

(若发现问题, 请及时告知)

1 设有文法  $G[S]$ :

$$S \rightarrow aSb \mid aab$$

若针对该文法设计一个自顶向下预测分析过程, 则需要向前察看多少个输入符号?

参考解答:

需要向前察看 3 个单词。若向前察看 3 个单词是 aab 时, 可选第 2 个分支; aaa 时, 可选第 1 个分支。

3 设有文法  $G[E]$ :

$$\begin{aligned} E &\rightarrow TA \\ A &\rightarrow \vee TA \mid \varepsilon \\ T &\rightarrow FB \\ B &\rightarrow \wedge FB \mid \varepsilon \\ F &\rightarrow (E) \mid i \end{aligned}$$

试针对该文法填写下表

$G$ 中的规则 $r$	$first(rhs(r))$	$follow(lhs(r))$	$PS(r)$
$E \rightarrow TA$			
$A \rightarrow \vee TA$			
$A \rightarrow \varepsilon$		此处不填	
$T \rightarrow FB$			
$B \rightarrow \wedge FB$			
$B \rightarrow \varepsilon$		此处不填	
$F \rightarrow (E)$			
$F \rightarrow i$		此处不填	

其中,  $rhs(r)$  表示产生式  $r$  的右部,  $lhs(r)$  表示产生式  $r$  的左部

参考解答: 如下表。

$G$ 中的规则 $r$	$first(rhs(r))$	$follow(lhs(r))$	$PS(r)$
$E \rightarrow TA$	$i, ($	$), \#$	$i, ($

$A \rightarrow \vee TA$	$\vee$	$), \#$	$\vee$
$A \rightarrow \varepsilon$	$\varepsilon$	此处不填	$), \#$
$T \rightarrow FB$	$i, ($	$\vee, ), \#$	$i, ($
$B \rightarrow \wedge FB$	$\wedge$	$\vee, ), \#$	$\wedge$
$B \rightarrow \varepsilon$	$\varepsilon$	此处不填	$\vee, ), \#$
$F \rightarrow (E)$	$($	$\wedge, \vee, ), \#$	$($
$F \rightarrow i$	$i$	此处不填	$i$

4 试验证下列文法  $G[S]$  是 LL(1) 文法:

$$S \rightarrow P \mid \varepsilon$$

$$P \rightarrow (P)P \mid a$$

其中  $(, ),$  以及  $a$  为终结符

参考解答:

先对于每条语法规则  $r$ , 计算  $PS(r)$ 。

$G$ 中的规则 $r$	$First(rhs(r))$	$Follow(lhs(r))$	$PS(r)$
$S \rightarrow P$	$a, ($	$\#$	$a, ($
$S \rightarrow \varepsilon$	$\varepsilon$		$\#$
$P \rightarrow (P)P$	$($	$), \#$	$($
$P \rightarrow a$	$a$		$a$

因为  $PS(S \rightarrow P) \cap PS(S \rightarrow \varepsilon) = \emptyset$  及  $PS(P \rightarrow (P)P) \cap PS(P \rightarrow a) = \emptyset$ , 所以文法  $G[S]$  是 LL(1) 文法。

5 计算下列文法中每个非终结符的 First 集和 Follow 集, 以及每个产生式的预测集合, 并判断该文法是否 LL(1)文法 (说明原因):

(1) 文法  $G_1[S]$ :

$$S \rightarrow (S) \mid aT$$

$$T \rightarrow +ST \mid \varepsilon$$

参考解答:

(1) 计算非终结符的 FIRST 集和 FOLLOW 集, 结果如下:

$$FIRST(S) = \{ (, a \}$$

$$FOLLOW(S) = \{ \#, \}, + \}$$

$$FIRST(T) = \{ +, \varepsilon \}$$

$$FOLLOW(T) = \{ \#, \}, + \}$$

$$PS(S \rightarrow (S)) = \{ ( \}$$

$$PS(S \rightarrow aT) = \{ a \}$$

$$PS(T \rightarrow +ST) = \{ + \}$$

$$PS(T \rightarrow \varepsilon) = \{ \#, \}, + \}$$

因为,  $PS(T \rightarrow +ST) \cap PS(T \rightarrow \varepsilon) = \{ + \} \cap \{ \#, \}, + \} \neq \Phi$ , 所以,  $G(S)$ 不是 LL(1)文法。

6 验证如下文法是 LL(1)文法, 并基于该文法构造递归下降分析程序:

(1) 文法  $G[S]$ :

$$S \rightarrow AB$$

$$A \rightarrow aA$$

$$A \rightarrow \varepsilon$$

$$B \rightarrow bB$$

$$B \rightarrow \varepsilon$$

**参考解答:**

(1) 计算非终结符的 FIRST 集和 FOLLOW 集, 结果如下:

$$FIRST(S) = \{a, b\} \quad FOLLOW(S) = \{ \# \}$$

$$FIRST(A) = \{a, \varepsilon\} \quad FOLLOW(A) = \{b, \#\}$$

$$FIRST(B) = \{b, \varepsilon\} \quad FOLLOW(B) = \{ \# \}$$

因为

$$FIRST(aA) \cap FOLLOW(A) = \{a\} \cap \{b, \#\} = \Phi$$

$$FIRST(bB) \cap FOLLOW(B) = \{b\} \cap \{ \# \} = \Phi$$

或

$$PS(A \rightarrow aA) \cap PS(A \rightarrow \varepsilon) = \{a\} \cap \{b, \#\} = \Phi$$

$$PS(B \rightarrow bB) \cap PS(B \rightarrow \varepsilon) = \{b\} \cap \{ \# \} = \Phi$$

