

# Les analyseurs SLR(1)

*INFO010 – Théorie des langages – Partie  
pratique*

S. COLLETTE et G. GEERAERTS



# Introduction

- LR(1) est une méthode efficace et qui fonctionne pour de nombreuses grammaires...
- ... mais ses tables sont souvent très grandes.
- SLR(1) tente de pallier ce problème.

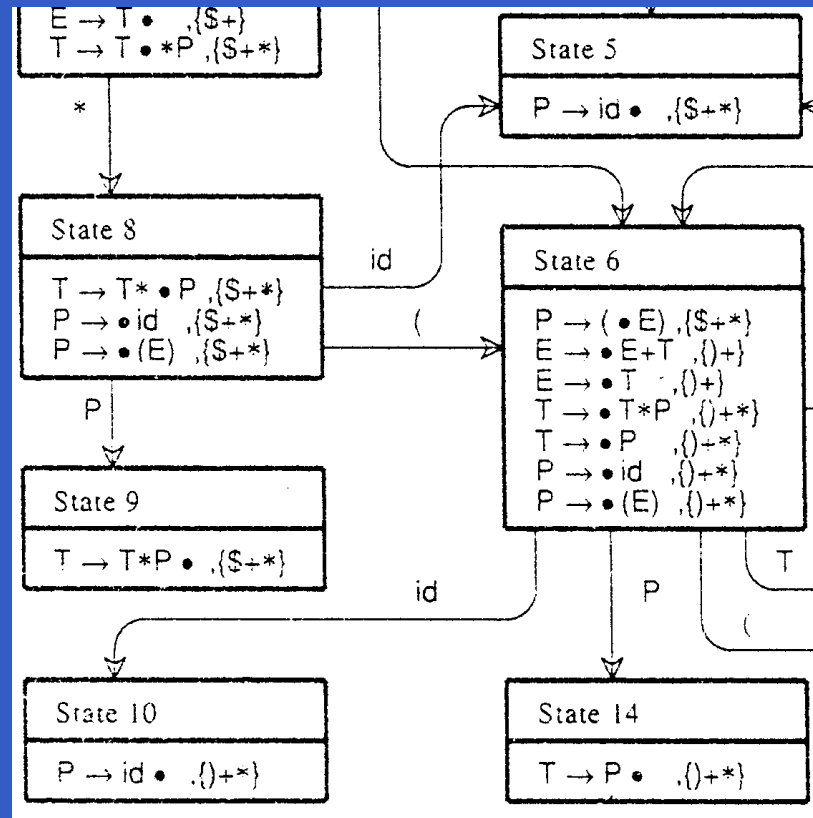
**Idée** : On commence par construire le CFSM LR(0). On calcule **ensuite** les *follows* de chaque non-terminal. On se sert de cette information pour lever les conflits.

# Introduction – LR(1) vs SLR(1)

- En quoi SLR(1) diffère-t-elle de LR(1) ?
- Dans LR(1), on construit l'automate et les *follows* **en même temps** ;
  - On a donc des *follows* **locaux**.
- Dans SLR(1), les *follows* sont construits **après l'automate** ;
  - On a donc des *follows* **globaux**.
- SLR(1) est donc **moins précise** mais **plus compacte** que LR(1).

# Exemple intuitif

Dans l'exemple suivant :



On pourrait très bien fusionner les états 5 et 10...

# Analyseur SLR(1) – Table des actions

Ayant les items LR(0), on construit la tables des actions comme suit ( $a \in T$ ) :

**pour chaque état  $s$  du CFSM faire**

**si  $s$  contient  $A \rightarrow \alpha \bullet a\beta$  alors**

└ Action[ $s, a$ ]  $\leftarrow$  Action[ $s, a$ ]  $\cup$  Shift ;

**sinon si  $s$  contient  $A \rightarrow \alpha \bullet$ , qui est la règle  $i$  alors**

└ **pour chaque  $a \in Follow^1(A)$  faire**

└└ Action[ $s, a$ ]  $\leftarrow$  Action[ $s, a$ ]  $\cup$  Reduce $_i$  ;

**sinon si  $s$  contient  $S' \rightarrow S\$ \bullet$  alors**

└ Action[ $s$ ]  $\leftarrow$  Action[ $s$ ]  $\cup$  Accept ;

# Exercice 1

Construisez l'analyseur SLR(1) pour la grammaire :

