ShiftReduce

Go <P> → begin <S> and $
<S> → SS ; <S>
<S> → begin <S> and ; <S>
<S> → λ

Symbol	State											
	0	1	2	3	4	5	6	7	8	9	10	11
begin	S	S			S		S			S		
end		R4	S		R4		R4	S		R4	R2	R3
;						S			S			
SimpleStmt		S			S		S			S		
$				A								
<program>												
<stmts>		S			S		S			S		

Figure 6.2 A Shift-Reduce **action** Table for G0

Symbol	State											
	0	1	2	3	4	5	6	7	8	9	10	11
begin	1	4			4		4			4		
end			3					8				
;						6			9			
SimpleStmt		5			5		5			6		
$												
<program>												
<stmts>		2			7		10			11		

Figure 6.3 A Shift-Reduce **go_to** Table for G_0

Step	Parser Stack	Remaining Input	Parser Action
(1)	0	**begin** SimpleStmt ; SimpleStmt ; **end $**	Shift
(2)	0,1	SimpleStmt ; SimpleStmt ; **end $**	Shift
(3)	0,1,5	; SimpleStmt ; **end $**	Shift
(4)	0,1,5,6	SimpleStmt ; **end $**	Shift
(5)	0,1,5,6,5	; **end $**	Shift
(6)	0,1,5,6,5,6	**end $**	Reduce4
(7)	0,1,5,6,5,6,10	**end $**	Reduce2
(8)	0,1,5,6,10	**end $**	Reduce2
(9)	0,1,2	**end $**	Shift
(10)	0,1,2,3	**$**	Accept

