

Exemplo

- $S' \rightarrow S \$$

1. $S \rightarrow V = E$

2. $S \rightarrow E$

3. $E \rightarrow V$

4. $V \rightarrow x$

5. $V \rightarrow *E$

É SLR ?

LR(1)

Mais poderoso do que SLR

A maioria das linguagens de programação é LR(1)

Algoritmo similar a LR(0), mas agora incorpora o lookahead.

Item: $(A \rightarrow \alpha.\beta, x)$ — **Lookahead**

Algoritmos – Closure(I) e Goto(I,X)

```
Closure ( $I$ ) =  
  repeat  
    for any item  $A \rightarrow (\alpha.X\beta, z)$  in  $I$   
      for any production  $X \rightarrow \gamma$   
        for any  $w \in \text{FIRST}(\beta z)$   
           $I \leftarrow I \cup \{X \rightarrow \cdot\gamma, w\}$   
  until  $I$  does not change.  
  return  $I$ 
```

```
Goto ( $I, X$ ) =  
  
  set  $J$  to the empty set  
  
  for any item  $A \rightarrow (\alpha.X\beta, z)$  in  $I$   
    add  $A \rightarrow (\alpha X.\beta, z)$  to  $J$   
  
  return Closure ( $J$ )
```

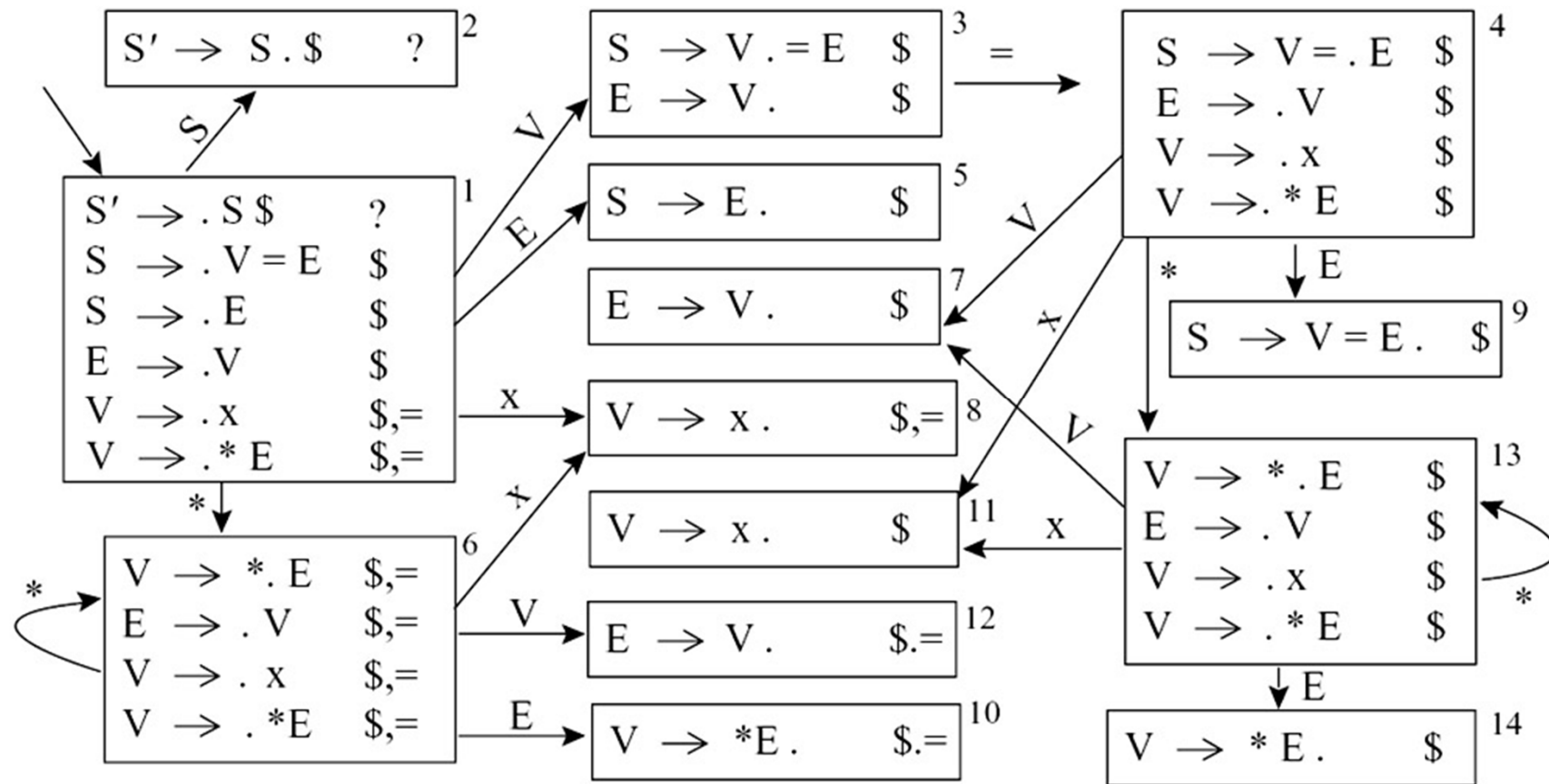
LR(1)

