



Analyses Ascendants LR(k)



8. Analyse ascendante déterministe



8.5 Analyseur LR(1) (K=1)

- *Un automate est NON LR(0) s'il existe un état (ensemble d'items) qui comporte des **conflits** :*
- ✓ *reduction/reduction.*
 - ✓ *Décalage/reduction*

*On peut résoudre les conflits **décalage/réduction** en regardant si le **terminal en tête du mot restant** à analyser est dans l'ensemble des **Suivants** du **non-terminal** en membre gauche de la règle utilisable pour la réduction.*

*- **Si c'est le cas**, on effectue la réduction, **sinon** on effectue un décalage.*

*Ceci est formalisé par la méthode **LR(1)***



Construction de l'ensemble des items LR(1)



8. Analyse ascendante déterministe



8.5 Analyseur LR(1)

- La *prévision* sera utile quand il est question de faire une **Réduction**,
- Cela revient à calculer les **Suivants locaux** aux règles (items)

Calcul des LR(1)-items

- ✓ On aura des **LR(1)-items** de la forme $[A \rightarrow \alpha_1 \bullet \alpha_2, u]$ où ***u*** est un suivant local à $A \rightarrow \alpha$ ($\alpha = \alpha_1 \alpha_2$)
- ✓ Le **LR(1)-item** “**initial**” est $[S' \rightarrow \bullet S, \$]$
- ✓ La méthode pour construire la collection des **ensembles d’items LR(1)** est **essentiellement la même** que celle de **LR(0)**. Il suffit uniquement de **modifier** les deux procédures **Fermeture** «closure» et transition «**GOTO**»

8. Analyse ascendante déterministe



8.5 Analyseur LR(1)

Fermeture (LR(1)) :

La fermeture de l'item **LR(1) $[A \rightarrow \alpha \cdot \beta, a]$** contient:

1. l'item **LR(1) $[A \rightarrow \alpha \cdot \beta, a]$** lui même
2. si $\beta = B\gamma$ où B est un non terminal et si $B \rightarrow \delta$ ajouter l'item $[B \rightarrow \cdot \delta, b]$ où b est un terminal de **Deb(γa)**.
3. Itérer le pas 2 jusqu'à ce qu'aucun items ne puisse être ajouté.

8. Analyse ascendante déterministe



8.5 Analyseur LR(1)

Transition GOTO LR(1) :

La fonction **GOTO**(I_i, X) existe si dans I_i , il existe un item de la forme : $[A \rightarrow \alpha \cdot X \beta , a]$ et dans ce cas elle est égale à la fermeture de l'item $[A \rightarrow \alpha X \cdot \beta , a]$



8. Analyse ascendante déterministe



Exemple (Ensemble d'items LR(1))

Soit la grammaire Augmentée
 $Z \rightarrow S$ (0)
 $S \rightarrow Sbc$ (1)
 $S \rightarrow a$ (2)

Calculer l'ensemble des items LR(1)?

Items LR(1):

$I_0 = \{ [Z \rightarrow .S, \#], [S \rightarrow .Sbc, \#], [S \rightarrow .a, \#], [S \rightarrow .Sbc, b], [S \rightarrow .a, b] \}$

$I_1 = Goto(I_0, S) = \{ [Z \rightarrow S., \#], [S \rightarrow S.bc, \#], [S \rightarrow S.bc, b] \}$

$I_2 = Goto(I_0, a) = \{ [S \rightarrow a., \#], [S \rightarrow a., b] \}$

$I_3 = Goto(I_1, b) = \{ [S \rightarrow Sb.c, \#], [S \rightarrow Sb.c, b] \}$

$I_4 = Goto(I_3, c) = \{ [S \rightarrow Sbc., \#], [S \rightarrow Sbc., b] \}$



Construction de la table d'analyse LR(1)



8. Analyse ascendante déterministe



Table d'Analyse LR(1)

1. Construire $C = \{I_0, I_1, \dots, I_n\}$, la collection des ensembles d'items LR(1) qui représente les n états de l'analyseur
2. Les **actions d'analyse** pour l'état i sont déterminées de la façon suivante :
 - a. Si $[A \rightarrow \alpha \bullet a\beta, b]$ est dans I_i , remplir **Action** $[i, a]$ avec **D_j** où j est tel que $GOTO(I_i, a) = I_j$
 - b. Si $[A \rightarrow \alpha \bullet, a]$ est dans I_i , $A \neq S'$ remplir **Action** $[i, a]$ avec **R_j** où j est le num de la règle $A \rightarrow \alpha$ dans la grammaire.
 - c. Si $[S' \rightarrow S \bullet, \#]$ est dans I_i remplir **action** $[i, \#]$ avec **accepter**
3. les transitions **GOTO** pour l'état i sont déterminées comme suit :

si $GOTO(I_i, A) = I_j$ alors placer dans la **case** (i, A) le **numéro** j .
4. toutes les entrées non définies par les règles 2 et 3 sont remplies avec erreur.