Figure 6.4 Example of a Shift-Reduce Parser

	id	$	S
0	1	4	2
1	4	4	4
2	4	3	4
3	4	4	4

Figure 6.10 `go_to` Table

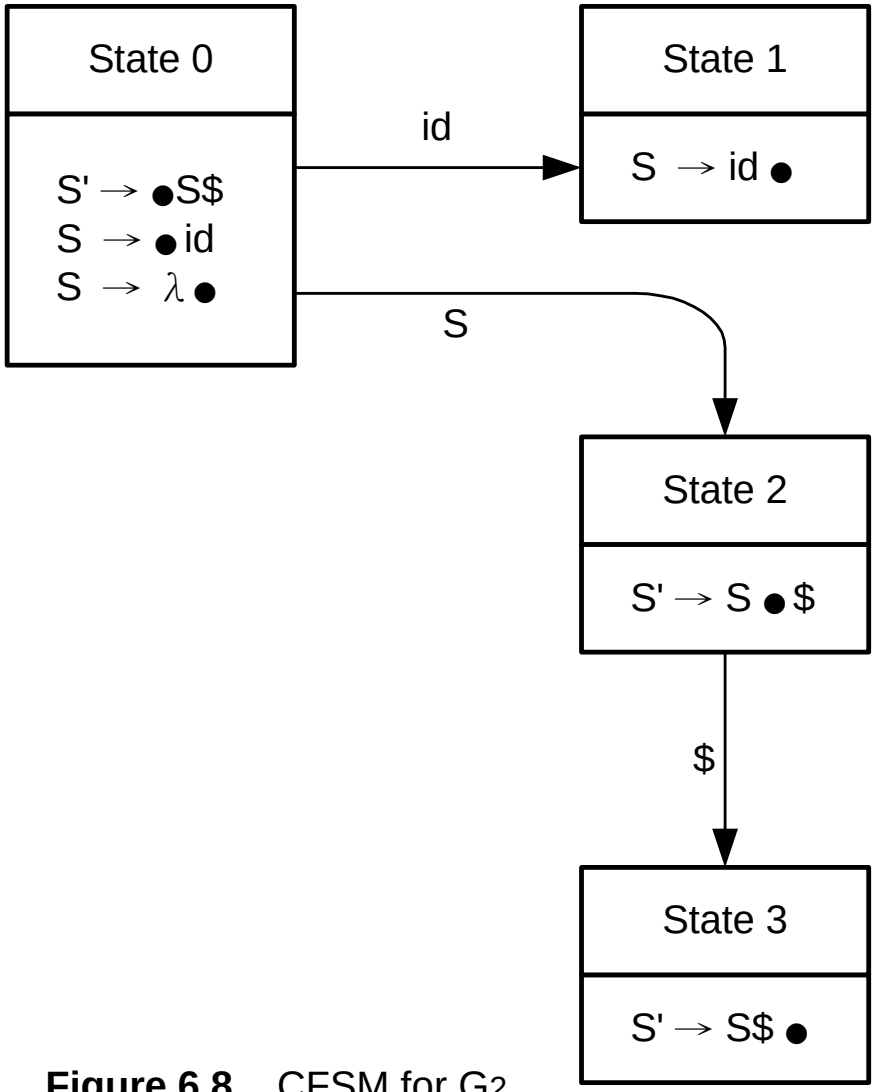

Figure 6.8 CFSM for G_2

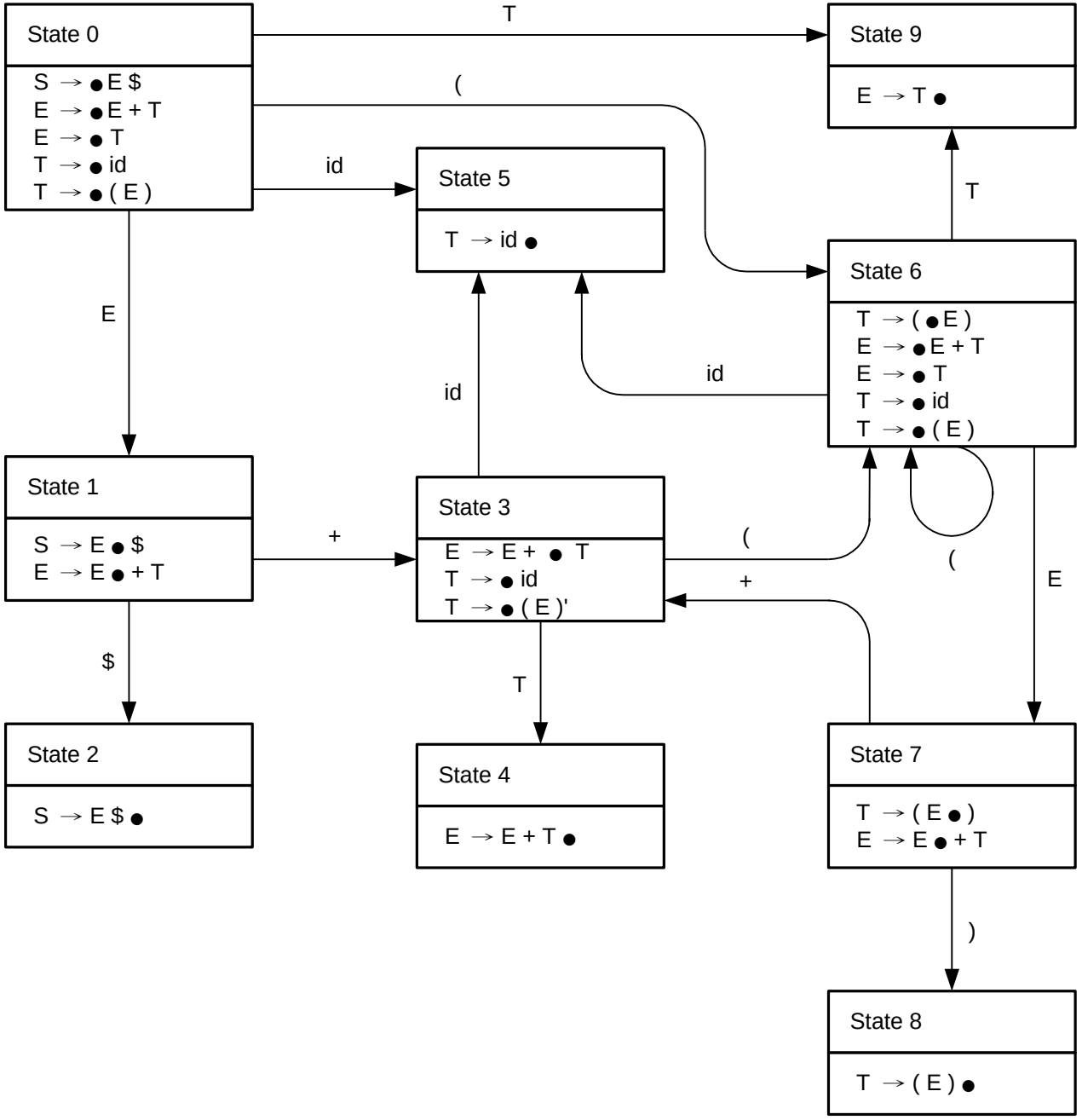

