

(若发现问题, 请及时告知)

1 设有文法 $G[S]$:

$$S \rightarrow a S b \mid a a b$$

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

这个文法不是 $LL(1)$ 文法。然而, 要构造自顶向下预测分析过程, 只需要在 $LL(1)$ 分析方法的基础上进行简单的扩展即可。你可以想想如何去做?

2 设有文法 $G[E]$:

$$\begin{aligned} S &\rightarrow i S A \\ A &\rightarrow e S \mid \varepsilon \\ S &\rightarrow a \end{aligned}$$

针对文法 $G'[S]$, 试给出各产生式右部文法符号串的 $First$ 集合, 各产生式左部非终结符的 $Follow$ 集合, 以及各产生式的预测集合 PS , 即完成下表:

G 中的规则 r	$First(rhs(r))$	$Follow(lhs(r))$	$PS(r)$
$S \rightarrow i S A$			
$A \rightarrow e S$			
$A \rightarrow \varepsilon$		此处不填	
$S \rightarrow a$		此处不填	

表中的 $rhs(r)$ 表示产生式 r 右部的文法符号串, $lhs(r)$ 表示产生式 r 左部的非终结符。

参考解答:

最终计算结果如下表所示:

G 中的规则 r	$First(rhs(r))$	$Follow(lhs(r))$	$PS(r)$
$S \rightarrow i S A$	i	$e, \#$	i
$A \rightarrow e S$	e	$e, \#$	e
$A \rightarrow \varepsilon$	ε	此处不填	$e, \#$
$S \rightarrow a$	a	此处不填	a

表中的 $rhs(r)$ 表示产生式 r 右部的文法符号串, $lhs(r)$ 表示产生式 r 左部的非终结符。

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	$First(rhs(r))$	$Follow(lhs(r))$	$PS(r)$
$S \rightarrow P$	$a, ($	$\#$	$a, ($
$S \rightarrow \varepsilon$	ε		$\#$
$P \rightarrow (P)P$	$($	$), \#$	$($
$P \rightarrow a$	a		a

表中的 $rhs(r)$ 表示产生式 r 右部的文法符号串, $lhs(r)$ 表示产生式 r 左部的非终结符。

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):

(3) 文法 $G_2[S]$:

$$S \rightarrow A1B$$

$$A \rightarrow 0A \mid \varepsilon$$

$$B \rightarrow 0B \mid 1B \mid \varepsilon$$

参考解答:

G 中的规则 r	$First(rhs(r))$	$Follow(lhs(r))$	$PS(r)$
$S \rightarrow A1B$	$0, 1$	$\#$	$0, 1$
$A \rightarrow 0A$	0	1	0
$A \rightarrow \varepsilon$	ε	此处不填	1
$B \rightarrow 0B$	0	$\#$	0
$B \rightarrow 1B$	1	此处不填	1
$B \rightarrow \varepsilon$	ε	此处不填	$\#$

表中的 $rhs(r)$ 表示产生式 r 右部的文法符号串, $lhs(r)$ 表示产生式 r 左部的非终结符。

因为 $PS(A \rightarrow 0A) \cap PS(A \rightarrow \varepsilon) = \emptyset$ 及 $PS(B \rightarrow 0B)$ 、 $PS(B \rightarrow 1B)$ 和 $PS(B \rightarrow \varepsilon)$ 两两互不相交，所以文法 $G[S]$ 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：

(2) 文法 $G[E]$:

$E \rightarrow [F]A$

$A \rightarrow E \mid \varepsilon$

$F \rightarrow aB$

$B \rightarrow aB \mid \varepsilon$

其中 $[$, $]$, 以及 a 为终结符

参考解答:

(2) 观察文法规则可知，可能产生规则选择冲突的规则只能是 $A \rightarrow E \mid \varepsilon$ 和 $B \rightarrow aB \mid \varepsilon$ 。我们只要求出这四条规则的 PS 集合（预测集合）即可。欲求这四个 PS 集合，需要先求出（过程略）：

$first(E) = \{ [\}$

$follow(A) = \{ \# \}$

$first(aB) = \{ a \}$

$follow(B) = \{] \}$

从而

$PS(A \rightarrow E) = first(E) = \{ [\}$

$PS(A \rightarrow \varepsilon) = follow(A) = \{ \# \}$

$PS(B \rightarrow aB) = first(aB) = \{ a \}$

$PS(B \rightarrow \varepsilon) = follow(B) = \{] \}$

因为

对于 $A \rightarrow E \mid \varepsilon$ 有: $PS(A \rightarrow E) \cap PS(A \rightarrow \varepsilon) = \emptyset$

对于 $B \rightarrow aB \mid \varepsilon$ 有: $PS(B \rightarrow aB) \cap PS(B \rightarrow \varepsilon) = \emptyset$

所以，文法 $G[E]$ 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。实际答题时不需要考虑太多编程语言相关的细节。*/
TOKEN lookahead;

```
VOID Parse_E( ) {
    Match_Token ( [ );
    Parse_F( );
    Match_Token ( ] );
    Parse_A( );
}
```

```
VOID Parse_A( ) {
    SWITCH (lookahead) {
        CASE [:
```

```

        Parse_E( );
        BREAK;
    CASE #:
        BREAK;
    DEFAULT:
        Syntax_Error( );
    }
}

VOID Parse_F( ) {
    Match_Token ( a );
    Parse_B ( );
}

VOID Parse_B( ) {
    SWITCH (lookahead) {
        CASE a:
            Match_Token ( a );
            Parse_B ( );
            BREAK;
        CASE ]:
            BREAK;
        CASE:
            Syntax_Error( );
    }
}

VOID Match_Token(TOKEN expected) {
    //判别当前单词是否与期望的终结符匹配
    IF (lookahead != expected)
        Syntax_Error( );
    ELSE // 若匹配,消费掉当前单词并调用词法分析器读入下一个单词
        lookahead = yylex();
}

VOID Syntax_Error() {
    PRINT ( "syntax error" );
    EXIT ( 1 );
}

```

8 给出习题 6 中所有文法的 LL(1)分析表。

参考解答:

(2) 文法 $G[S]$:

$$E \rightarrow [F]A$$

$$A \rightarrow E \mid \varepsilon$$

$$F \rightarrow aB$$

$$B \rightarrow aB \mid \varepsilon$$

其中 $[$, $]$, 以及 a 为终结符

的 LL(1)分析表为:

	a	[]	#
E		$E \rightarrow [F] A$		
A		$A \rightarrow E$		$A \rightarrow \varepsilon$
F	$F \rightarrow a B$			
B	$B \rightarrow a B$		$B \rightarrow \varepsilon$	

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。