8. Analyse ascendante déterministe



Exemple (Table d'analyse LR(1))

Soit la grammaire Augmentée

$Z \rightarrow S$ (0)
 $S \rightarrow Sbc$ (1)
 $S \rightarrow a$ (2)

Items LR(1):

$I_0 = \{ [Z \rightarrow .S, \#], [S \rightarrow .Sbc, \#], [S \rightarrow .a, \#], [s \rightarrow .Sbc, b], [S \rightarrow .a, b] \}$
 $I_1 = Goto(I_0, S) = \{ [Z \rightarrow S. , \#], [S \rightarrow S.bc, \#], [S \rightarrow S.bc, b] \}$
 $I_2 = Goto(I_0, a) = \{ [S \rightarrow a. , \#], [S \rightarrow a. , b] \}$
 $I_3 = Goto(I_1, b) = \{ [S \rightarrow Sb.c, \#], [S \rightarrow Sb.c, b] \}$
 $I_4 = Goto(I_3, c) = \{ [S \rightarrow Sbc. , \#], [S \rightarrow Sbc. , b] \}$

Table d'analyseLR(1):

	a	b	c	#	S
I₀	D, I ₂				I ₁
I₁		D, I ₃		Acc	
I₂		R ₂		R ₂	
I₃			D, I ₄		
I₄		R ₁		R ₁	

La table est mono-définie, donc la grammaire est LR(1).

8. Analyse ascendante déterministe

Remarque



PresenterMedia

Une grammaire réursive gauche directe en S n'est pas **LR(0)**.



PresenterMedia

Une grammaire est **LR(k)** alors elle est **LR(k+1)**.



PresenterMedia

Une grammaire est **LR(k)** Ssi la *table d'analyse* **LR(k)** est **monodéfinie**.

8. Analyse ascendante déterministe



Critique de LR(1) (LR(k)):

*Les analyseurs LR sont construits pour reconnaître tous les langages de programmation, ils sont très **performants**, mais leurs **tables d'analyse** contiennent des milliers d'états, de ce fait elles sont **couteuses** à implémenter. Pour cela, on a pensé aux **SLR et LALR** qui permettent d'optimiser les grammaires LR.*



Analysers **SLR** (Simple LR) & **LALR** (LookAhead LR)





Motivations

8. Analyse ascendante déterministe

- La classe **LR(1)** est largement suffisante pour les langages modernes, et la table n'a qu'une colonne par token.
- Mais le **nombre d'états** qui **grandit trop**, en raisons de la présence de **look-ahead** dans les états qui départage des états qui sont **très peu différents**.
- C'est pour cela que dans la pratique on utilise **deux types d'analyseurs** dont la **puissance** est comprise entre celle de **LR(0)** et celle de **LR(1)**: **SLR** et **LALR(1)**



8. Analyse ascendante déterministe

Analyseurs SLR

SLR (Simple LR) est un analyseur dont

- ✓ l'automate est celui de **LR(0)**,
- ✓ donc la partie **GOTO** est la même que LR(0),
- ✓ et les actions de décalage sont aussi les mêmes que LR(0)
- ✓ mais la **table d'analyse** (basée sur les items LR(0)) est construite de façon plus fine: on pallie à l'absence de look-ahead dans les états avec l'information contenue dans les **ensembles suivants** construits à partir de la grammaire.



8. Analyse ascendante déterministe

Analyseurs SLR

SLR (Simple LR) est un analyseur dont

- ✓ l'automate est celui de **LR(0)**,
- ✓ donc la partie **GOTO** est la même que LR(0),
- ✓ et les actions de décalage sont aussi les mêmes que LR(0)
- ✓ mais la **table d'analyse** (basée sur les items LR(0)) est construite de façon plus fine: on pallie à l'absence de look-ahead dans les états avec l'information contenue dans les **ensembles suivants** construits à partir de la grammaire.



