

# 编译原理参考答案

齐王璟 周 帆

## 第四章 自顶向下语法分析

### 1. 对文法 $G[S]$

$$S \rightarrow a|\wedge|(T)$$

$$T \rightarrow T,S|S$$

- (1) 给出 $(a, (a, a))$ 和 $((a, a), \wedge, (a)), a)$ 的最左推导。
- (2) 对文法 $G$ 进行改写，然后对每个非终结符写出不带回溯的递归子程序。
- (3) 经改写后的文法是否是 $LL(1)$ 的？给出它的预测分析表。
- (4) 给出输入串 $(a, a)\#$ 的分析过程，并说明该串是否为 $G$ 的句子。

答：

- (1) 对 $(a, (a, a))$ 的最左推导为：

$$\begin{aligned} S &\Rightarrow (T) \\ &\Rightarrow (T, S) \\ &\Rightarrow (S, S) \\ &\Rightarrow (a, S) \\ &\Rightarrow (a, (T)) \\ &\Rightarrow (a, (T, S)) \\ &\Rightarrow (a, (S, S)) \\ &\Rightarrow (a, (a, S)) \\ &\Rightarrow (a, (a, a)) \end{aligned}$$

对 $((a, a), \wedge, (a)), a)$ 的最左推导为：

$$\begin{aligned} S &\Rightarrow (T) \\ &\Rightarrow (T, S) \\ &\Rightarrow (S, S) \\ &\Rightarrow ((T), S) \\ &\Rightarrow ((T, S), S) \\ &\Rightarrow ((T, S, S), S) \\ &\Rightarrow ((S, S, S), S) \\ &\Rightarrow (((T), S, S), S) \\ &\Rightarrow (((T, S), S, S), S) \\ &\Rightarrow (((S, S), S, S), S) \\ &\Rightarrow (((a, S), S, S), S) \\ &\Rightarrow (((a, a), S, S), S) \\ &\Rightarrow (((a, a), \wedge, S), S) \\ &\Rightarrow (((a, a), \wedge, (T)), S) \\ &\Rightarrow (((a, a), \wedge, (S)), S) \\ &\Rightarrow (((a, a), \wedge, (a)), S) \end{aligned}$$

$$\Rightarrow (((a, a), \wedge, (a)), a)$$

(2) 改写文法为：

$S \rightarrow a$

$S \rightarrow \wedge$

$S \rightarrow (T)$

$T \rightarrow SN$

$N \rightarrow, SN$

$N \rightarrow \varepsilon$

非终结符	FIRST集	FOLLOW集
$S$	$\{a, \wedge, ()\}$	$\{\#, ,, )\}$
$T$	$\{a, \wedge, ()\}$	$\{\}$
$N$	$\{, , \varepsilon\}$	$\{\}$

递归子程序：

```

1. Procedure S (Token) :
2.   Begin
3.     if Token ≠ 'a' and Token ≠ '∧' and Token ≠ '('
4.     then error
5.     If Token == '(' then
6.       Begin
7.         getNext(Token);
8.         T(Token);
9.         getNext(Token);
10.        if Token ≠ ')' then error
11.      End
12.   End
13.
14. Procedure T (Token) :
15.   Begin
16.     if Token == 'a' or Token == '∧' or Token == '(' then
17.       Begin
18.         S(Token);
19.         getNext(Token);
20.         N(Token);
21.       End
22.     else error
23.   End
24.
25. Procedure N (Token) :
26.   Begin
27.     if Token == ',' then
28.       Begin
29.         getNext(Token);
30.         S(Token)
31.         getNext(Token);

```

```

32.         N(Token)
33.     End
34.     else if TOKEN ≠ 'end' then error;
35. End

```

(3) 对左部为 $N$ 的产生式可知:

$SELECT(S \rightarrow a) = \{a\}$   
 $SELECT(S \rightarrow \wedge) = \{\wedge\}$   
 $SELECT(S \rightarrow (T)) = \{( \}$   
 $SELECT(T \rightarrow SN) = \{a, \wedge, ( \}$   
 $SELECT(N \rightarrow , SN) = \{, \}$   
 $SELECT(N \rightarrow \varepsilon) = \{ \}$

由于 $SELECT(N \rightarrow , SN) \cap SELECT(N \rightarrow \varepsilon) = \{ \} \cap \{ \} = \emptyset$ ,  
 $SELECT(S \rightarrow a) \cap SELECT(S \rightarrow \wedge) \cap SELECT(S \rightarrow (T)) = \{a\} \cap \{\wedge\} \cap \{( \} = \emptyset$ ,  
 所以文法是 $LL(1)$ 的。

