


مدرس: قاسم ثانی	به نام خدا طراحی کامپایلرها پاسخ تمرین سری دوم (نیم سال اول ۹۷-۹۸)	 دانشگاه صنعتی شریف دانشکده‌ی مهندسی کامپیوتر
-----------------	---	--

سوال یک

گرامر زیر را در نظر بگیرید. (توجه کنید پایانه‌ها پررنگ‌تر از غیرپایانه‌ها نمایش داده شده‌اند)

- 1) $Stmts \rightarrow Stmt$
- 2) $Stmts \rightarrow Stmts ; Stmt$
- 3) $Stmt \rightarrow Var = E$
- 4) $Var \rightarrow id [E]$
- 5) $Var \rightarrow id$
- 6) $E \rightarrow id$
- 7) $E \rightarrow (E)$

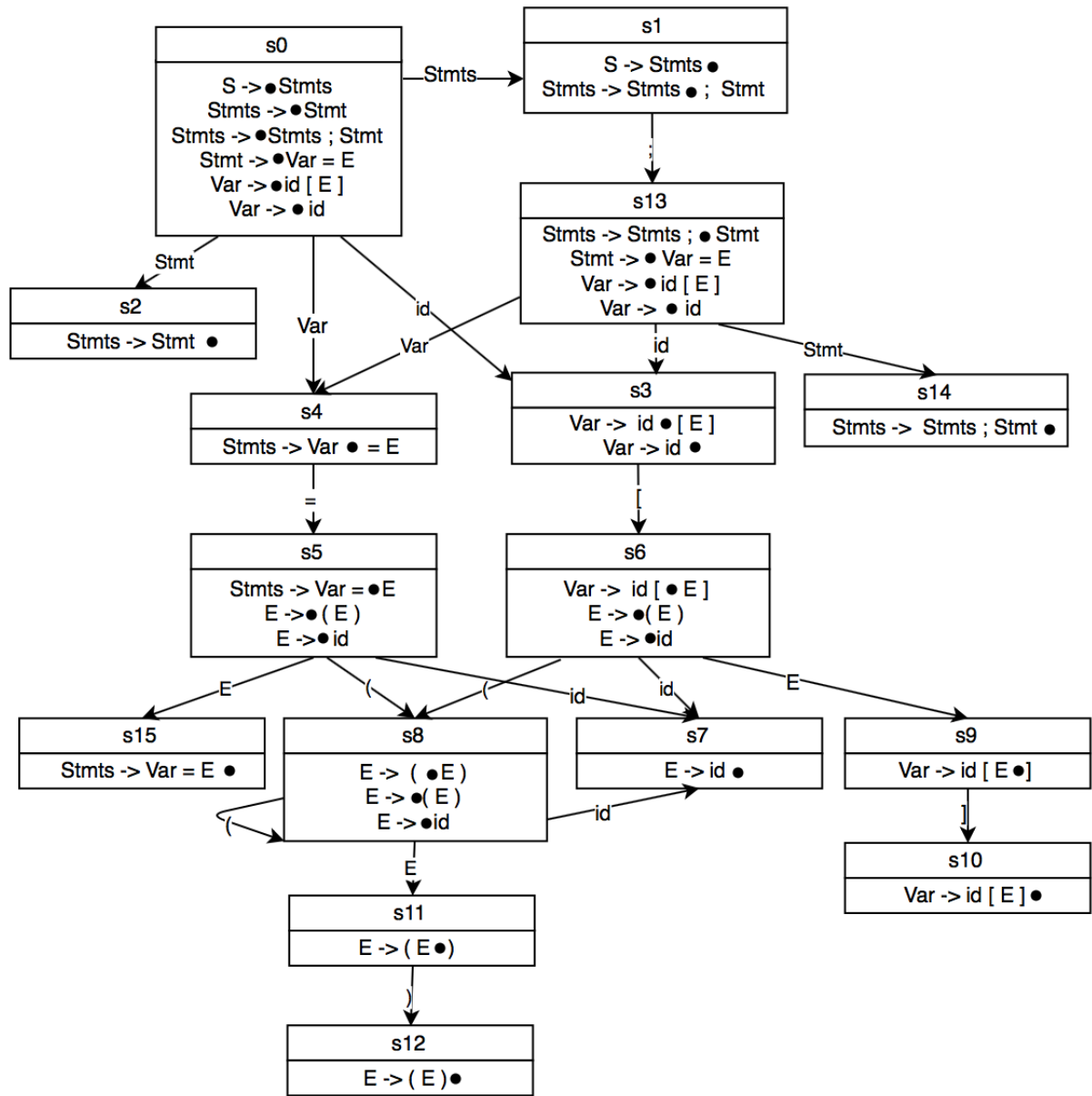
الف) دیاگرام و جدول پارس $SLR(1)$ گرامر فوق را رسم کنید.