8. Analyse ascendante déterministe

Analyseurs SLR

SLR (Simple LR) est un analyseur dont

- ✓ l'automate est celui de LR(0),
- ✓ donc la partie GOTO est la même que LR(0),
- ✓ et les actions de décalage sont aussi les mêmes que LR(0)
- ✓ mais la table d'analyse (basée sur les items LR(0)) est construite de façon plus fine: on pallie à l'absence de look-ahead dans les états avec l'information contenue dans les ensembles suivants construits à partir de la grammaire.

La règle de placement des Réductions n'est plus de mettre Réduire (R_i) sur toute la ligne *i* et devient alors: si l'état *j* contient un ITEM LR(0) $X \rightarrow \gamma \cdot$, et que $X \rightarrow \gamma$ est la production numéro *j* ($X \leftrightarrow$ Axiome), on met R_k (Réduire par la règle num *K*) dans toutes les cases (*j*, *t*) telles que $t \in \text{Suivant}(X)$.



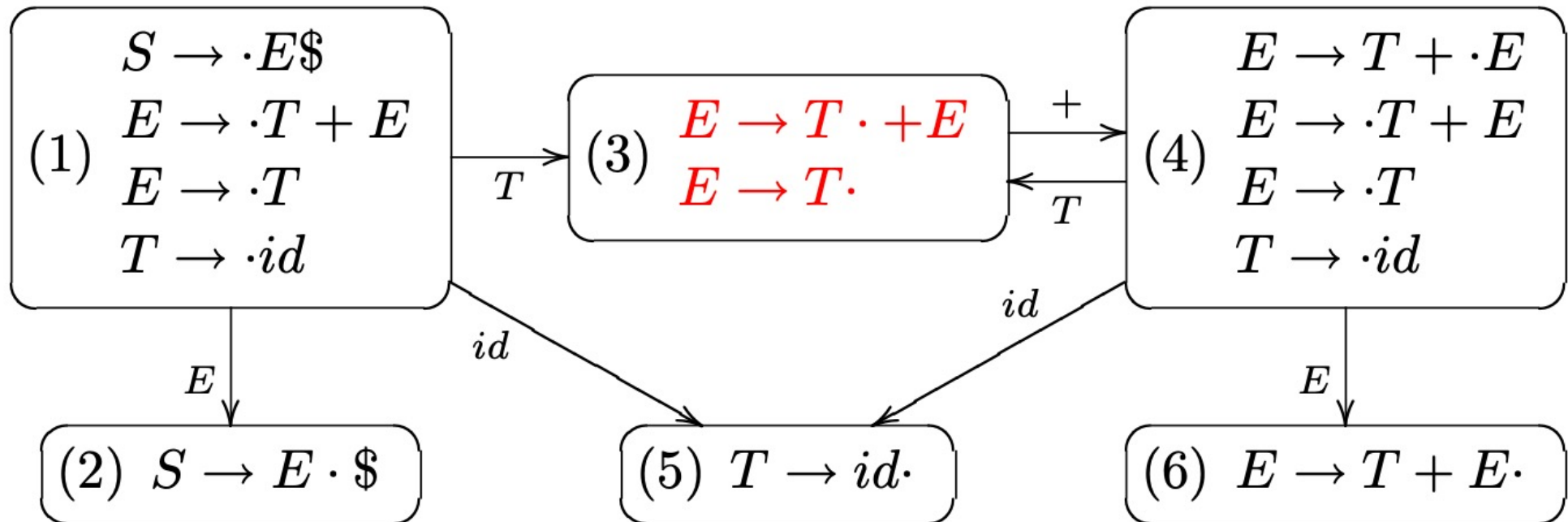
8. Analyse ascendante déterministe

Analyseurs SLR Exemple

$S \rightarrow E$ (0)
 $E \rightarrow T + E$ (1)
 $E \rightarrow T$ (2)
 $T \rightarrow id$ (3)

Est-ce que la grammaire est LR(0)? LR(1)?

