2019 级《编译原理》课程期末试卷

语法分析（第二、三题）题解

使用网站生成

https://jsmachines.sourceforge.net/machines/

二、（26 分）二义文法 G[T]：

T → TAT|bTe|a

A → ,|;

为消除二义性而做的规定如下：

终结符","满足右结合性，终结符";"满足左结合性，且","的优先级高于";"的优先级。

（1）求非终结符 T 和 A 的 First 和 Follow 集合；

（2）写出字符串 bae,a;a 的最右推导过程，以及该字符串的句柄；

（3）构造一个 LR(0)自动机；

（4）结合上述规定构造出 SLR(1)分析表。

[题解]

（1）T 和 A 的 First 和 Follow 集合

FIRST / FOLLOW table		
Nonterminal	FIRST	FOLLOW
T'	{b,a}	{$}
T	{b,a}	{$,,,;,e}
A	{,,;}	{b,a}

（2）bae,a;a 的最右推导

无答案。

（3）LR(0)自动机

SLR closure table			
Goto	Kernel	State	Closure
	{T' -> .T}	0	{T' -> .T; T -> .T A T; T -> .b T e; T -> .a}
goto(0, T)	{T' -> T.; T -> T.A T}	1	{T' -> T.; T -> T.A T; A -> .,; A -> .;}
goto(0, b)	{T -> b.T e}	2	{T -> b.T e; T -> .T A T; T -> .b T e; T -> .a}
goto(0, a)	{T -> a.}	3	{T -> a.}
goto(1, A)	{T -> T A.T}	4	{T -> T A.T; T -> .T A T; T -> .b T e; T -> .a}
goto(1, ,)	{A -> ,.}	5	{A -> ,.}
goto(1, ;)	{A -> ;.}	6	{A -> ;.}
goto(2, T)	{T -> b T.e; T -> T.A T}	7	{T -> b T.e; T -> T.A T; A -> .,; A -> .;}
goto(2, b)	{T -> b.T e}	2	
goto(2, a)	{T -> a.}	3	
goto(4, T)	{T -> T A T.; T -> T.A T}	8	{T -> T A T.; T -> T.A T; A -> .,; A -> .;}
goto(4, b)	{T -> b.T e}	2	
goto(4, a)	{T -> a.}	3	
goto(7, e)	{T -> b T e.}	9	{T -> b T e.}
goto(7, A)	{T -> T A.T}	4	
goto(7, ,)	{A -> ,.}	5	
goto(7, ;)	{A -> ;.}	6	
goto(8, A)	{T -> T A.T}	4	
goto(8, ,)	{A -> ,.}	5	
goto(8, ;)	{A -> ;.}	6	

（4）SLR(1)分析表

LR table									
State	ACTION						GOTO		
	b	e	a	,	;	$	T'	T	A
0	s2		s3					1	
1				s5	s6	acc			4
2	s2		s3					7	
3		r_3		r_3	r_3	r_3			
4	s2		s3						8
5	r_4		r_4						
6	r_5		r_5						
7		s9		s5	s6				4
8		r_1		⊙s5 / ○r_1	○s6 / ⊙r_1	r_1			4
9		r_2		r_2	r_2	r_2			

结合性和优先级消除二义性：参考课本 4.8 节。

三、（14 分）给定文法 G[A]:A->(A)|a

（1）构造 LR(1)自动机；

LR(1) grammar ('' is ε):
```
(0) A' -> A
(1) A -> ( A )
(2) A -> a
```

	LR(1) closure table		
Goto	Kernel	State	Closure
	{[A' -> .A, $]}	0	{[A' -> .A, $]; [A -> .(A), $]; [A -> .a, $]}
goto(0, A)	{[A' -> A., $]}	1	{[A' -> A., $]}
goto(0, ()	{[A -> (.A), $]}	2	{[A -> (.A), $]; [A -> .(A),)]; [A -> .a,)]}
goto(0, a)	{[A -> a., $]}	3	{[A -> a., $]}
goto(2, A)	{[A -> (A.), $]}	4	{[A -> (A.), $]}
goto(2, ()	{[A -> (.A),)]}	5	{[A -> (.A),)]; [A -> .(A),)]; [A -> .a,)]}
goto(2, a)	{[A -> a.,)]}	6	{[A -> a.,)]}
goto(4,))	{[A -> (A)., $]}	7	{[A -> (A)., $]}
goto(5, A)	{[A -> (A.),)]}	8	{[A -> (A.),)]}
goto(5, ()	{[A -> (.A),)]}	5	
goto(5, a)	{[A -> a.,)]}	6	
goto(8,))	{[A -> (A).,)]}	9	{[A -> (A).,)]}

FIRST table

Nonterminal	FIRST
A'	{ (, a}
A	{ (, a}

LR table

| State | ACTION | | | | GOTO | |
	(	)	a	$	A'	A
0	s2		s3			1
1				acc		
2	s5		s6			4
3				r2		
4		s7				
5	s5		s6			8
6		r2				
7				r1		
8		s9				
9		r1				

Input (tokens): c d d
Maximum number of steps: 100
PARSE

| | Trace | | | Tree |
Step	Stack	Input	Action	
1	0	c d d $		

（2）合并同心项，构造 LALR(1)分析表。

LALR(1) grammar ('' is ε):
```
(0) A' -> A
(1) A -> ( A )
(2) A -> a
```

	LALR(1) closure table		
Goto	Kernel	State	Closure
	{[A' -> .A, $]}	0	{[A' -> .A, $]; [A -> .(A), $]; [A -> .a, $]}
goto(0, A)	{[A' -> A., $]}	1	{[A' -> A., $]}
goto(0, ()	{[A -> (.A), $/)]}	2	{[A -> (.A), $/)]; [A -> .(A),)]; [A -> .a,)]}
goto(0, a)	{[A -> a., $/)]}	3	{[A -> a., $/)]}
goto(2, A)	{[A -> (A.), $/)]}	4	{[A -> (A.), $/)]}
goto(2, ()	{[A -> (.A), $/)]}	2	
goto(2, a)	{[A -> a., $/)]}	3	
goto(4,))	{[A -> (A)., $/)]}	5	{[A -> (A)., $/)]}

FIRST table

Nonterminal	FIRST
A'	{ (, a}
A	{ (, a}

LR table

| State | ACTION | | | | GOTO | |
	(	)	a	$	A'	A
0	s2		s3			1
1				acc		
2	s2		s3			4
3		r2		r2		
4		s5				
5		r1		r1		

Input (tokens): c d d
Maximum number of steps: 100
PARSE

| | Trace | | | Tree |
Step	Stack	Input	Action	
1	0	c d d $		