پاسخ

ابتدا قاعده‌ای به شکل $S \rightarrow Stmts$ به گرامر فوق اضافه می‌کنیم. بدین ترتیب داریم:

- 0) $S \rightarrow Stmts$
- 1) $Stmts \rightarrow Stmt$
- 2) $Stmts \rightarrow Stmts ; Stmt$
- 3) $Stmt \rightarrow Var = E$
- 4) $Var \rightarrow id [E]$
- 5) $Var \rightarrow id$
- 6) $E \rightarrow id$
- 7) $E \rightarrow (E)$

دیاگرام پارس SLR(1) گرامر فوق به شکل زیر است:



جدول پارس SLR(1) گرامر فوق به شکل زیر است:

State	Terminals								Goto			
	id	;	=	[]	()	\$	Stmts	Stmt	E	Var
0	s3								1	2		4
1		s13						acc				
2		r1						r1				
3			r5	s6								
4			s5									
5	s7					s8					15	
6	s7					s8					9	
7		r6			r6		r6	r6				
8	s7					s8					11	
9					s10							
10			r4									
11							s12					
12		r7			r7		r7	r7				
13	s3									14		4
14		r2						r2				
15		r3						r3				

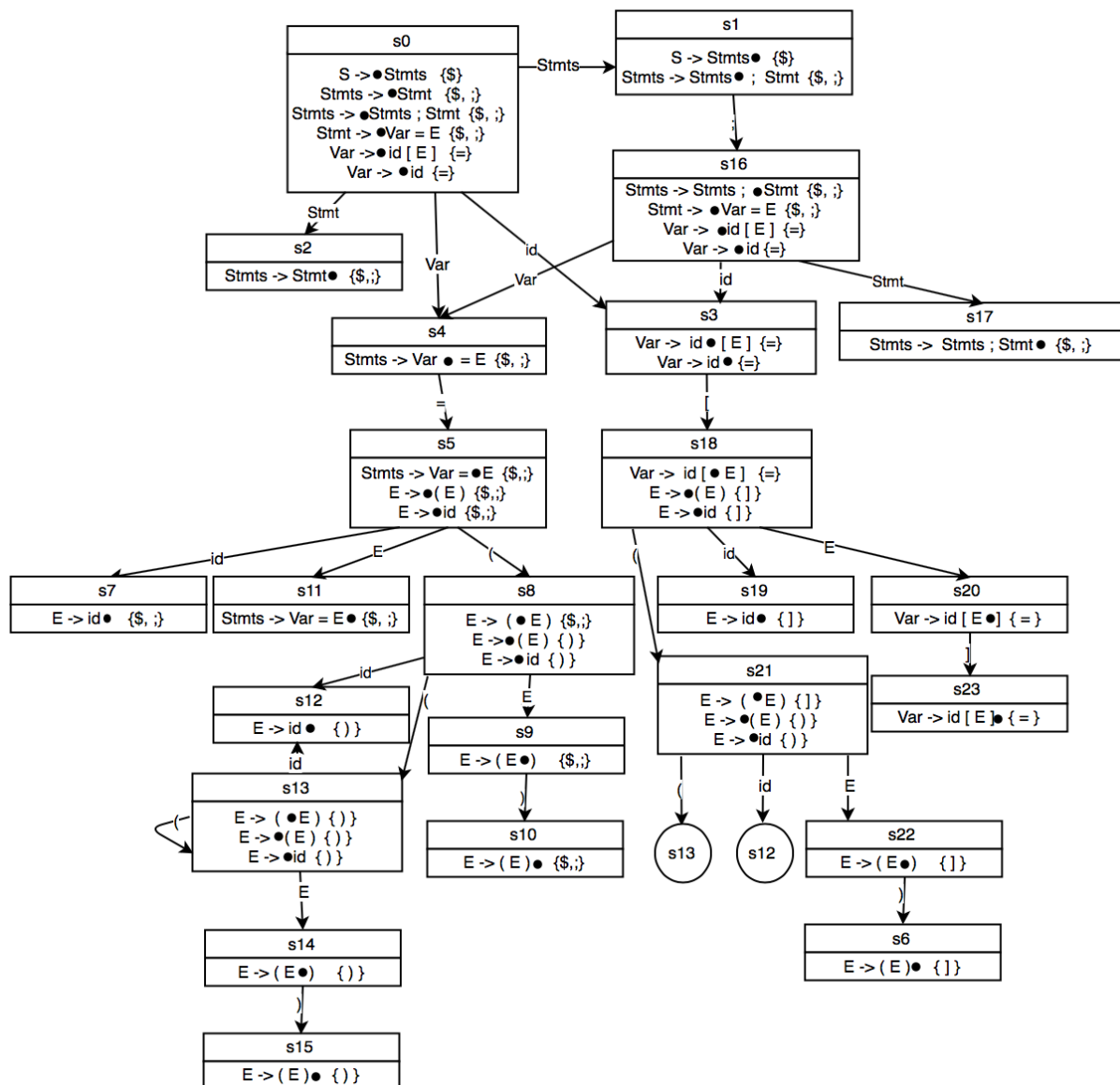
ب) آیا گرامر فوق LR(0) است؟ چرا؟