Items LR(0) et automate LR(0):



8. Analyse ascendante déterministe



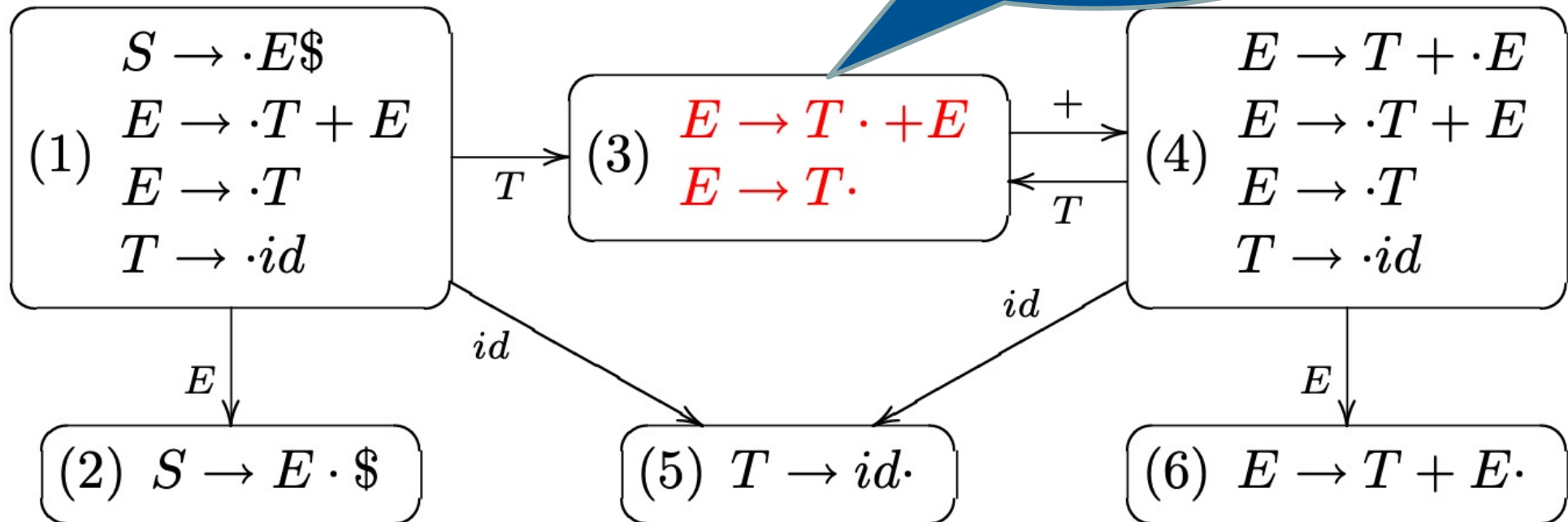
Analyseurs SLR Exemple

- $S \rightarrow E$ (0)
- $E \rightarrow T + E$ (1)
- $E \rightarrow T$ (2)
- $T \rightarrow id$ (3)

Est-ce que la grammaire est LR(0)? LR(1)?

L'état (3) : Conflit
Décaler/Reduire
→ **non LR(0)**

Items LR(0) et automate LR(0):



8. Analyse ascendante déterministe

Analyseurs SLR Exemple

$S \rightarrow E$ (0)
 $E \rightarrow T + E$ (1)
 $E \rightarrow T$ (2)
 $T \rightarrow id$ (3)

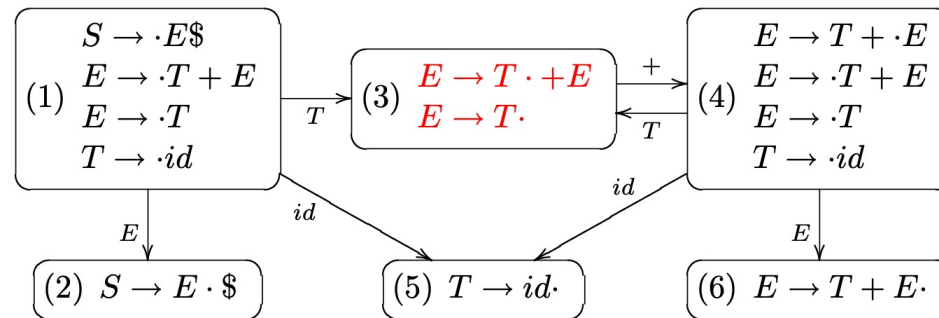


Table d'Analyse LR(0):

	Action			GOTO	
	+	id	\$	E	T
1		s5		2	3
2			Acc		
3	r2,s4	r2	r2		
4		s7		6	3
5	r3	r3	r3		
6	r1	r1	r1		

Donc dans la table d'analyse LR(0) on trouve un **conflit décaler/réduire** dans la case (3,+).

