

## 第 5 章练习 P<sub>99</sub>

作业布置：P<sub>100</sub>

2 ( 1 )( 2 )( 3 )

3 ( 补充：分析符号串 bef 是否为文法的句子 )

7 ( 1 )( 2 )

2. 对下面的文法

$G[E]: E \rightarrow TE'$

$E' \rightarrow +E | \varepsilon$

$T \rightarrow FT'$

$T' \rightarrow T | \varepsilon$

$F \rightarrow PF'$

$F' \rightarrow *F' | \varepsilon$

$P \rightarrow (E) | a | b | ^$

(1) 计算这个文法的每个非终结符的 FIRST 集和 FOLLOW 集。

(2) 证明这个方法是 LL(1) 的。

(3) 构造它的预测分析表。

(4) 构造它的递归下降分析程序。

解：

(1) 计算这个文法的每个非终结符的 FIRST 集和 FOLLOW 集。

FIRST 集合有：

$FIRST(E) = FIRST(T) = FIRST(F) = FIRST(P) = \{ (, a, b, ^ \}$ ;

$$\text{FIRST}(E') = \{+, \varepsilon\}$$

$$\text{FIRST}(T) = \text{FIRST}(F) = \text{FIRST}(P) = \{ (, a, b, ^ \}$$

$$\text{FIRST}(T') = \text{FIRST}(T) \cup \{ \varepsilon \} = \{ (, a, b, ^, \varepsilon \}$$

$$\text{FIRST}(F) = \text{FIRST}(P) = \{ (, a, b, ^ \}$$

$$\text{FIRST}(F') = \text{FIRST}(P) = \{ *, \varepsilon \}$$

$$\text{FIRST}(P) = \{ (, a, b, ^ \}$$

FOLLOW 集合有：

$$\text{FOLLOW}(E) = \{ ), \# \}$$

$$\text{FOLLOW}(E') = \text{FOLLOW}(E) = \{ ), \# \}$$

$$\text{FOLLOW}(T) = \text{FIRST}(E') \cup \text{FOLLOW}(E) = \{ +, ), \# \}; // \text{不包含 } \varepsilon$$

$$\text{FOLLOW}(T') = \text{FOLLOW}(T) = \text{FIRST}(E') \cup \text{FOLLOW}(E) = \{ +, ), \# \};$$

$$\text{FOLLOW}(F) = \text{FIRST}(T') \cup \text{FOLLOW}(T) = \{ (, a, b, ^, +, ), \# \}; // \text{不包含 } \varepsilon$$

$$\text{FOLLOW}(F') = \text{FOLLOW}(F) = \text{FIRST}(T') \cup \text{FOLLOW}(T) = \{ (, a, b, ^, +, ), \# \};$$

$$\text{FOLLOW}(P) = \text{FIRST}(F') \cup \text{FOLLOW}(F) = \{ *, (, a, b, ^, +, ), \# \}; // \text{不包含 } \varepsilon$$

(2) 证明这个方法是 LL(1) 的。

各产生式的 SELECT 集合有：

$$\text{SELECT}(E \rightarrow TE') = \text{FIRST}(T) = \{ (, a, b, ^ \}$$

$$\text{SELECT}(E' \rightarrow +E) = \{ + \}$$

$$\text{SELECT}(E' \rightarrow \varepsilon) = \text{FOLLOW}(E') = \{ ), \# \}$$

$$\text{SELECT}(T \rightarrow FT') = \text{FIRST}(F) = \{ (, a, b, ^ \}$$

$$\text{SELECT}(T' \rightarrow T) = \text{FIRST}(T) = \{ (, a, b, ^ \}$$

$$\text{SELECT}(T' \rightarrow \varepsilon) = \text{FOLLOW}(T') = \{ +, ), \# \}$$