$$(1) \quad S' \rightarrow S\$$$

$$(2) \quad S \rightarrow A$$

$$(3) \quad A \rightarrow bB$$

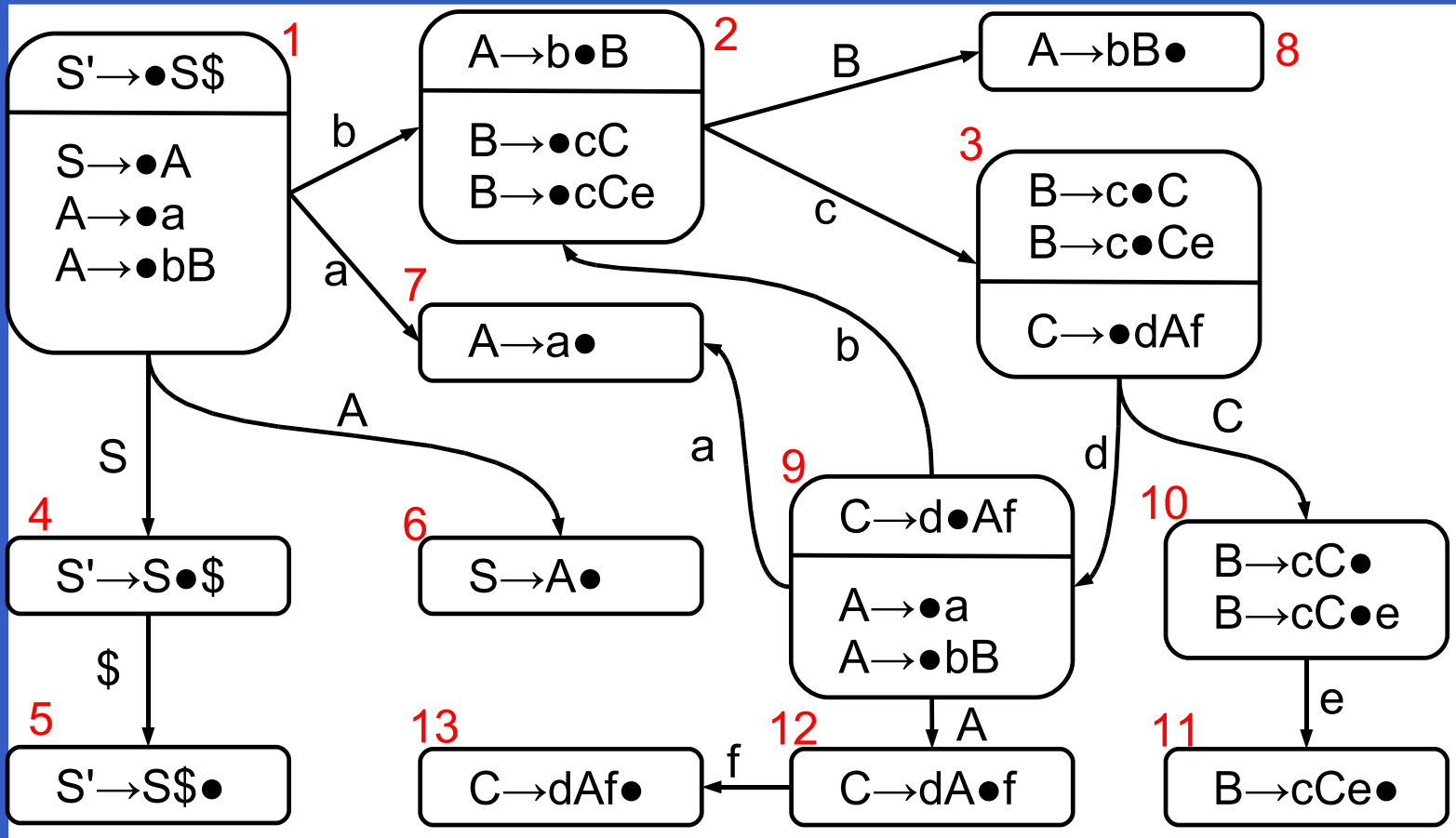
$$(4) \quad A \rightarrow a$$

$$(5) \quad B \rightarrow cC$$

$$(6) \quad B \rightarrow cCe$$

$$(7) \quad C \rightarrow dAf$$

# Exercice 1 – Correction



# Exercice 1 – Correction

- $\text{Follow}^1(S) = \{\$, \}$
- $\text{Follow}^1(A) = \{\mathbf{f}, \$\}$
- $\text{Follow}^1(B) = \{\mathbf{f}, \$\}$
- $\text{Follow}^1(C) = \{\mathbf{e}, \mathbf{f}, \$\}$



# Exercice 1 – Correction

	a	b	c	d	e	f	\$
1	S	S					
2			S				
3				S			
4							S
5	Accept						
6							R2
7						R4	R4

# Exercice 1 – Correction

	a	b	c	d	e	f	\$
8						R3	R3
9	S	S					
10					S	R5	R5
11						R6	R6
12						S	
13					R7	R7	R7

# LALR(1) – Cœur d'un état

$S \rightarrow a \bullet Ab$	$\{x\}$
$A \rightarrow \bullet c$	$\{b\}$
$A \rightarrow \bullet cE$	$\{b\}$

# LALR(1) – Cœur d'un état

$S \rightarrow a \bullet Ab$	$\{x\}$
$A \rightarrow \bullet c$	$\{b\}$
$A \rightarrow \bullet cE$	$\{b\}$

**Cœur d'un état** = ensemble des *handles* sans les contextes

# LALR(1) – Principe

- Méthode plus puissante que SLR(1) :
  - Construire l'automate LR(1)
  - Fusionner les états qui ont le même cœur (en prenant l'union des contextes)
    - p. ex. si les deux cœurs contiennent  $A \rightarrow \beta_1 \bullet \beta_2 \{x_1\}$  et  $A \rightarrow \beta_1 \bullet \beta_2 \{x_2\}$ , on trouvera  $A \rightarrow \beta_1 \bullet \beta_2 \{x_1, x_2\}$  dans la fusion.