8. Analyse ascendante déterministe

Analyseurs SLR

Exemple

$S \rightarrow E$ (0)
 $E \rightarrow T + E$ (1)
 $E \rightarrow T$ (2)
 $T \rightarrow id$ (3)

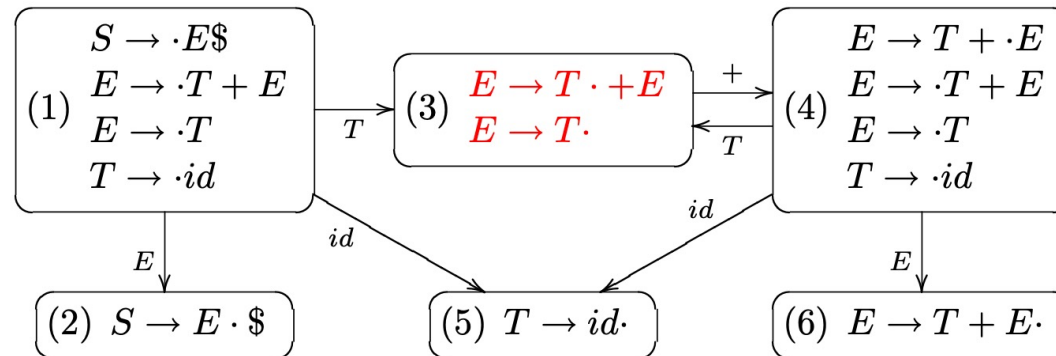


Table d'Analyse **SLR**:

	Action			GOTO	
	+	id	\$	E	T
1		s5		2	3
2			Acc		
3	s4		r2		
4		s5		6	3
5	r3		r3		
6			r1		

La table **SLR** contiendra des entrées Réduire seulement sur certains nonterminaux, **pas tous!** En particulier, on pourra éviter le conflit décaler/réduire dans l'état (3) $\{(E \rightarrow T \cdot + E), (E \rightarrow T \cdot)\}$.

r2: dans \$ car $\text{suiv}(E) = \{\$\}$
r3: dans + et \$ car $\text{suiv}(T) = \{+, \$\}$

8. Analyse ascendante déterministe

Analyseurs SLR Exemple

$S \rightarrow E$ (0)
 $E \rightarrow T + E$ (1)
 $E \rightarrow T$ (2)
 $T \rightarrow id$ (3)

Cette grammaire n'est pas LR(0) mais elle est SLR.

Table d'Analyse LR(0):

	Action			GOTO	
	+	id	\$	E	T
1		s5		2	3
2			Acc		
3	r2,s4	r2	r2		
4		s7		6	3
5	r3	r3	r3		
6	r1	r1	r1		

Table d'Analyse SLR:

	Action			GOTO	
	+	id	\$	E	T
1		s5		2	3
2			Acc		
3	s4		r2		
4		s5		6	3
5	r3		r3		
6			r1		

8. Analyse ascendante déterministe

Analyseurs LALR(1)



- LALR (1)** (Look-Ahead **LR(1)**) est une classe d'analyseurs dont
- l'automate est obtenu de l'automate LR(1) en fusionnant les états qui diffèrent seulement par leur look-ahead (suivants).
 - ✓ On dit aussi que l'on fusionne les états ayant le même cœur, c'est des états identiques excepté leurs suivants.
 - ✓ un analyseur LALR(1) a autant d'états qu'un LR(0) ou SLR et donc une table plus réduite que LR(1).
 - ✓ Les analyseurs LALR(1) sont les plus utilisés parce que, même s'ils ont moins d'états qu'un analyseur LR(1), il est très rare qu'on retrouve un conflit dans la table LALR(1) quand il n'y en a pas dans la table LR(1).

Algorithme : Calculer les items LR(1) puis fusionner les items avec le même cœur.