$$\text{SELECT}(F \rightarrow PF') = \text{FIRST}(P) = \{ (, a, b, ^ \}$$

$SELECT(F' \rightarrow *F') = \{*\};$

$SELECT(F' \rightarrow \epsilon) = FOLLOW(F') = \{ (, a, b, ^, +, ), \# \};$

$SELECT(P \rightarrow (E)) = \{ ( \}$

$SELECT(P \rightarrow a) = \{ a \}$

$SELECT(P \rightarrow b) = \{ b \}$

$SELECT(P \rightarrow ^) = \{ ^ \}$

可见，相同左部产生式的 SELECT 集的交集均为空，所以文法 G[E] 是 LL(1) 文法。

(3) 构造它的预测分析表。

文法 G[E] 的预测分析表如下：

	+	*	(	)	a	b	^	#
E			TE'		TE'	TE'	TE'	
E'	+E			$\epsilon$				$\epsilon$
T			FT'		FT'	FT'	FT'	
T'	$\epsilon$		T	$\epsilon$	T	T	T	$\epsilon$
F			PF'		PF'	PF'	PF'	
F'	$\epsilon$	*F'	$\epsilon$	$\epsilon$	$\epsilon$	$\epsilon$	$\epsilon$	$\epsilon$
P			(E)		a	b	^	

(4) 构造它的递归下降分析程序。(没有布置作业)

对每个非终结符写出不带回溯的递归子程序如下：

char CH; // 存放当前的输入符号

void P\_E() // 非终结符 E 的子程序

{

if(IsIn(CH, FIRST\_TEP)) // FIRST\_TEP 为  $T \rightarrow TE'$  的右部的

FIRST 集合，产生式  $E \rightarrow TE'$

```
{
    P_T();
    P_EP();
}
else ERR;
}
void P_EP()//非终结符  $E'$  的子程序
{
    if(CH=='+')//产生式  $E' \rightarrow +E$ 
    {
        READ(CH);
        P_E();
    }
    else//产生式  $E' \rightarrow \epsilon$ 
    {
        if(IsIn(CH,FOLLOW_EP)) //FOLLOW_EP 为  $E'$  的 FOLLOW
        集合
            return ;
        else ERR;
    }
}
void P_T()//非终结符 T 的子程序
{
    if(IsIn(CH,FIRST_TEP)) //FIRST_TEP 为  $T \rightarrow TE'$  的右部的
    FIRST 集合，产生式  $T \rightarrow TE'$ 
    {
        P_F();
        P_TP();
    }
    else ERR;
}
void P_TP()//非终结符  $T'$  的子程序
{
    if(IsIn(CH,FIRST_T)) //FIRST_T 为产生式  $T' \rightarrow T$  的右部的
    FIRST 集合，产生式  $T' \rightarrow T$ 
    {
```

```

        P_T();
    }
    else//产生式  $T' \rightarrow \varepsilon$ 
    {
        if(IsIn(CH,FOLLOW_TP)) //FOLLOW_TP 为  $T'$  的 FOLLOW
        集合
            return ;
        else ERR;
    }
}
void P_F()//非终结符 F 的子程序
{
    if(IsIn(CH,FIRST_PFP)) //FIRST_PFP 为  $F \rightarrow PF'$  的右部的
    FIRST 集合，产生式  $F \rightarrow PF'$ 
    {
        P_P();
        P_FP();
    }
    else ERR;
}
void P_FP()//非终结符  $F'$  的子程序
{
    if(CH=='*') //产生式  $F' \rightarrow *F'$ 
    {
        READ(CH);
        P_FP();
    }
    else//产生式  $F' \rightarrow \varepsilon$ 
    {
        if(IsIn(CH,FOLLOW_FP)) //FOLLOW_FP 为  $F'$  的 FOLLOW
        集合
            return ;
        else ERR;
    }
}
void P_P()//非终结符 P 的子程序
{

```

```

if(CH=='(')
{
    READ(CH);
    P_E();
    if(CH=='') READCH(CH);
    else
        ERR;
}
else if(CH=='a') READ(CH);
else if(CH=='b') READ(CH);
else if(CH=='^') READ(CH);
else ERR;
}