Figure 6.11 CFSM for G1

State	0	1	2	3	4	5	6	7	8	9	10
Adim	S	S	A	S	R2	R4	S	S	R5	R3	

Figure 6.2

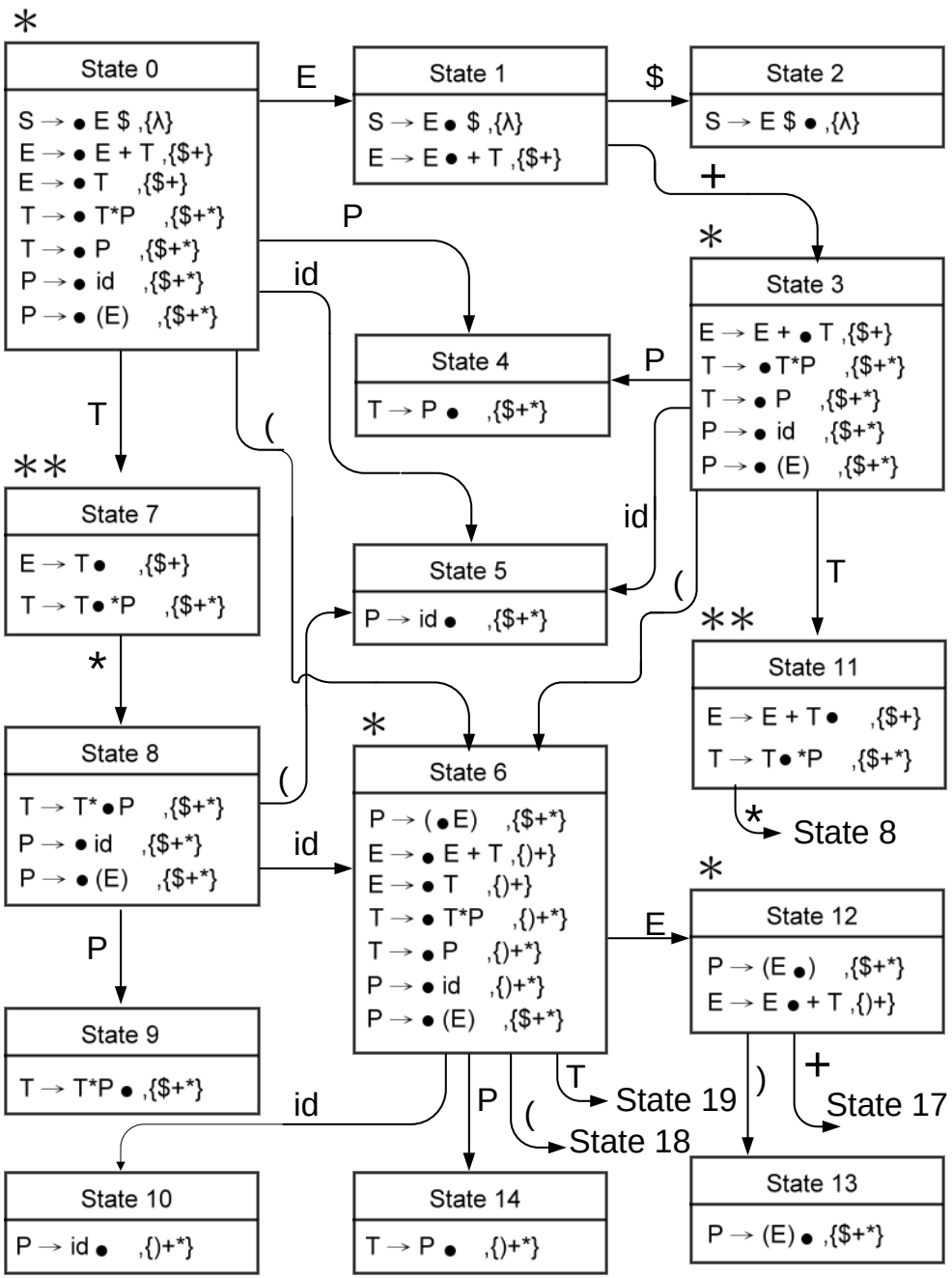

Figure 6.16 LR(1) Machine for G3

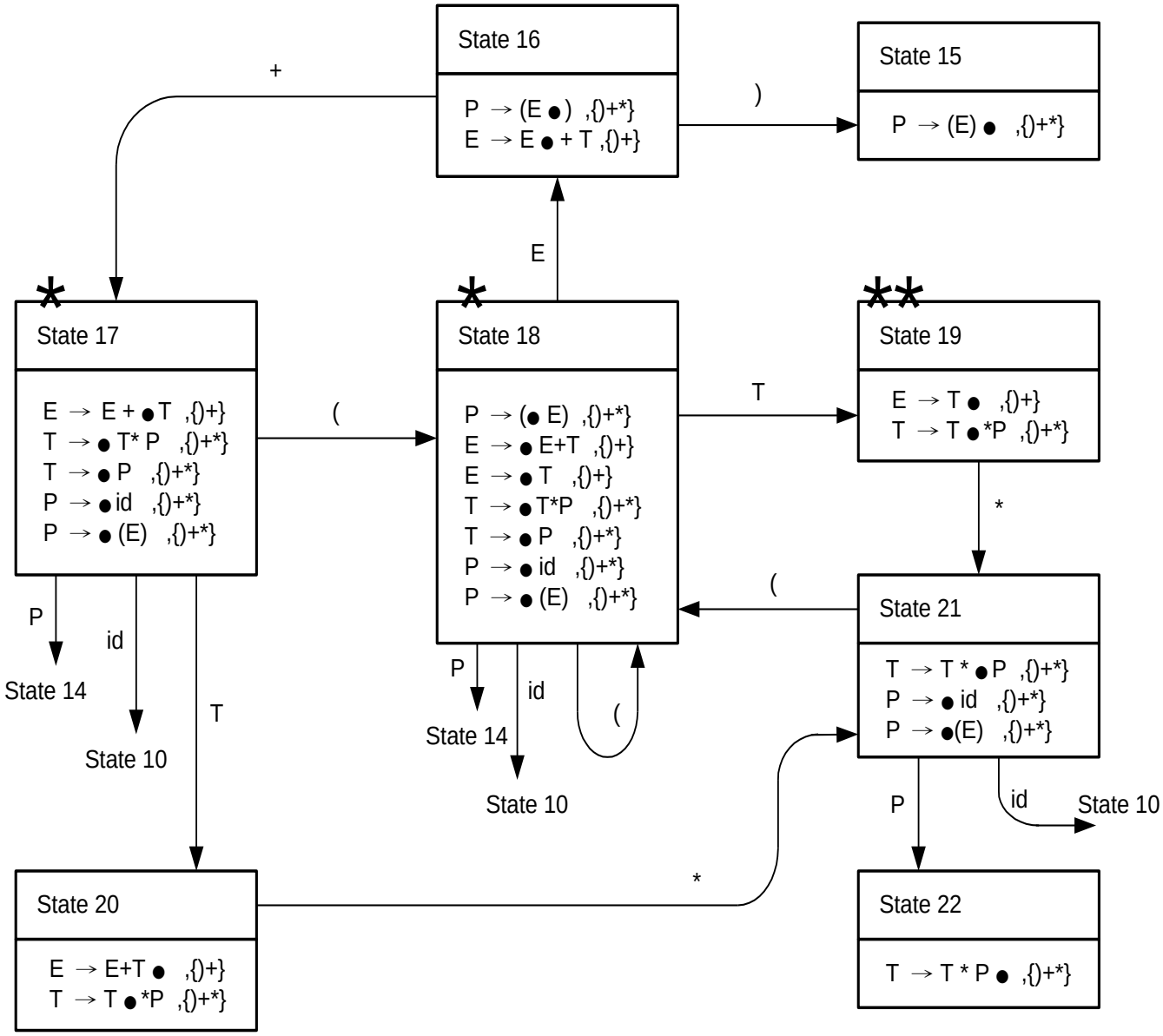

Figure 6.16 (continued)

State	Lookahead					
	+	*	ID	(	)	$
0			S	S		
1	S					A
2						
3			S	S		
4	R5	R5				R5
5	R6	R6				R6
6			S	S		
*7	R3	S				R3
8			S	S		
9	R4	R4				R4
10	R6	R6			R6	

Figure 6.17 LR(1) **action** Function for G3

*11	R2	S				R2
12	S				S	
13	R7	R7				R7
14	R5	R5			R5	
15	R7	R7			R7	
16	S				S	
17			S	S		
18			S	S		
*19	R3	S			R3	
20	R2	S			R2	
21			S	S		
22	R4	R4			R4	

Figure 6.17 (continued)