8. Analyse ascendante déterministe

Analyseurs LALR(1)

Exemple



$S' \rightarrow S$
 $S \rightarrow CC$
 $C \rightarrow cC \mid d$

$I_0: S' \rightarrow .S, \$$
 $S \rightarrow .CC, \$$
 $C \rightarrow .cC, c/d$
 $C \rightarrow .d, c/d$

$I_1: S' \rightarrow S., \$$

$I_2: S \rightarrow C.C, \$$
 $C \rightarrow .cC, \$$
 $C \rightarrow .d, \$$

$I_4: C \rightarrow d., c/d$

$I_9: C \rightarrow cC., \$$

$I_3: C \rightarrow c.C, c/d$
 $C \rightarrow .cC, c/d$
 $C \rightarrow .d, c/d$

$I_5: S \rightarrow CC., \$$

$I_6: C \rightarrow c.C, \$$
 $C \rightarrow .cC, \$$
 $C \rightarrow .d, \$$

$I_7: C \rightarrow d., \$$

$I_8: C \rightarrow cC., c/d$

les items LR(1)

8. Analyse ascendante déf



Analyseurs LALR(1)

Exemple

$S' \rightarrow S$
 $S \rightarrow CC$
 $C \rightarrow cC \mid d$

$I_0: S' \rightarrow .S, \$$
 $S \rightarrow .CC, \$$
 $C \rightarrow .cC, c/d$
 $C \rightarrow .d, c/d$
 $I_1: S' \rightarrow S., \$$
 $I_2: S \rightarrow C.C, \$$
 $C \rightarrow .cC, \$$
 $C \rightarrow .d, \$$
 $I_3: C \rightarrow c.C, c/d$
 $C \rightarrow .cC, c/d$
 $C \rightarrow .d, c/d$
 $I_4: C \rightarrow d., c/d$
 $I_5: S \rightarrow CC., \$$
 $I_6: C \rightarrow c.C, \$$
 $C \rightarrow .cC, \$$
 $C \rightarrow .d, \$$
 $I_7: C \rightarrow d., \$$
 $I_8: C \rightarrow cC., c/d$
 $I_9: C \rightarrow cC., \$$

STATE	action			goto	
	c	d	\$	S	C
0	s3	s4		1	2
1			acc		
2	s6	s7			5
3	s3	s4			8
4	r3	r3			
5			r1		
6	s6	s7			9
7			r3		
8	r2	r2			
9			r2		

La table LR(1)

Analyse dante ministe



Analyseurs LALR(1)

Exemple

$I_0: S' \rightarrow .S, \$$
 $S \rightarrow .CC, \$$
 $C \rightarrow .cC, c/d$
 $C \rightarrow .d, c/d$

$I_1: S' \rightarrow S, .\$$

$I_2: S \rightarrow C.C, \$$
 $C \rightarrow .cC, \$$
 $C \rightarrow .d, \$$

$I_3: C \rightarrow c.C, c/d$
 $C \rightarrow .cC, c/d$
 $C \rightarrow .d, c/d$

$I_4: C \rightarrow d., c/d$

$I_5: S \rightarrow CC., \$$

$I_6: C \rightarrow c.C, \$$
 $C \rightarrow .cC, \$$
 $C \rightarrow .d, \$$

$I_7: C \rightarrow d., \$$

$I_8: C \rightarrow cC., c/d$

$I_9: C \rightarrow cC., \$$

La table LR(1)

STATE	action			goto	
	c	d	\$	S	C
0	s3	s4		1	2
1			acc		
2	s6	s7			5
3	s3	s4			8
4	r3	r3			
5			r1		
6	s6	s7			9
7			r3		
8	r2	r2			
9			r2		

STATE	action			goto	
	c	d	\$	S	C
0	s36	s47		1	2
1			acc		
2	s36	s47			5
36	s36	s47			89
47	r3	r3	r3		
5			r1		
89	r2	r2	r2		