- $S' \rightarrow S \$$
 1. $S \rightarrow V = E$
 2. $S \rightarrow E$
 3. $E \rightarrow V$
 4. $V \rightarrow x$
 5. $V \rightarrow *E$

$S' \rightarrow . S \$$?
$S \rightarrow . V = E$	\$
$S \rightarrow . E$	\$
$E \rightarrow . V$	\$
$V \rightarrow . x$	\$
$V \rightarrow . * E$	\$
$V \rightarrow . x$	=
$V \rightarrow . * E$	=

$S' \rightarrow . S \$$?
$S \rightarrow . V = E$	\$
$S \rightarrow . E$	\$
$E \rightarrow . V$	\$
$V \rightarrow . x$	\$, =
$V \rightarrow . * E$	\$, =

LR(1)



- $S' \rightarrow S \$$
- 1. $S \rightarrow V = E$
- 2. $S \rightarrow E$
- 3. $E \rightarrow V$
- 4. $V \rightarrow x$
- 5. $V \rightarrow * E$

LR(1)

	x	*	=	\$	S	E	V
1	s8	s6			g2	g5	g3
2				a			
3			s4	r3			
4	s11	s13				g9	g7
5				r2			
6	s8	s6				g10	g12
7				r3			
8			r4	r4			
9				r1			
10			r5	r5			
11				r4			
12			r3	r3			
13	s11	s13				g14	g7
14				r5			

LR(1)

$R \leftarrow \{\}$

for each state I in T

for each item $(A \rightarrow \alpha., z)$ in I

$R \leftarrow R \cup \{(I, z, A \rightarrow \alpha)\}$

LALR(1)

O tamanho das tabelas LR(1) pode ser muito grande.

É possível reduzir unindo estados do DFA.

