

طراحی کامپایلرها

نیم سال دوم ۰۳-۰۲



استاد: سمانه حسینمردی

دانشکده‌ی مهندسی کامپیوتر

پاسخ‌دهنده: معین آعلی - ۴۰۱۱۰۵۵۶۱

تمرین سوم

پاسخ مسئله‌ی ۱.

```
۱ program A()  
۲   var i,j : integer  
۳   procedure B(a : real)  
۴     var b[1..10] : real  
۵     procedure C(k : integer)  
۶       var c : real  
۷       b(4) := a + c  
۸     end C  
۹   end B  
۱۰  procedure D(l : integer)  
۱۱    l := i + j  
۱۲    procedure E()  
۱۳      var d : real  
۱۴    end E  
۱۵  end D  
۱۶ end A
```

وضعیت Symbol Table و Scope Stack هنگام اجرای خط ۷ به صورت زیر است:

Scope Stack

7
4
0

Symbol Table

	lexeme	type	scope
0	A	-	1
1	;	int	1
2	j	int	1
3	B	-	1
4	a	real	2
5	b	real*	2
6	c	-	2
7	c	real	3

وضعیت Symbol Table و Scope Stack هنگام اجرای خط ۱۱ به صورت زیر است:

Scope Stack

5
0

Symbol Table

	lexeme	type	scope
0	A	-	1
1	i	int	1
2	j	int	1
3	B	-	1
4	D	-	1
5	l	int	2

پاسخ مسئله‌ی ۲.

```

۱ program A()
۲   var i : real
۳   procedure B(j : integer)
۴     var k : integer
۵     i := 0
۶     j := 6
۷     k := i + j
۸   end B
۹   procedure C(l : integer)
۱۰    var l : integer
۱۱    procedure D(k : real)
۱۲      var a[1..5] : real
۱۳      a[j] := 5
۱۴      k := 3/i
۱۵    end D
۱۶  end C
۱۷ end A

```

خط (۷)

$k := i + j \xrightarrow{\text{Static Error}} \text{Type Checking Error}$

خط (۱۰)

$\text{var } l : \text{integer} \xrightarrow{\text{Static Error}} \text{Uniqueness Checking Error}$

خط (۱۳)

$a[j] := ۵ \xrightarrow{\text{Static Error}} j \text{ not Defined in this Scope}$

خط (۱۴)

$k := ۳/i \xrightarrow{\text{Static Error}} i \text{ not Defined in this Scope}$

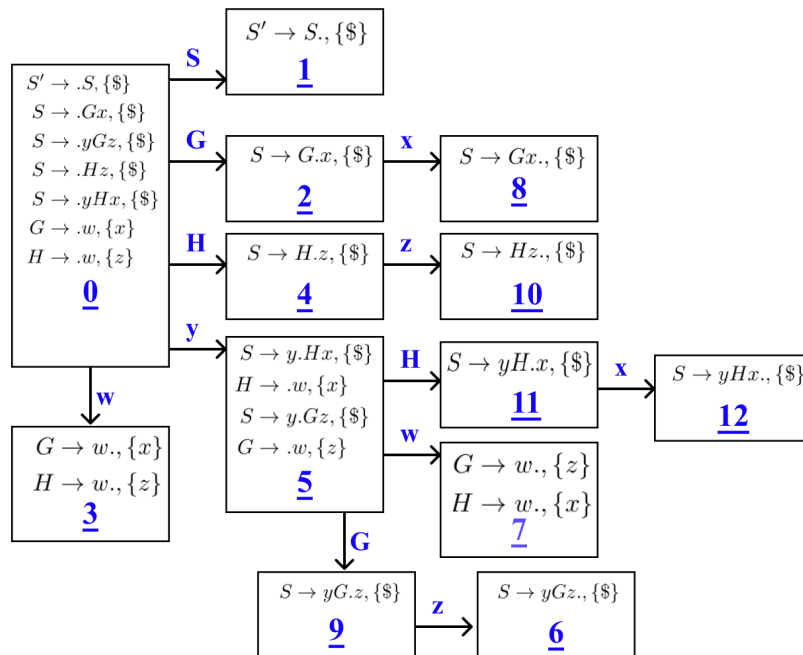
تمامی ارورهای این برنامه از نوع Static است و ارور Dynamic ندارد.