预测分析表:

	a	$\wedge$	(	)	,	#
S	$\rightarrow a$	$\rightarrow \wedge$	$\rightarrow (T)$			
T	$\rightarrow SN$	$\rightarrow SN$	$\rightarrow SN$			
N				$\rightarrow \varepsilon$	$\rightarrow , SN$	

(4) 对输入串 $(a, a)\#$ 的分析过程:

栈	当前输入符	剩余输入符	所用产生式
#S	(	a, a)#	$S \rightarrow (T)$
#)T(	(	a, a)#	
#)T	a	, a)#	$T \rightarrow SN$
#)NS	a	, a)#	$S \rightarrow a$
#)Na	a	, a)#	
#)N	,	a)#	$N \rightarrow , SN$
#)NS,	,	a)#	
#)NS	a	)#	$S \rightarrow a$
#)Na	a	)#	
#)N	)	#	$N \rightarrow \varepsilon$
#)	)	#	
#	#		

可见输入串 $(a, a)\#$ 是文法的句子。

5. 文法 $G$ 如下:

$\langle \text{程序} \rangle \rightarrow \text{begin} \langle \text{语句表} \rangle \text{end}$   
 $\langle \text{语句表} \rangle \rightarrow \langle \text{语句} \rangle | \langle \text{语句表} \rangle ; \langle \text{语句} \rangle$   
 $\langle \text{语句} \rangle \rightarrow \langle \text{无条件语句} \rangle | \langle \text{条件语句} \rangle$   
 $\langle \text{无条件语句} \rangle \rightarrow a$

〈条件语句〉→〈如果语句〉|〈如果语句〉*else*〈语句〉

〈如果语句〉→〈如果子句〉〈无条件语句〉

〈如果子句〉→ *if b then*

试将*G*改写为LL(1)文法，并构造其预测分析表，判断改写后的文法是否为LL(1)文法。

答：

文法*G*表示为： $S \rightarrow begin\ A\ end$

$A \rightarrow B|A;B$

$B \rightarrow C|D$

$C \rightarrow a$

$D \rightarrow E|E\ else\ B$

$E \rightarrow FC$

$F \rightarrow if\ b\ then$

文法*G*改写成*G'*：

$S \rightarrow begin\ A\ end$

$A \rightarrow BA'$

$A' \rightarrow ;BA'|\epsilon$

$B \rightarrow C|D$

$C \rightarrow a$

$D \rightarrow ED'$

$D' \rightarrow else\ B|\epsilon$

$E \rightarrow FC$

$F \rightarrow if\ b\ then$

文法*G'*的first集与follow集：

非终结符	FIRST集	FOLLOW集
<i>S</i>	{ <i>begin</i> }	{#}
<i>A</i>	{ <i>a, if b then</i> }	{ <i>end</i> }
<i>A'</i>	{ <i>;, ε</i> }	{ <i>end</i> }
<i>B</i>	{ <i>a, if b then</i> }	{ <i>;, end</i> }
<i>C</i>	{ <i>a</i> }	{ <i>;, end, else</i> }
<i>D</i>	{ <i>if b then</i> }	{ <i>;, end</i> }
<i>D'</i>	{ <i>else, ε</i> }	{ <i>;, end</i> }
<i>E</i>	{ <i>if b then</i> }	{ <i>;, end, else</i> }
<i>F</i>	{ <i>if b then</i> }	{ <i>a</i> }

文法*G'*的SELECT集：

	产生式	SELECT
<i>S</i>	$S \rightarrow begin\ A\ end$	{ <i>begin</i> }
<i>A</i>	$A \rightarrow BA'$	{ <i>a, if b then</i> }
<i>A'</i>	$A' \rightarrow ;BA'$	{ <i>;</i> }
	$A' \rightarrow \epsilon$	{ <i>end</i> }
<i>B</i>	$B \rightarrow C$	{ <i>a</i> }
	$B \rightarrow D$	{ <i>if b then</i> }

$C$	$C \rightarrow a$	$\{a\}$
$D$	$D \rightarrow ED'$	$\{if\ b\ then\}$
$D'$	$D' \rightarrow else\ B$	$\{else\}$
	$D' \rightarrow \varepsilon$	$\{;, end\}$
$E$	$E \rightarrow FC$	$\{if\ b\ then\}$
$F$	$F \rightarrow if\ b\ then$	$\{if\ b\ then\}$