# Exercice 2

Construisez l'automate LALR(1) pour la grammaire de l'exercice 1.

# Exercice 2 – Correction

On repart de l'automate LR(1) déjà construit et on procède aux fusions suivantes :

● 2 et 16

● 3 et 12

● 5 et 17

● 6 et 15

● 7 et 13

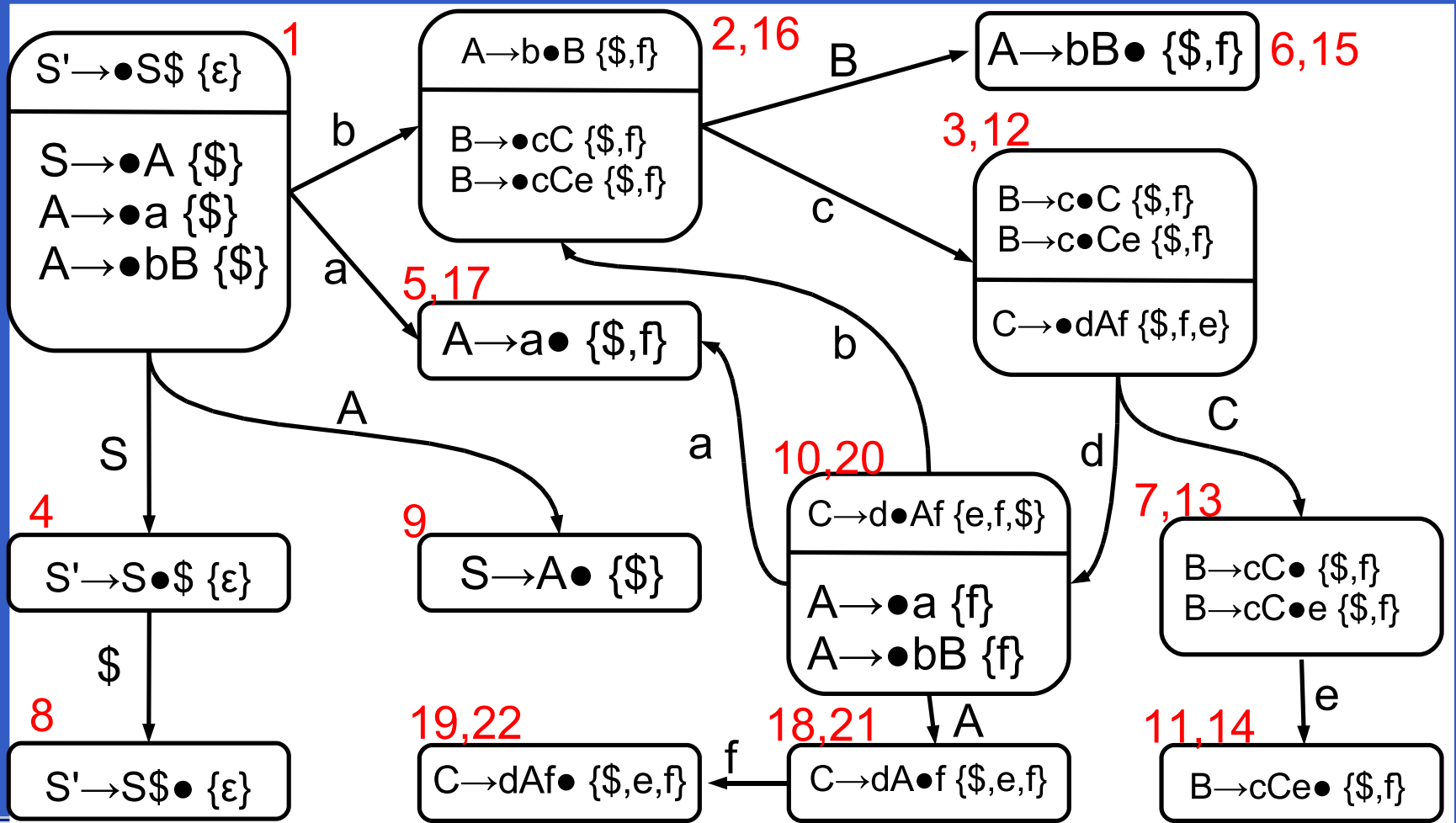
● 10 et 20

● 11 et 14

● 18 et 21

● 19 et 22

# Exercice 2 – Correction





# Exercice 2 – Correction

	a	b	c	d	e	f	\$
1	S	S					
2,16			S				
3,12				S			
4							S
5,17						R4	R4
6,15						R3	R3

# Exercice 2 – Correction

	a	b	c	d	e	f	\$
7,13					S	R5	R5
8	Accept						
9							R2
10,20		S	S				
11,14						R6	R6
18,21						S	
19,22					R7	R7	R7