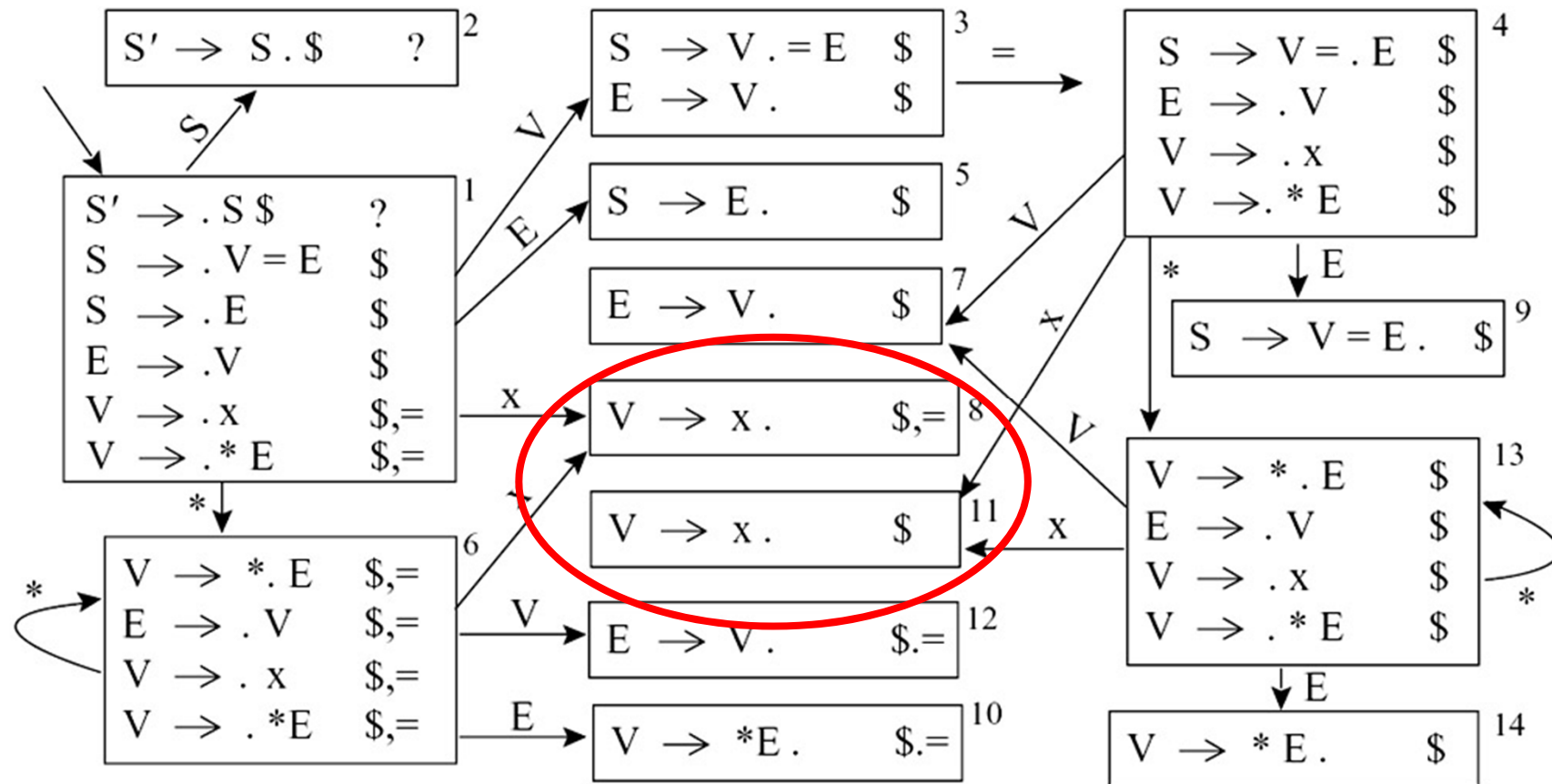
Junte os estados que possuam os itens idênticos, exceto pelo lookahead.