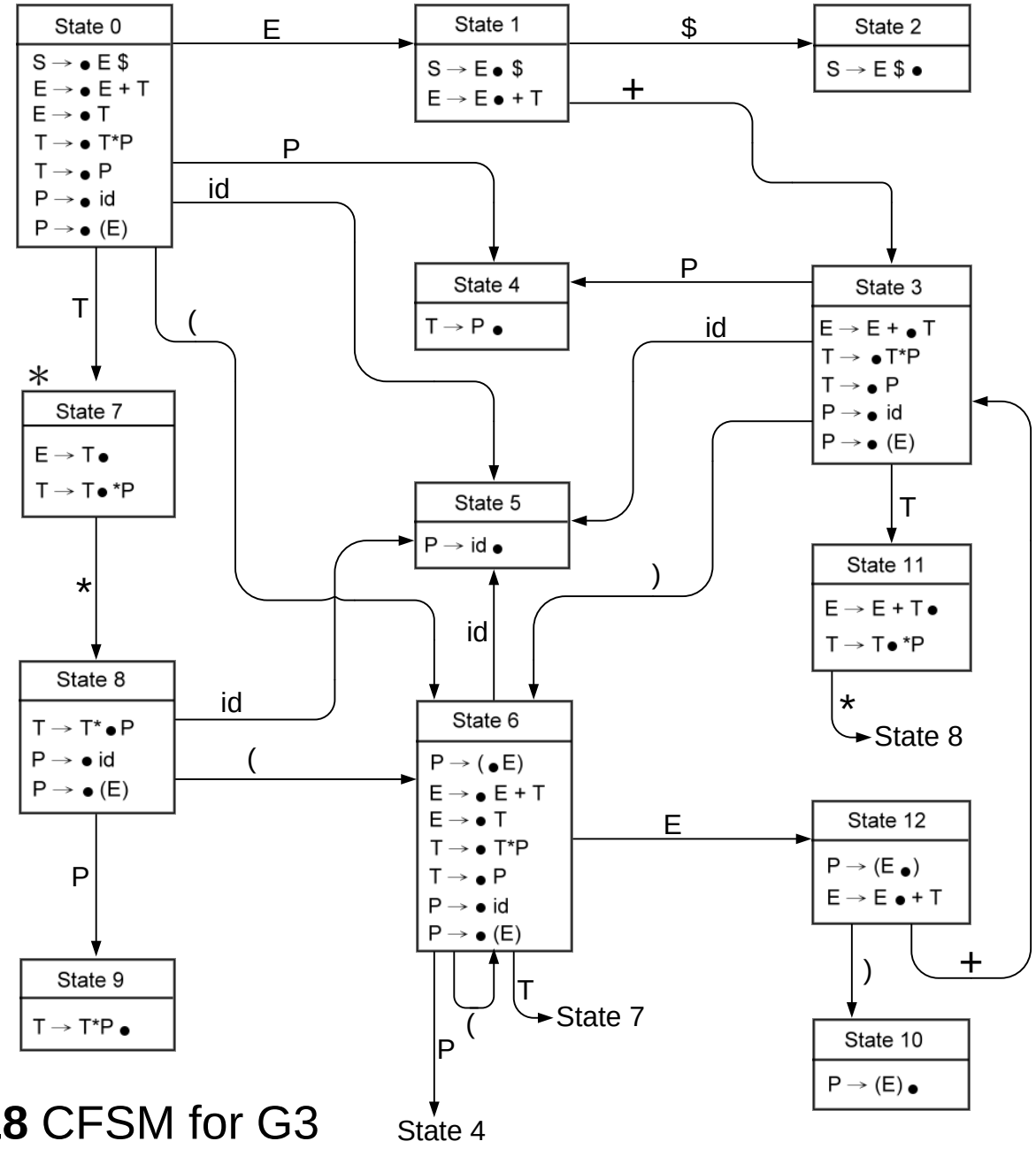

Figure 6.18 CFSM for G3

State	Lookahead					
	+	*	ID	(	)	$
0			S	S		
1	S					A
2						
3			S	S		
4	R5	R5			R5	R5
5	R6	R6			R6	R6
6			S	S		
7	R3	S			R3	R3
8			S	S		
9	R4	R4			R4	R4
10	R7	R7			R7	R7
11	R2	S			R2	R2
12	S				S	