پاسخ مسئله‌ی ۳.

گرامر سوال:

$S \rightarrow Gx$
 $S \rightarrow yGz$
 $S \rightarrow Hz$
 $S \rightarrow yHx$
 $G \rightarrow w$
 $H \rightarrow w$

ترنزیشن دیاگرام مربوط به گرامر فوق:



الف

جدول $LR(1)$ مربوط به گرامر فوق:

	x	y	z	w	\$	S	G	H
0		S5		S3		1	2	8
1					acc			
2	S8							
3	R9		R6					
4			S10					
5				S7			9	11
6					R2			
7	R6		R9					
8					R1			
9			S6					
10					R3			
11	S12							
12					R8			

ب

جدول $LALR(1)$ مربوط به گرامر فوق:

	x	y	z	w	\$	S	G	H
0		S5		S3,7		1	2	4
1					acc			
2	S8							
3,7	R9/R6			R9,R6				
4			S10					
5				S3,7			9	11
6					R2			
8					R1			
9			S6					
10					R3			
11	S12							
12					R8			

ج

همانطور که مشخص است، چون جدول $LR(1)$ تداخلی ندارد ولی جدول $LALR(1)$ دارای تداخل است. پس گرامر فوق از نوع $LR(1)$ است.

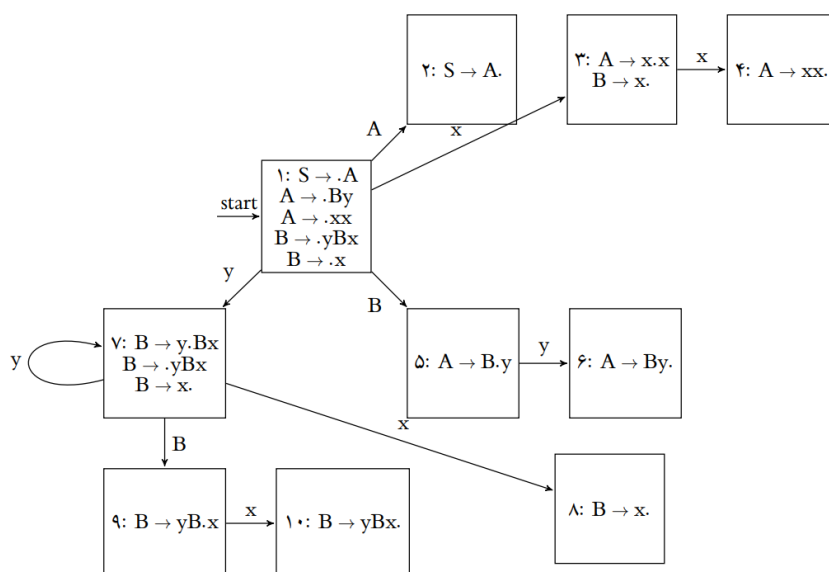
به عنوان مثال رشته ورودی $yxw\$$ را با استفاده از جدول فوق اگر پارس کنیم، داخل خانه $(x, 37)$ به تداخل می‌خوریم.

پاسخ مسئله‌ی ۴.

گرامر سوال:

$$\begin{aligned} S &\rightarrow A \\ A &\rightarrow By|xx \\ B &\rightarrow x|yBx \end{aligned}$$

ترنیزشن دیاگرام $LR(0)$ مربوط به گرامر فوق:



الف

در یکی از قواعد استتیت شماره ۳، • به انتها رسیده و در باقی قواعد می‌توان با ترمینال x حرکت کرد. چون که x عضو $follow(x)$ است پس در خانه‌ی $(3, x)$ نداخلی از نوع $shift - reduce$ داریم.

$shift\text{4}/reduce\text{4}$

پس گرامر ما نوع $SLR(1)$ نیست.