文法  $G'$  的预测分析表：

	$begin$	$;$	$a$	$else$	$if\ b\ then$	$end$
$S$	$\rightarrow begin\ A\ end$					
$A$			$\rightarrow BA'$		$\rightarrow BA'$	
$A'$		$\rightarrow; BA'$				$\rightarrow \varepsilon$
$B$			$\rightarrow C$		$\rightarrow D$	
$C$			$\rightarrow a$			
$D$					$\rightarrow ED'$	
$D'$		$\rightarrow \varepsilon$		$\rightarrow else\ B$		$\rightarrow \varepsilon$
$E$					$\rightarrow FC$	
$F$					$\rightarrow if\ b\ then$	

由于  $SELECT(B \rightarrow C) \cap SELECT(B \rightarrow D) = \{a\} \cap \{if\ b\ then\} = \emptyset$ ,  
 $SELECT(A' \rightarrow; BA') \cap SELECT(A' \rightarrow \varepsilon) = \{;\} \cap \{end\} = \emptyset$ ,  
 $SELECT(D' \rightarrow else\ B) \cap SELECT(D' \rightarrow \varepsilon) = \{else\} \cap \{;, end\} = \emptyset$ ,  
 改写后的文法是  $LL(1)$  文法。

6. 判断下面哪些文法是  $LL(1)$  的，哪些能被改写成  $LL(1)$  文法，并对每个  $LL(1)$  文法设计相应的递归下降识别器。

答：

(1)  $S \rightarrow A|B$   
 $A \rightarrow aA|a$   
 $B \rightarrow bB|b$

答：

该文法有左公因子，因此不是  $LL(1)$  文法。

改写为：

$S \rightarrow A|B$   
 $A \rightarrow aA'$   
 $A' \rightarrow A|\varepsilon$   
 $B \rightarrow bB'$   
 $B' \rightarrow B|\varepsilon$

产生式	$SELECT$
$S \rightarrow A$	$\{a\}$
$S \rightarrow B$	$\{b\}$
$A \rightarrow aA'$	$\{a\}$
$A' \rightarrow A$	$\{a\}$

$A' \rightarrow \varepsilon$	$\{\#\}$
$B \rightarrow bB'$	$\{b\}$
$B' \rightarrow bB'$	$\{b\}$
$B' \rightarrow \varepsilon$	$\{\#\}$

$SELECT(S \rightarrow A) \cap SELECT(S \rightarrow B) = \emptyset$ ;  $SELECT(A' \rightarrow A) \cap SELECT(A' \rightarrow \varepsilon) = \emptyset$ ;  $SELECT(B' \rightarrow B) \cap SELECT(B' \rightarrow \varepsilon) = \emptyset$  所以改写后的文法是LL(1)文法。

递归下降识别器：

```

1. void S(){
2.     if(sym = 'a')
3.         A();
4.     else if(sym = 'b')
5.         B();
6.     else error();
7. }
8. void A(){
9.     if(sym = 'a') {
10.        advance;
11.        A'(); }
12.    else error();
13. }
14. void B(){
15.     if(sym = 'b') {
16.        advance;
17.        B'(); }
18.    else error();
19. }
20. void A'(){
21.     if(sym = 'a')
22.        A();
23.    else if(sym = '#'){ }
24.    else error();
25. }
26. void B'(){
27.     if(sym = 'b')
28.        B();
29.    else if(sym = '#'){ }
30.    else error();
31. }

```

(3)  $S \rightarrow aAaB|bAbB$

$A \rightarrow S|db$

$B \rightarrow bB|a$

答:

该文法各个产生式的可选集为:

产生式	<i>SELECT</i>
$S \rightarrow aAaB$	$\{a\}$
$S \rightarrow bAbB$	$\{b\}$
$A \rightarrow S$	$\{a, b\}$
$A \rightarrow db$	$\{d\}$
$B \rightarrow bB$	$\{b\}$
$B \rightarrow a$	$\{a\}$

$SELECT(S \rightarrow aAaB) \cap SELECT(S \rightarrow bAbB) = \{a\} \cap \{b\} = \emptyset$  ;

$SELECT(A \rightarrow S) \cap SELECT(A \rightarrow db) = \{a, b\} \cap \{d\} = \emptyset$ ;

$SELECT(B \rightarrow bB) \cap SELECT(B \rightarrow a) = \{b\} \cap \{a\} = \emptyset$ 。