خیر، زیرا حالت شماره ۳ حالتی مشکوک به تداخل s6/r5 است.

ج) دیاگرام و جدول پارس CLR(1) گرامر فوق را رسم کنید.

پاسخ

دیاگرام پارس (1) CLR گرامر فوق به شکل زیر است:



د) رشته \$ (id) id = id ; را به روش (1) CLR پارس کنید.

پاسخ

Stack	Input	Action
0	id = id ; id = (id) \$	S3
0 id 3	= id ; id = (id) \$	R5
0 Var 4	= id ; id = (id) \$	S5
0 Var 4 = 5	id ; id = (id) \$	S7
0 Var 4 = 5 id 7	; id = (id) \$	R6
0 Var 4 = 5 E 11	; id = (id) \$	R3
0 Stmt 2	; id = (id) \$	R1
0 Stmts 1	; id = (id) \$	S16
0 Stmts 1 ; 16	id = (id) \$	S3
0 Stmts 1 ; 16 id 3	= (id) \$	R5
0 Stmts 1 ; 16 Var 4	= (id) \$	S5
0 Stmts 1 ; 16 Var 4 = 5	(id) \$	S8
0 Stmts 1 ; 16 Var 4 = 5 (8	id) \$	S12
0 Stmts 1 ; 16 Var 4 = 5 (8 id 12) \$	R6
0 Stmts 1 ; 16 Var 4 = 5 (8 E 9) \$	S10
0 Stmts 1 ; 16 Var 4 = 5 (8 E 9) 10	\$	R7
0 Stmts 1 ; 16 Var 4 = 5 E 11	\$	R3
0 Stmts 1 ; 16 Stmt 17	\$	R2
0 Stmts 1	\$	acc

ه) جدول پارس LALR(1) گرامر فوق را رسم کنید.