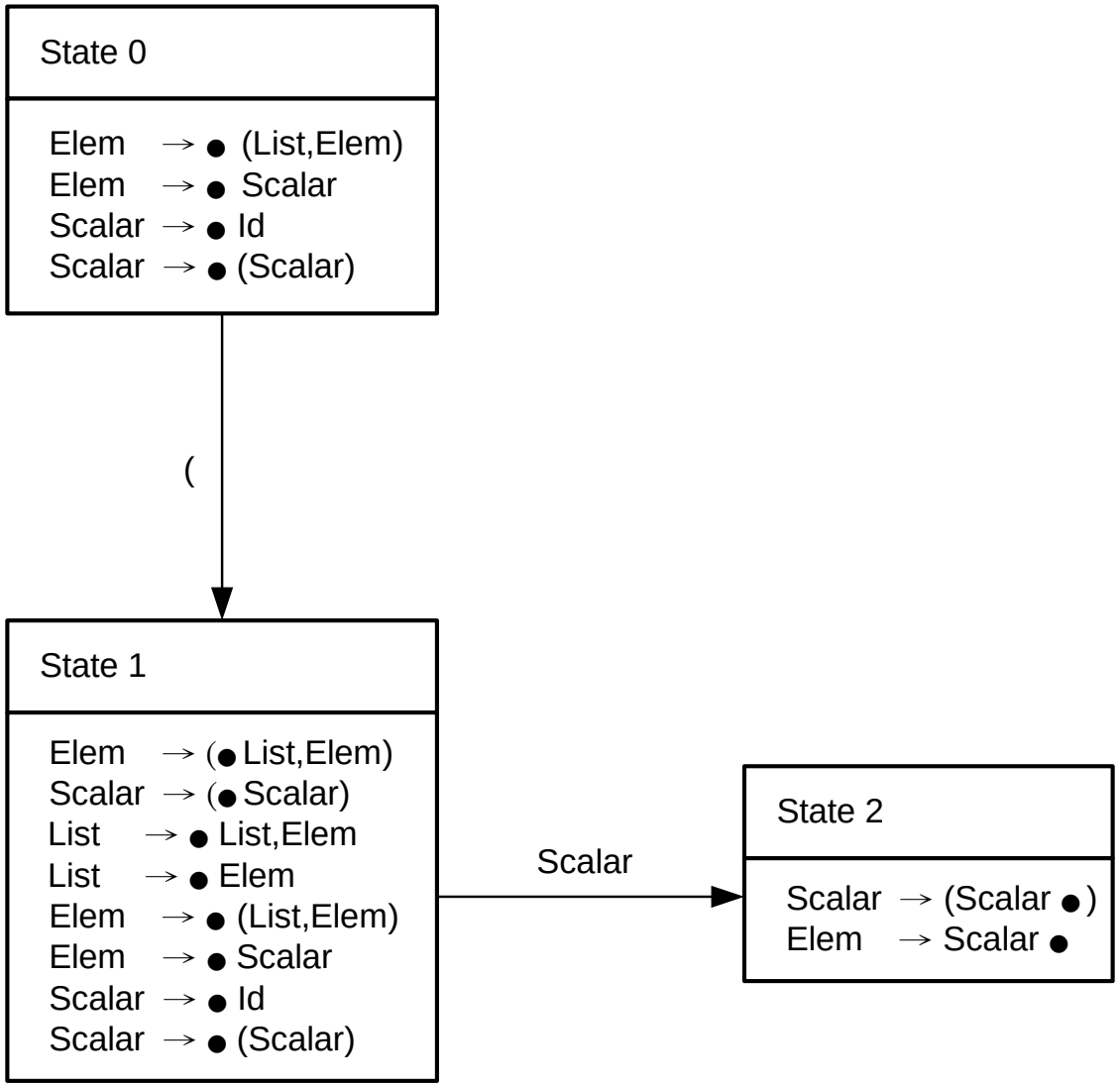

Figure 6.20 Part of the CFSM for G_4

Follow(Scalar) = Follow(Elem) = {)}

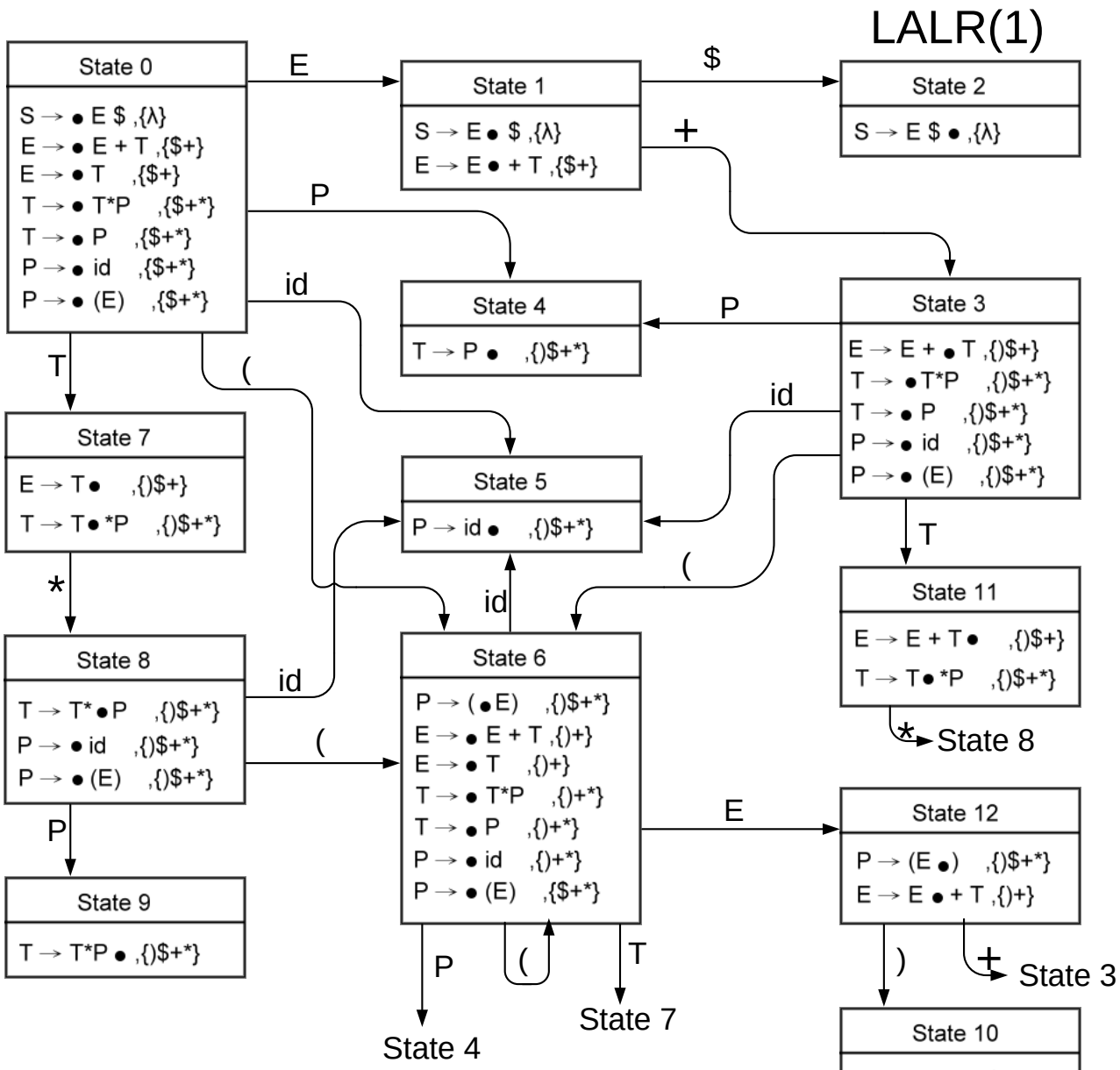

Figure 6.22 LALR(1) Machine for G3

State	Symbol								
	+	*	ID	(	)	$	E	T	P
0			S5	S6			S1	S7	S4*
1	S3					A			
2									
3			S5	S6				S11	S4*
4*	R5	R5			R5	R5			
5	R6	R6			R6	R6			
6			S5	S6			S12	S7	S4*
7	R3	S8			R3	R3			
8			S5	S6					S9
9	R4	R4			R4	R4			
10	R7	R7			R7	R7			
11	R2	S8			R2	R2			
12	S3				S10				

Figure 6.35 SLR(1) Parse Table for G_3

State	Symbol								
	+	*	ID	(	)	$	E	T	P
0			L6	S6			S1	S7	L5*
1	S3					A			
3			L6	S6				S11	L5*
6			L6	S6			S12	S7	L5*
7	R3	S8			R3	R3			
8			L6	S6					L4
11	R2	S8			R2	R2			
12	S3				L7				

Figure 6.36 Optimized SLR(1) Parse Table for G_3

	E	F	T	ID	+	(	)	$
E							≐	≐