```

3. 已知文法  $G[S]: S \rightarrow MH \mid a$

$$H \rightarrow LSo \mid \varepsilon$$

$$K \rightarrow dML \mid \varepsilon$$

$$L \rightarrow eHf$$

$$M \rightarrow K \mid bLM$$

判断  $G$  是否是 LL (1) 文法, 如果是, 构造 LL (1) 分析表, 并分析符号串 bef 是否为文法的句子。

解:

(1) 首先求各非终结符的 **FIRST** 集合:

$$\text{FIRST}(S) = \{a\} \cup \text{FIRST}(M) \cup \text{FIRST}(H) = \{a\} \cup \{b, d, \varepsilon\} \cup \{e, \varepsilon\} = \{a, b, d, e, \varepsilon\};$$

$$\text{FIRST}(H) = \text{FIRST}(L) \cup \{\varepsilon\} = \{e, \varepsilon\};$$

$$\text{FIRST}(K) = \{d, \varepsilon\};$$

$$\text{FIRST}(L) = \{e\};$$

$$\text{FIRST}(M) = \text{FIRST}(K) \cup \{b\} = \{b, d, \varepsilon\};$$

然后求非终结符的 FOLLOW 集合：

$$\text{FOLLOW}(S) = \{o, \#\}$$

$$\text{FOLLOW}(H) = \text{FOLLOW}(S) \cup \{f\} = \{f, o, \#\}$$

$$\text{FOLLOW}(K) = \text{FOLLOW}(M) = \text{FIRST}(H) \cup \text{FOLLOW}(S) = \{e, o, \#\}; //$$

不包含  $\epsilon$

$$\begin{aligned} \text{FOLLOW}(L) &= \text{FIRST}(S) \cup \{o\} \cup \text{FOLLOW}(K) \cup \text{FIRST}(M) \cup \\ &\text{FOLLOW}(M) \end{aligned}$$

$$= \{a, b, d, e, o, \#\} \cup \text{FOLLOW}(M) = \{a, b, d, e, o, \#\}; // \text{不包含 } \epsilon$$

$$\begin{aligned} \text{FOLLOW}(M) &= \text{FIRST}(L) \cup \text{FIRST}(H) \cup \text{FOLLOW}(S) = \{e, o, \#\} // \text{不} \\ &\text{包含 } \epsilon \end{aligned}$$

最后求各产生式的 SELECT 集合：

$$\begin{aligned} \text{SELECT}(S \rightarrow MH) &= (\text{FIRST}(MH) - \{\epsilon\}) \cup \text{FOLLOW}(S) = \{b, d, e\} \cup \\ &\{o, \#\} = \{b, d, e, o, \#\}; \end{aligned}$$

$$\text{SELECT}(S \rightarrow a) = \{a\}$$

$$\text{SELECT}(H \rightarrow LSo) = \{e\}$$

$$\text{SELECT}(H \rightarrow \epsilon) = \text{FOLLOW}(H) = \{f, o, \#\}$$

$$\text{SELECT}(K \rightarrow dML) = \{d\}$$

$$\text{SELECT}(K \rightarrow \epsilon) = \text{FOLLOW}(K) = \{e, o, \#\}$$

$$\text{SELECT}(L \rightarrow eHf) = \{e\}$$

$$\text{SELECT}(M \rightarrow K) = (\text{FIRST}(K) - \{\epsilon\}) \cup \text{FOLLOW}(M) = \{d, e, o, \#\}$$

$$\text{SELECT}(M \rightarrow bLM) = \{b\}$$

可见，相同左部产生式的 SELECT 集的交集均为空，所以文法  $G[S]$  是 LL(1) 文法。

(2) 文法  $G[E]$  的预测分析表如下：

	a	o	d	e	f	b	#
S	a	MH	MH	MH		MH	MH
H		$\varepsilon$		LSO	$\varepsilon$		$\varepsilon$
K		$\varepsilon$	dML	$\varepsilon$			$\varepsilon$
L				eHf			
M		K	K	K		blm	K