پاسخ

برای ساخت جدول LALR(1) گرامر داده شده، حالت‌هایی که core item یکسانی دارند را با هم ادغام می‌کنیم:

$$s_{25} = \{s_7, s_{12}, s_{19}\} \quad s_{26} = \{s_6, s_{10}, s_{15}\} \quad s_{27} = \{s_9, s_{14}, s_{22}\} \quad s_{28} = \{s_8, s_{13}, s_{21}\}$$

State	Terminals								Goto			
	id	;	=	[]	()	\$	Stmts	Stmt	E	Var
0	S3								1	2		4
1		S16						acc				
2		R1						R1				
3			R5	S18								
4			S5									
5	S25					S28					11	
11		R3						R3				
16	S3									17		4
17		R2						R2				
18	S25					S28					20	
20					S23							
23			R4									
25		R6			R6		R6	R6				
26		R7			R7		R7	R7				
27							S26					
28	S12					S13					27	

سوال دو

گرامر زیر را در نظر بگیرید که در آن E غیر پایانه مولد عبارات جبری است.

- 1) $S \rightarrow id := E$
- 2) $S \rightarrow \text{case } E \text{ of } L \text{ end}$
- 3) $L \rightarrow id : S L'$
- 4,5,6) $L' \rightarrow \epsilon \mid ; \text{ otherwise } S \mid ; id : S L'$

محل علائم لازم برای تولید کد دستورات case با گرامر فوق را در قواعد شماره ۲ تا ۶ تعیین و روتین‌های مفهومی مربوطه را بنویسید و برای جمله زیر کد سه‌آدرسه تولید کنید.

```
case (c * d) of
  a: a := a + 1;
  b: b := b + 2;
  otherwise: e := c*d
end
```

پاسخ

```
S -> case #JmpSave E of L #Jp end
L -> #Pid id #CmpSave : S L'
L' -> #Jpf
L' -> #Jpf otherwise S
L' -> ; #Pid id #JpCmpSave : S L'
```

```
#JmpSave
  PB[i] <- (JP, i + 2,,)
  i <- i + 1
  push(i)
  i <- i + 1
end
```

```
#CmpSave
  t <- gettemp
  PB[i] <- (=, SS(top), SS(top - 1), t)
  i <- i + 1
  pop(1)
  push(t)
  push(i)
  i <- i + 1
end
```



```

#JpCmpSave
  PB[i] <- (JP, SS(top - 4), ,)
  i <- i + 1
  t <- getTemp
  PB[SS(top - 1)] <- (JPF, SS(top - 2), i,)
  PB[i] <- (=, SS(top), SS(top - 3), t)
  i <- i + 1
  pop(3)
  push(t)
  push(i)
  i <- i + 1
end

#Jpf
  PB[SS(top)] <- (JPF, SS(top - 1), i,)
  pop(3)
end

#Jp
  PB[SS(top)] <- (JP, i, ,)
  pop
end

```

0	(JP, 2, ,)			
1	(JP, 15, ,)			
2	(*, c, d, t1)			
3	(=, t1, a, t2)			
4	(JPF, t2, 8,)			
5	(+, a, #1, t3)			
6	(:=, t3, a,)			
7	(JP, 1, ,)			
8	(=, t1, b, t4)			
9	(JPF, t4, 13)			
10	(+, b, #2, t5)			
11	(:=, t5, b,)			
12	(JP, 1, ,)			
13	(*, c, d, t6)			
14	(:=, t6, e,)			

When Executing #JPF	SS After Second #Pid	SS After First #Pid
	b	
9	4	
t4	t2	A
t1	t1	t1
1	1	1