所以该文法为LL(1)文法。

递归下降识别器:

```
3. void S(){
4.     if(sym == 'a'){
5.         advance;
6.         A();
7.         if(sym != 'a') error();
8.         advance;
9.         B();
10.    }
11.    else if (sym == 'b'){
12.        advance;
13.        A();
14.        if(sym != 'b') error();
15.        advance;
16.        B();
17.    }
18.    else error();
19. }
20.
21. void A(){
22.     if(sym == 'a' || sym == 'b'){
23.         S();
24.     }
25.     else if(sym == 'd'){
26.         advance;
27.         if(sym != 'b') error();
28.         advance;
29.     }
```

```

30.     else error();
31. }
32.
33. void B(){
34.     if(sym = 'b'){
35.         advance;
36.         B();
37.     }
38.     else if (sym = 'a'){
39.         advance;
40.     }
41.     else error();
42. }

```

(5)  $S \rightarrow SaA|bB$

$A \rightarrow aB|c$

$B \rightarrow Bb|d$

答：该文法含有左递归，因此不是LL(1)的。

改写为：

$S \rightarrow bBS'$

$S' \rightarrow aAS'|\epsilon$

$A \rightarrow aB|c$

$B \rightarrow dB'$

$B' \rightarrow bB'|\epsilon$

产生式	SELECT
$S \rightarrow bBS'$	{b}
$S' \rightarrow aAS'$	{a}
$S' \rightarrow \epsilon$	{#}
$A \rightarrow aB$	{a}
$A \rightarrow c$	{c}
$B \rightarrow dB'$	{d}
$B' \rightarrow bB'$	{b}
$B' \rightarrow \epsilon$	{a, #}

$SELECT(S' \rightarrow aAS') \cap SELECT(S' \rightarrow \epsilon) = \{a\} \cap \{\#\} = \emptyset;$

$SELECT(A \rightarrow aB) \cap SELECT(A \rightarrow c) = \{a\} \cap \{c\} = \emptyset;$

$SELECT(B' \rightarrow bB') \cap SELECT(B' \rightarrow \epsilon) = \{b\} \cap \{a, \#\} = \emptyset。$

所以改写后的文法是LL(1)文法。

递归下降识别器：

```

5. void S(){
6.     if(sym = 'b'){
7.         advance;
8.         B();
9.         S'();

```



```
10.     }
11.     else error();
12. }
13.
14. void A(){
15.     if(sym = 'a'){
16.         advance;
17.         B();
18.     }
19.     else if(sym = 'c'){
20.         advance;
21.     }
22.     else error();
23. }
24.
25. void B(){
26.     if(sym = 'd'){
27.         advance;
28.         B'();
29.     }
30.     else error();
31. }
32.
33. void S'(){
34.     if(sym = 'a'){
35.         advance;
36.         A();
37.         S'();
38.     }
39.     else if(sym = '#'){ }
40.         else error();
41. }
42.
43. void B'(){
44.     if(sym = 'b'){
45.         advance;
46.         B'();
47.     }
48.     else if(sym = '#' || sym = 'a'){ }
49.         else error();
50.
51.
52. }
```

7. 对于一个文法若消除了左递归, 提取了左公共因子后是否一定为LL(1)文法?  
试对下面的文法进行改写, 并对改写后的文法进行判断。

- (1)  $A \rightarrow baB|\varepsilon$   
 $B \rightarrow Abb|a$   
 (2)  $A \rightarrow aABe|a$   
 $B \rightarrow Bb|d$   
 (3)  $S \rightarrow Aa|b$   
 $A \rightarrow SB$   
 $B \rightarrow ab$

答:

- (1)  $A \rightarrow baB|\varepsilon$   
 $B \rightarrow Abb|a$   
 先改写文法为:  
 $A \rightarrow baB$   
 $A \rightarrow \varepsilon$   
 $B \rightarrow baBbb$   
 $B \rightarrow bb$   
 $B \rightarrow a$   
 再改写文法为:  
 $A \rightarrow baB$   
 $A \rightarrow \varepsilon$   
 $B \rightarrow bB'$   
 $B \rightarrow a$   
 $B' \rightarrow aBbb$

$$B' \rightarrow b$$

(1) 计算这个文法的每个非终结符的FIRST集和FOLLOW集:

非终结符	FIRST集	FOLLOW集
$A$	$\{b\}$	$\{\#\}$
$B$	$\{b, a\}$	$\{\#, b\}$
$B'$	$\{b, a\}$	$\{\#, b\}$

(2) 每个产生式的可选集:

产生式	SELECT
$A \rightarrow baB$	$\{b\}$
$A \rightarrow \varepsilon$	$\{\#\}$
$B \rightarrow bB'$	$\{b\}$
$B \rightarrow a$	$\{a\}$
$B' \rightarrow aBbb$	$\{a\}$
$B' \rightarrow b$	$\{b\}$