ب

مجموعه‌ی $Follow$ ها عبارتند از:

$$\begin{aligned} Follow(S) &= \{\$ \} \\ Follow(A) &= \{\$ \} \\ Follow(B) &= \{x, y\} \end{aligned}$$

ج

مجموعه $Look - Ahead$ های دیاگرام فوق:

$$\begin{array}{llll} LA_{state\gamma} = \{\$ \} & LA_{state\gamma} = \{y\} & LA_{state\gamma} = \{\$ \} & \\ LA_{state\delta} = \{\$ \} & LA_{state\gamma} = \{x, y\} & LA_{state\Lambda} = \{x\} & LA_{state\backslash\bullet} = \{x, y\} \end{array}$$

د

جدول را تشکیل می‌دهیم، اگر تداخلی وجود نداشت این گرامر $LALR(1)$ است و اگر تداخلی وجود داشت گرامر $LALR(1)$ نیست:

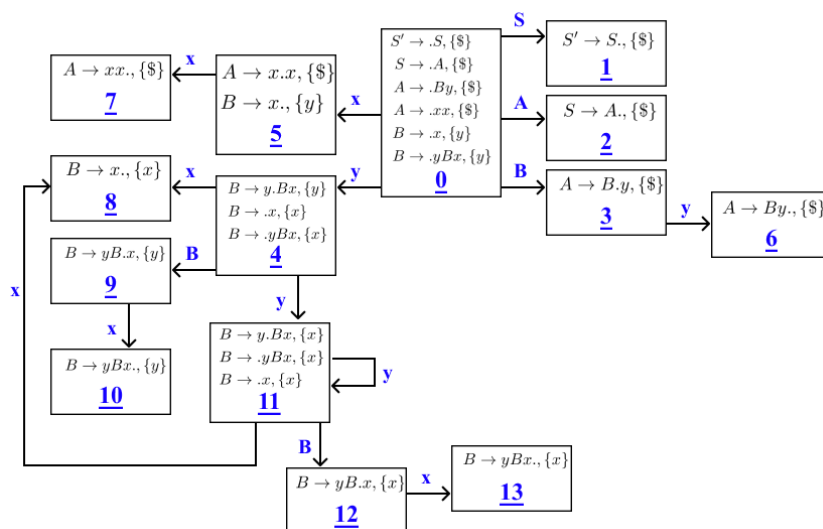
	x	y	\$	S	A	B
0	S5	S4		1	2	3
1			acc			
2			R1			
3		S6				
4,11	S8	S4,11				9,12
5	S7	R4				
6						
7						
8	R4					
9,12	S10,13					
10,13	R5	R5				

پس گرامر فول از نوع $LALR(1)$ است.

ه

با توجه به بخش قبلی، چون گرامر ما $LALR(1)$ است، پس $LA(1)$ هم است.

ترنزیشن دیاگرام $LR(1)$ مربوط به گرامر فوق:



پاسخ مسئله‌ی ۵.

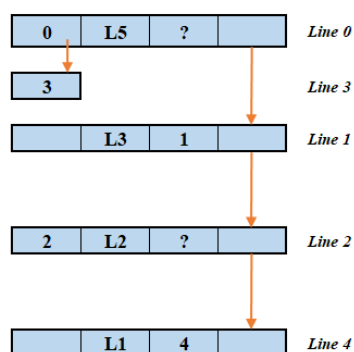
```

۱ goto L5
۲ L3: Stmt1
۳   goto L2
۴   goto L5
۵ L1: Stmt2
۶ L5: Stmt3
۷   goto L3
۸   goto L5
۹   goto L2
۱۰ L4: Stmt4
۱۱   goto L1
۱۲   goto L4
۱۳ L2: Stmt5

```

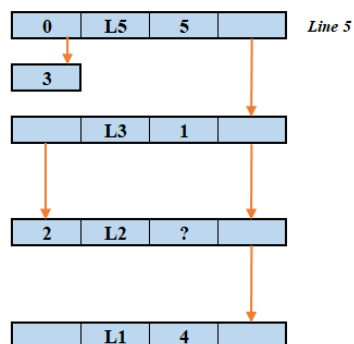
i	PB[i]
0	(jp,?, ,)
1	Stmt1
2	(jp,?, ,)
3	(jp,?, ,)
4	Stmt2
5	
6	
7	
8	
9	
10	
11	
12	

goto L5
L3: Stmt1
goto L2
goto L5
L1: Stmt2
L5: Stmt3
goto L3
goto L5
goto L2
L4: Stmt4
goto L1
goto L4
L2: Stmt5