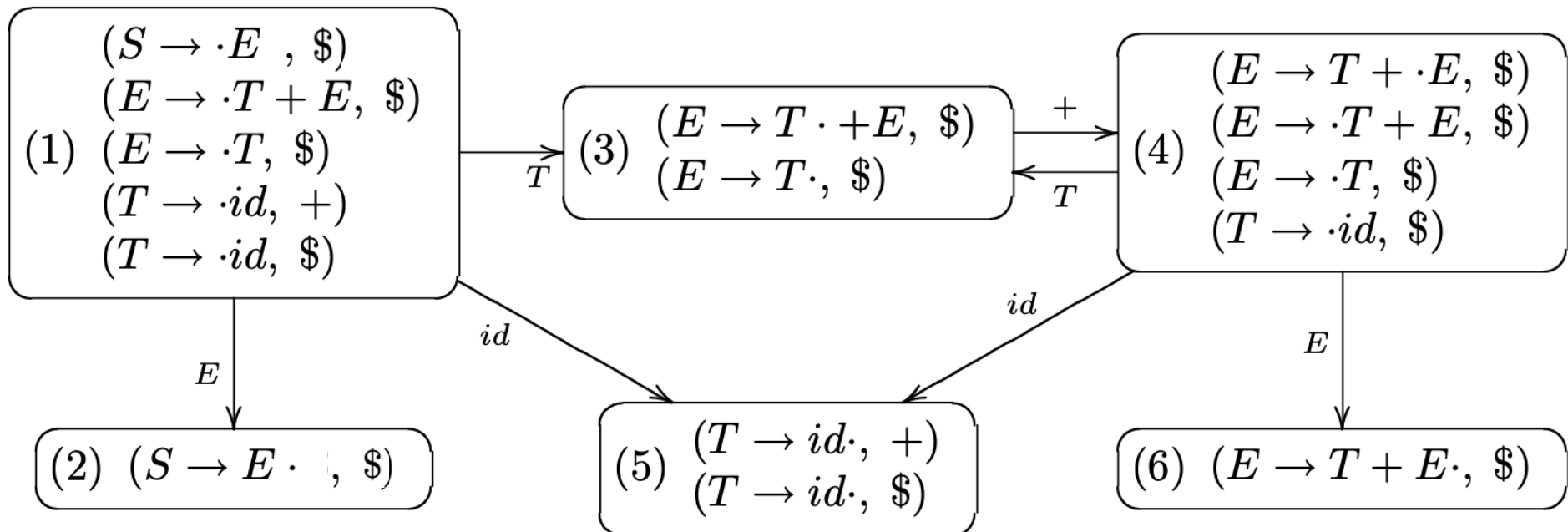
La table LALR(1)

8. Analyse ascendante déterministe

Analyseurs SLR Exemple

$S \rightarrow E$ (0)
 $E \rightarrow T + E$ (1)
 $E \rightarrow T$ (2)
 $T \rightarrow id$ (3)

Items LR(1) et automate LR(1):



8. Analyse ascendante déterministe

Analyseurs SLR Exemple

$S \rightarrow E$ (0)
 $E \rightarrow T + E$ (1)
 $E \rightarrow T$ (2)
 $T \rightarrow id$ (3)

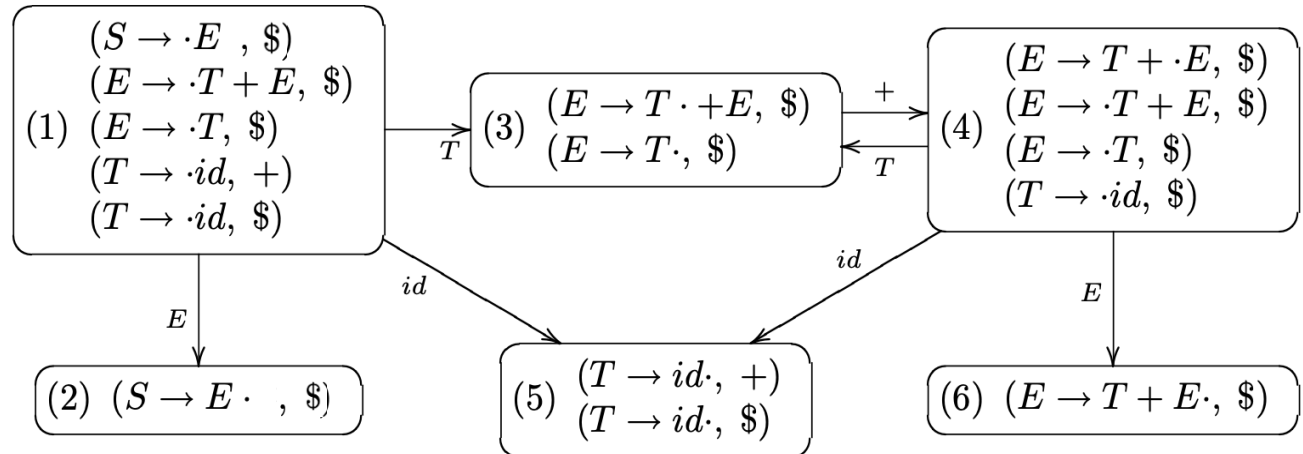


Table d'analyse LR(1):

	Action			GOTO	
	+	id	\$	E	T
1		s5		2	3
2			Acc		
3	s4		r2		
4		s5		6	3
5	r3		r3		
6			r1		



T H E

E N D



PresenterMedia

2016

PresenterMedia