所以,  $G(S)$ 是 LL(1)文法。

用类似C语言写出  $G[E]$ 的递归子程序, 其中  $yylex()$  为取下一单词过程, 变量 `lookahead` 存放当前单词。不需要考虑太多编程语言相关的细节。程序如下:

```
void ParseS( )                // 主函数
{
    ParseA( );
    ParseB( );
}
void ParseA( )
{
    switch (lookahead)        // lookahead 为下一个输入符号
    {
        case ' a' :
```

```

        MatchToken( 'a' );
        ParseA();
        break;
    case ' b' , ' #' :
        break;
    default:
        printf("syntax error \n");
        exit(0);
    }
    return A_num;
}
void ParseB( )
{
    switch (lookahead) {
        case ' b' :
            MatchToken( 'b' );
            ParseB( );
            break;
        case ' #' :
            break;
        default:
            printf("syntax error \n");
            exit(0);
    }
}
void Match_Token(int expected)
{
    if (lookahead != expected)
    {
        printf("syntax error \n");
        exit(0);
    }
    else
        lookahead = getToken();
}

```

8 给出习题 6 中所有文法的 LL(1)分析表。

**参考解答:**

(1) 文法  $G[S]$ :

$S \rightarrow AB$

$A \rightarrow aA$

$A \rightarrow \varepsilon$

$B \rightarrow bB$

$$B \rightarrow \varepsilon$$

的 LL(1)分析表为:

	a	b	#
S	$S \rightarrow AB$	$S \rightarrow AB$	$S \rightarrow AB$
A	$A \rightarrow aA$	$A \rightarrow \varepsilon$	$A \rightarrow \varepsilon$
B		$B \rightarrow bB$	$B \rightarrow \varepsilon$

9 基于图 1 中的 LL (1) 分析表, 根据图 2 描述的方法, 给出输入符号串 *baacbd* 的表驱动 LL (1) 分析过程。

参考解答:

步骤	下推栈	余留符号串	下一步动作
1	#S	baacbd#	应用产生式 $S \rightarrow BbS$
2	# SbB	baacbd#	应用产生式 $B \rightarrow \varepsilon$
3	# Sb	baacbd#	匹配栈顶和当前输入符号
4	# S	aacbd#	应用产生式 $S \rightarrow AaS$
5	# SaA	aacbd#	应用产生式 $A \rightarrow a$
6	# Saa	aacbd#	匹配栈顶和当前输入符号
7	#Sa	acbd#	匹配栈顶和当前输入符号
8	#S	cbd#	应用产生式 $S \rightarrow BbS$
9	# SbB	cbd#	应用产生式 $B \rightarrow c$
10	# Sbc	cbd#	匹配栈顶和当前输入符号
11	# Sb	bd#	匹配栈顶和当前输入符号
12	# S	d#	应用产生式 $S \rightarrow d$
13	# d	d#	匹配栈顶和当前输入符号
14	#	#	返回, 分析成功

11 按照本讲介绍的消除一般左递归算法消除下面文法  $G[S]$  中的左递归（要求依非终结符的排序  $S$ 、 $Q$ 、 $P$  执行该算法）：

$$\begin{aligned} S &\rightarrow PQ \mid a \\ P &\rightarrow QS \mid b \\ Q &\rightarrow SP \mid c \end{aligned}$$

参考解答:

按照非终结符的特定顺序排列各规则:

$$\begin{aligned} S &\rightarrow PQ \mid a \\ Q &\rightarrow SP \mid c \\ P &\rightarrow QS \mid b \end{aligned}$$

第一步, 得:

$$\begin{aligned} S &\rightarrow PQ \mid a \\ Q &\rightarrow PQP \mid aP \mid c \\ P &\rightarrow QS \mid b \end{aligned}$$

第二步, 得:

$$\begin{aligned} S &\rightarrow PQ \mid a \\ Q &\rightarrow PQP \mid aP \mid c \\ P &\rightarrow PQPS \mid aPS \mid cS \mid b \end{aligned}$$

消去  $P \rightarrow PQPS$  的左递归得:

$$\begin{aligned} S &\rightarrow PQ \mid a \\ Q &\rightarrow PQP \mid aP \mid c \\ P &\rightarrow aPSR \mid cSR \mid bR \\ R &\rightarrow QPSR \mid \epsilon \end{aligned}$$

经检查, 此时得到的文法已经不含左递归, 可结束消除左递归过程。

13 变换下列文法, 求出与其等价的一个文法, 使变换后的文法不含左递归和左公因子:

(2) 文法  $G'[S]$ :

$$\begin{aligned} S &\rightarrow aSAc \mid a \\ A &\rightarrow Ab \mid d \end{aligned}$$

参考解答:

(2) 提取左公共因子和消除左递归后,  $G[S]$  变换为等价的  $G'[S]$  如下:

$$\begin{aligned} S &\rightarrow aB \\ B &\rightarrow SA \epsilon | \epsilon \\ A &\rightarrow dC \end{aligned}$$

$$C \rightarrow bC \mid \varepsilon$$