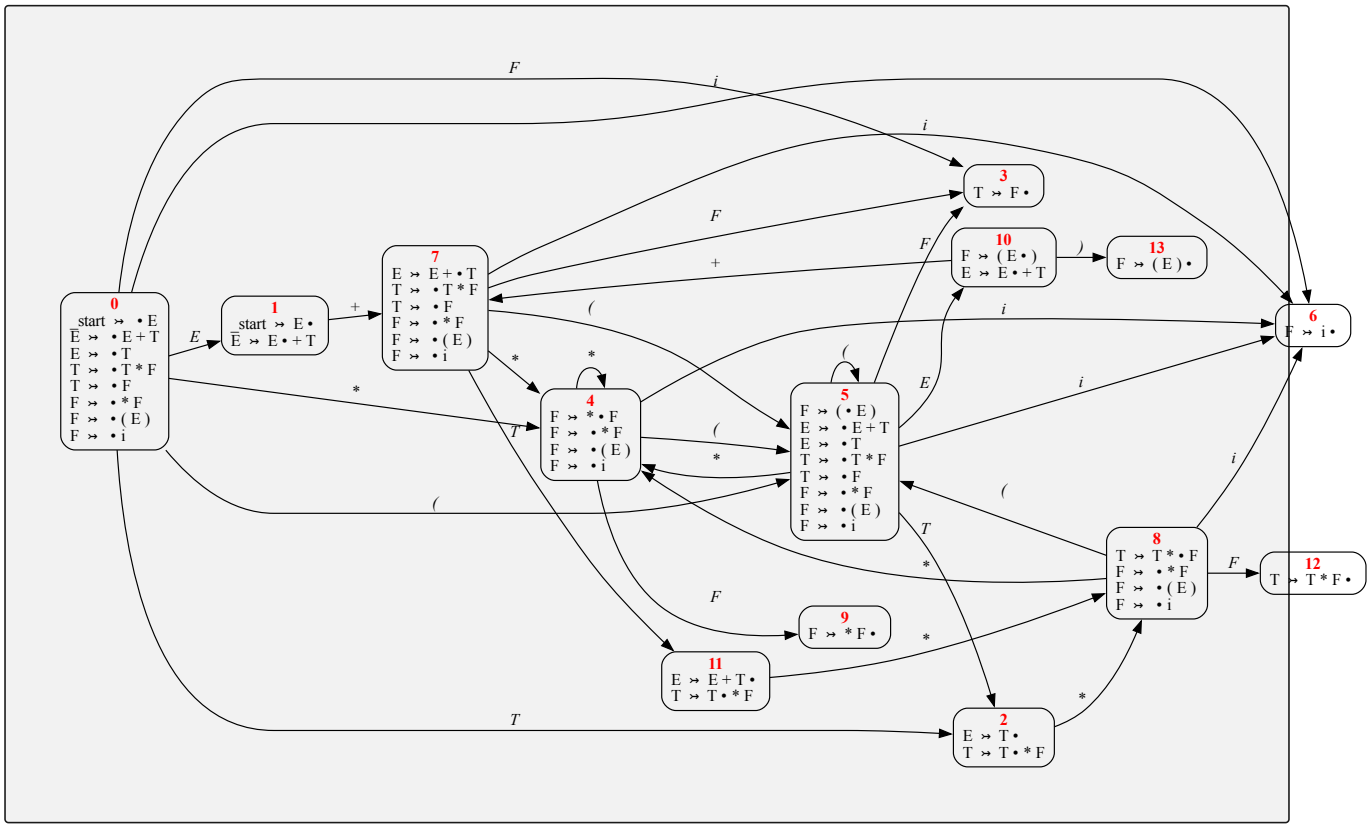
LR0	0	1	2	3	4	5
a	[S] 2		[S] 2		[R] $A \rightarrow a\ b$	[R] $A \rightarrow a\ A\ b$
b			[S] 4	[S] 5	[R] $A \rightarrow a\ b$	[R] $A \rightarrow a\ A\ b$
#		[A]			[R] $A \rightarrow a\ b$	[R] $A \rightarrow a\ A\ b$
A	1		3			

Exercice 2 :

On considère la grammaire G_1 suivante d'axiome E :

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

Q 1 . Construire l'automate $LR(0)$ correspondant.



Q 2 .

— Pourquoi cette grammaire n'est-elle pas LR(0) ?

— Est-elle SLR(1) ?

Indication : ne pas construire la table Action complète !

Non LR(0) car confilts shift/reduce dans l'état $2 = \delta(0, T)$ et 12 pour la lettre $*$
Les conflits disparaissent en SLR(1) car $Suiv(E)$ ne contient pas $*$
Suivants : `{'E': {')', '+', '#'}, 'T': {'#', '+', ')', '*'}, 'F': {'#', '+', ')', '*'}}`
La table complète est ... grande :

SLR1	0	1	2	3	4	5	6	7	8	9	10	11	12
(	[S] 5				[S] 5	[S] 5		[S] 5	[S] 5				
)			[R] E → T	[R] T → F			[R] F → i			[R] F → * F	[S] 13	[R] E → E + T	[R] T → T * F
*	[S] 4		[S] 8	[R] T → F	[S] 4	[S] 4	[R] F → i	[S] 4	[S] 4	[R] F → * F		[S] 8	[R] T → T * F
+		[S] 7	[R] E → T	[R] T → F			[R] F → i			[R] F → * F	[S] 7	[R] E → E + T	[R] T → T * F
i	[S] 6				[S] 6	[S] 6		[S] 6	[S] 6				
#		[A]	[R] E → T	[R] T → F			[R] F → i			[R] F → * F		[R] E → E + T	[R] T → T * F
T	2					2		11					
E	1					10							
F	3				9	3		3	12				

Q 3 . Cette grammaire est-elle LL(1) ?

Non car elle est récursive gauche.

Q 4 . Cette grammaire est-elle non ambiguë ?

Elle est SLR(1) donc n'est pas ambigüe.

On remplace la règle $F \longrightarrow *F$ de la grammaire G_1 par la règle $F \longrightarrow F*$, ce qui fournit une grammaire G_2.

Q 5 . Montrer que la grammaire G_2 n'est pas $SLR(1)$.
Indication : Calculer les ensembles « Suivant ». Calculer l'état 0 de l'automate LR(0), puis l'état $\delta(0, F)$.

Suivants : `{'E': {'+', ')', '#'}, 'T': {'+', ')', '#', '*'}, 'F': {'+', '#', '*', ')'}}`
Dans l'état $\delta(0, F) = 4$ on a un conflit shift/reduce pour la lettre `*` qui appartient à $Suiv(T)$
Inutile de calculer l'ensemble de l'automate.
(à titre d'info, le voici) :