相同左部的产生式的可选集的交集为空集

$$SELECT(A \rightarrow baB) \cap SELECT(A \rightarrow \varepsilon) = \{b\} \cap \{\#\} = \emptyset$$

$$SELECT(B \rightarrow bB') \cap SELECT(B \rightarrow a) = \{b\} \cap \{a\} = \emptyset$$

$$SELECT(B' \rightarrow aBbb) \cap SELECT(B' \rightarrow b) = \{a\} \cap \{b\} = \emptyset$$

所以该文法是LL(1)文法。

(3)构造它的预测分析表：

	<i>a</i>	<i>b</i>	#
<i>A</i>		$\rightarrow baB$	$\rightarrow \varepsilon$
<i>B</i>	$\rightarrow a$	$\rightarrow bB'$	
<i>B'</i>	$\rightarrow aBbb$	$\rightarrow b$	

(2)  $A \rightarrow aABe|a$

$B \rightarrow Bb|d$

答：

提取左公共因子和消除左递归后的文法变为：

$A \rightarrow aA'$

$A' \rightarrow ABe$

$A' \rightarrow \varepsilon$

$B \rightarrow dB'$

$B' \rightarrow bB'$

$B' \rightarrow \varepsilon$

(1)计算这个文法的每个非终结符的FIRST集和FOLLOW集：

非终结符	FIRST集	FOLLOW集
<i>A</i>	{ <i>a</i> }	{#, <i>d</i> }
<i>B</i>	{ <i>d</i> }	{ <i>e</i> }
<i>A'</i>	{ <i>a</i> , $\varepsilon$ }	{#, <i>d</i> }
<i>B'</i>	{ <i>b</i> , $\varepsilon$ }	{ <i>e</i> }

(2)每个产生式的可选集：

产生式	SELECT
$A \rightarrow aA'$	{ <i>a</i> }
$A' \rightarrow ABe$	{ <i>a</i> }
$A' \rightarrow \varepsilon$	{#, <i>d</i> }
$B \rightarrow dB'$	{ <i>d</i> }
$B' \rightarrow bB'$	{ <i>b</i> }
$B' \rightarrow \varepsilon$	{ <i>e</i> }

相同左部的产生式的可选集的交集为空集

$SELECT(A' \rightarrow ABe) \cap SELECT(A' \rightarrow \varepsilon) = \{a\} \cap \{\#, d\} = \emptyset$

$SELECT(B' \rightarrow bB') \cap SELECT(B' \rightarrow \varepsilon) = \{b\} \cap \{e\} = \emptyset$

所以该文法是LL(1)文法。

(3)构造它的预测分析表：

	<i>a</i>	<i>e</i>	<i>b</i>	<i>d</i>	#
<i>A</i>	$\rightarrow aA'$				
<i>B</i>				$\rightarrow dB'$	

$B'$		$\rightarrow \varepsilon$	$\rightarrow bB'$		
$A'$	$\rightarrow ABe$			$\rightarrow \varepsilon$	$\rightarrow \varepsilon$

(3)  $S \rightarrow Aa|b$

$A \rightarrow SB$

$B \rightarrow ab$

答:

用 A 的产生式右部代替 S 的产生式右部的 A 得:

$S \rightarrow SBa|b$

$B \rightarrow ab$

消除左递归后文法变为:

$S \rightarrow bS'$

$S' \rightarrow BaS'$

$S' \rightarrow \varepsilon$

$B \rightarrow ab$

(1)计算这个文法的每个非终结符的FIRST集和FOLLOW集:

非终结符	FIRST集	FOLLOW集
$S$	$\{b\}$	$\{\#\}$
$B$	$\{a\}$	$\{a\}$
$S'$	$\{\varepsilon, a\}$	$\{\#\}$

(2)每个产生式的可选集:

产生式	SELECT
$S \rightarrow bS'$	$\{b\}$
$S' \rightarrow BaS'$	$\{a\}$
$S' \rightarrow \varepsilon$	$\{\#\}$
$B \rightarrow ab$	$\{a\}$

相同左部的产生式的可选集的交集为空集

$SELECT(S' \rightarrow BaS') \cap SELECT(S' \rightarrow \varepsilon) = \{a\} \cap \{\#\} = \emptyset$

所以该文法是LL(1)文法

(3)构造它的预测分析表:

	$a$	$b$	$\#$
$S$		$\rightarrow bS'$	
$B$	$\rightarrow ab$		
$S'$	$\rightarrow BaS'$		$\rightarrow \varepsilon$