Look-Ahead Left-to-right parsing, Rightmost-derivation, 1-symbol lookahead

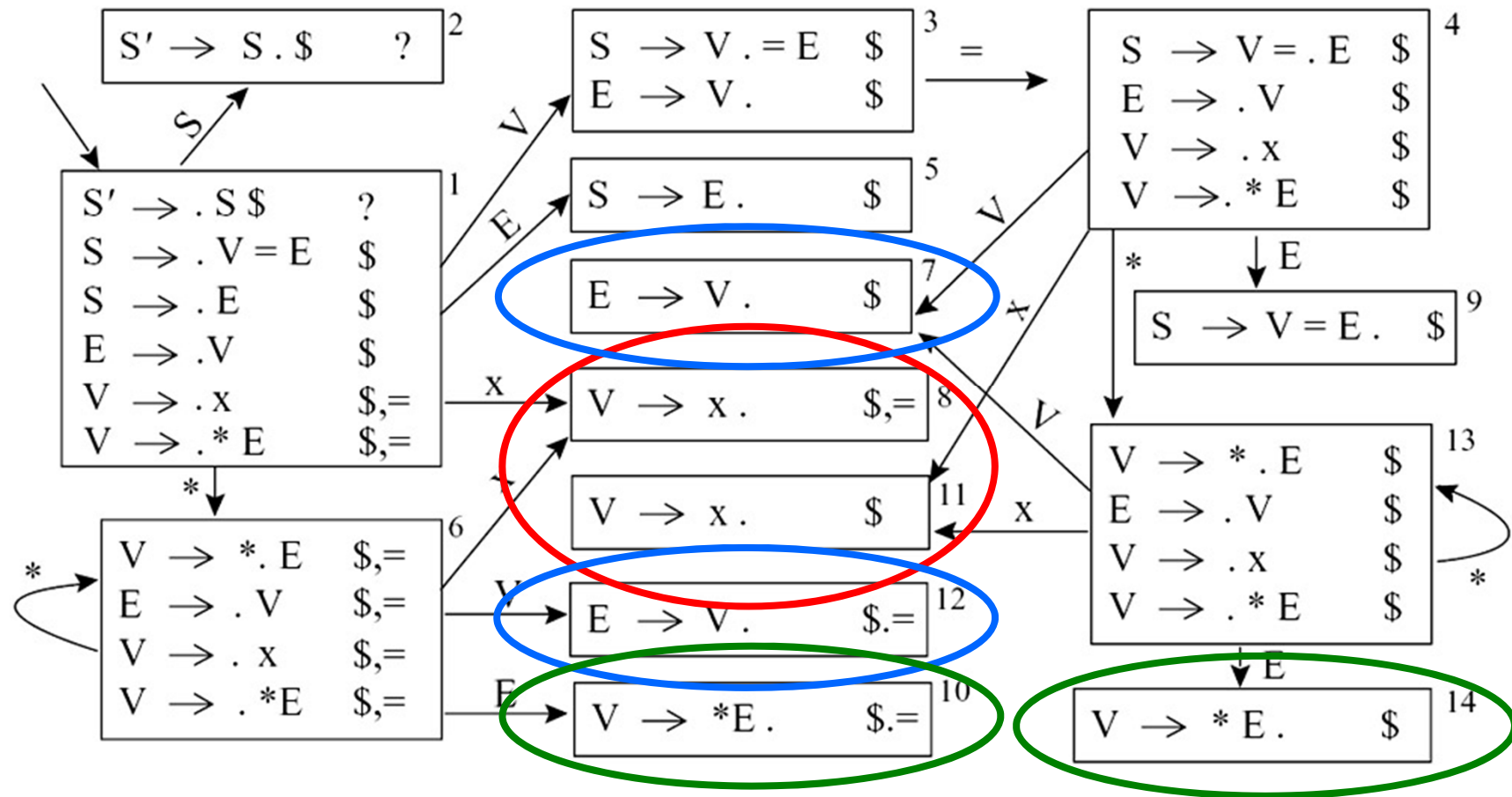
Voltando ao exemplo anterior...

LALR(1)

Mais algum???



LALR(1)



LALR(1)

	x	*	=	\$	S	E	V
1	s8	s6			g2	g5	g3
2				a			
3			s4	r3			
4	s11	s13				g9	g7
5				r2			
6	s8	s6				g10	g12
→ 7				r3			
→ 8			r4	r4			
9				r1			
→ 10			r5	r5			
→ 11				r4			
→ 12			r3	r3			
13	s11	s13				g14	g7
→ 14				r5			

(a) LR(1)

	x	*	=	\$	S	E	V
1	s8	s6			g2	g5	g3
2				a			
3			s4	r3			
4	s8	s6				g9	g7
5				r2			
6	s8	s6				g10	g7
7			r3	r3			
8			r4	r4			
9				r1			
10			r5	r5			

(b) LALR(1)

LALR(1)

- Pode gerar uma tabela com conflitos, onde a LR(1) não possuía.
- Na prática, o efeito de redução no uso de memória é bastante desejável.
- A maioria das linguagens de programação é LALR(1).
- É o tipo mais usado em geradores automáticos de *parser*.