(3) 下面用预测分析程序、栈和预测分析表对符号串 bef 进行分析，给出栈的变化过程如下表：

步骤	符号栈	读入符号	剩余符号串	使用产生式
1	#S	b	ef#	$S \rightarrow MH$
2	#HM	b	ef#	$M \rightarrow bLM$
3	#HMLb	b	ef#	“b” 匹配
4	#HML	e	f#	$L \rightarrow eHf$
5	#HMfHe	e	f#	“e” 匹配
6	#HMfH	f	#	$H \rightarrow \varepsilon$
7	#HMf	f	#	“f” 匹配
8	#HM	#		$M \rightarrow K$
9	#HK	#		$K \rightarrow \varepsilon$
10	#H	#		$H \rightarrow \varepsilon$
11	#	#		接受

分析成功，所以符号串 bef 是文法的句子。

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？试对下面方法进行改写，并对改写后的文法进行判



断。

$$(1) A \rightarrow baB | \varepsilon$$

$$B \rightarrow Abb | a$$

$$(2) A \rightarrow aABe | a$$

$$B \rightarrow Bb | d$$

解：对于一个文法若消除了左递归，提取了左公共因子后不一定为 LL(1)文法。如果新的文法中无空产生式，则一定为 LL(1)文法，如果有空产生式，则需要进行 LL(1)判断才能决定新方法是否一定是 LL(1)文法。

(1) 由于  $SELECT(A \rightarrow baB) = \{b\}$ ,  
 $SELECT(A \rightarrow \varepsilon) = FOLLOW(A) = \{b, \#\}$ , 两集合有交集, 所以该文法不是 LL(1)方法。

该文法已经消除了左递归与左公共因子, 一般来说是不能再改写了。但根据本文法的具体情况有以下改写:

用产生式  $A \rightarrow baB$  与  $A \rightarrow \varepsilon$  分别替换产生式  $B \rightarrow Abb$  有:  
 $B \rightarrow baBbb | bb$ , 提取这两个新产生式的左公共因子有:

$B \rightarrow bB'$ ,  $B' \rightarrow aBbb | b$ , 这样改写后文法  $G'[A]$  为:

$$A \rightarrow baB | \varepsilon$$

$$B \rightarrow bB' | a$$

$$B' \rightarrow aBbb | b$$

每个产生式的 SELECT 集合为:

$$SELECT(A \rightarrow baB) = \{b\}$$

$$SELECT(A \rightarrow \varepsilon) = \Phi$$

$$\text{SELECT}(B \rightarrow bB') = \{b\}$$

$$\text{SELECT}(B \rightarrow a) = \{a\}$$

$$\text{SELECT}(B' \rightarrow aBbb) = \{a\}$$

$$\text{SELECT}(B' \rightarrow b) = \{b\}$$

可见，相同左部产生式的 SELECT 集的交集均为空，所以文法  $G'[A]$  是 LL(1) 文法。

(2) 显然文法的第 1, 2 个产生式的右部具有左公共因子 a，而产生式  $B \rightarrow Bb$  具有左递归，因此文法可改写为：

$$A \rightarrow aA'$$

$$A' \rightarrow ABe | \varepsilon$$

$$B \rightarrow dB'$$

$$B' \rightarrow bB' | \varepsilon$$

由于  $\text{SELECT}(A' \rightarrow ABe) = \{a\}$ ,

$\text{SELECT}(A' \rightarrow \varepsilon) = \text{FOLLOW}(A') = \text{FOLLOW}(A) = \text{FIRST}(B) \cup \{\#\} = \{d, \#\}$   
，交集为空。

而  $\text{SELECT}(B' \rightarrow bB') = \{b\}$ ,

$\text{SELECT}(B' \rightarrow \varepsilon) = \text{FOLLOW}(B') = \text{FOLLOW}(B) = \{e\}$ ，交集也为空。

而非终结符 A 与 B 都只有一个产生式，不存在求 SELECT 的交集问题。所以改写后的方法为 LL(1) 文法。