F					≐		o>	o>
T					o>		o>	o>
ID					o>		o>	o>
+			≐	<o		<o		
(	≐	<o	<o	<o		<o		
)					o>		o>	o>
$	≐	<o	<o	<o		<o		

Figure 6.40 Simple Precedence Parse Table for G_8

Step	Parse Stack	Remaining Input
1		$ID+(ID+ID)$
2	$ <o	ID+(ID+ID)$
3	$ <o ID o>	+(ID+ID)$
4	$ <o T o>	+(ID+ID)$
5	$ <o F ≐	+(ID+ID)$
6	$ <o F ≐ + <o	(ID+ID)$
7	$ <o F ≐ + <o (<o	ID+ID)$
8	$ <o F ≐ + <o (<o ID o>	+ID)$
9	$ <o F ≐ + <o (<o T o>	+ID)$
10	$ <o F ≐ + <o (<o F ≐	+id)$
11	$ <o F ≐ + <o (<o F ≐ + <o	ID)$

Figure 6.41 Example of a Simple Precedence Parse

12	$ <o F ≗ + <o (<o F ≗ + <o ID o>	)$
13	$ <o F ≗ + <o (<o F ≗ + ≗ T o>	)$
14	$ <o F ≗ + <o (<o F o>	)$
15	$ <o F ≗ + <o (≗ E ≗	)$
16	$ <o F ≗ + <o (≗ E ≗) o>	$
17	$ <o F ≗ + ≗ T o>	$
18	$ <o F o>	$
19	$ ≗ E ≗	$
20	$ ≗ E ≗ $	

Figure 6.41 (continued)