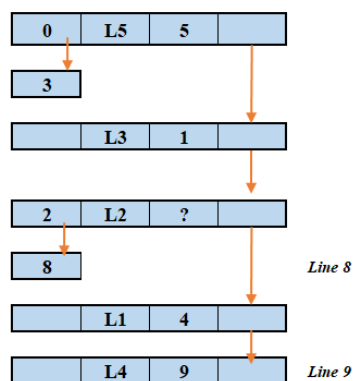
i	PB[i]
0	(jp,5, ,)
1	Stmt1
2	(jp,?, ,)
3	(jp,5, ,)
4	Stmt2
5	Stmt3
6	
7	
8	
9	
10	
11	
12	

goto L5
L3: Stmt1
goto L2
goto L5
L1: Stmt2
L5: Stmt3
goto L3
goto L5
goto L2
L4: Stmt4
goto L1
goto L4
L2: Stmt5

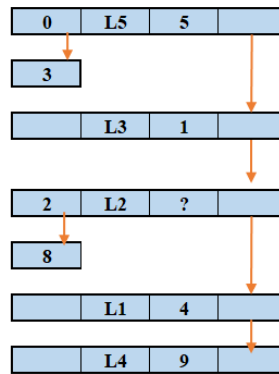


i	PB[i]
0	(jp,5, ,)
1	Stmt1
2	(jp,?, ,)
3	(jp,5, ,)
4	Stmt2
5	Stmt3
6	(jp,1, ,)
7	(jp,5, ,)
8	(jp,?, ,)
9	Stmt4
10	
11	
12	

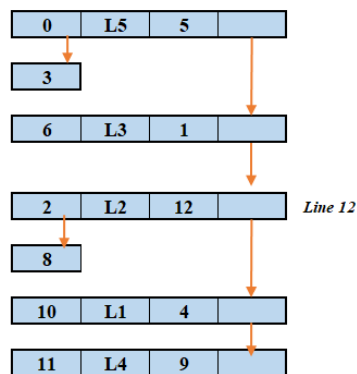
goto L5
L3: Stmt1
goto L2
goto L5
L1: Stmt2
L5: Stmt3
goto L3
goto L5
goto L2
L4: Stmt4
goto L1
goto L4
L2: Stmt5



	i	PB[i]
goto L5	0	(jp,5, ,)
L3: Stmt1	1	Stmt1
goto L2	2	(jp,?, ,)
goto L5	3	(jp,5, ,)
L1: Stmt2	4	Stmt2
L5: Stmt3	5	Stmt3
goto L3	6	(jp,1, ,)
goto L5	7	(jp,5, ,)
goto L2	8	(jp,?, ,)
L4: Stmt4	9	Stmt4
goto L1	10	(jp,4, ,)
goto L4	11	(jp,9, ,)
L2: Stmt5	12	



	i	PB[i]
goto L5	0	(jp,5, ,)
L3: Stmt1	1	Stmt1
goto L2	2	(jp,12, ,)
goto L5	3	(jp,5, ,)
L1: Stmt2	4	Stmt2
L5: Stmt3	5	Stmt3
goto L3	6	(jp,1, ,)
goto L5	7	(jp,5, ,)
goto L2	8	(jp,12, ,)
L4: Stmt4	9	Stmt4
goto L1	10	(jp,4, ,)
goto L4	11	(jp,9, ,)
L2: Stmt5	12	Stmt5



• اگر پرش به لیبل داشته باشیم که آدرس آن هنوز مشخص نیست، یک خانه خالی داخل *Liked List* ایجاد می‌کنیم.

• در هر مرحله از رسم نمودار، دستوراتی را نشان می‌دهیم که به هم ربطی ندارند و روی هم تاثیر نمی‌گذارند.

• برای دستوراتی که به یک آدرس مشخص جامپ می‌کنند کافیسیت به خانه مورد نظر آدرس را متصل کنیم.

با توجه به این که الگوریتم ما به درستی اجرا شده و تمامی مراحل بدون خطا اجرا شده‌اند، پس برنامه به پاسخ درست و نهایی خود می‌رسد.