

## 第四章

fsys 随不同上下文可以有所变化。例如,如果是在处理 write 语句内部的表达式而调用  $\text{expression}(\text{fsys}, \dots)$  时,则可以将表 4.6 中“表达式”的 FOLLOW 集合传给 fsys;然而,如果是在处理“因子”下一层的表达式而调用  $\text{expression}(\text{fsys}, \dots)$  时,则可以将这个 FOLLOW 集合中的符号“,”去掉后传给 fsys。

当然,还可以有更多的考虑,如关系运算符只会在“条件”上下文中,所以在其他上下文中调用  $\text{expression}(\text{fsys}, \dots)$  时,fsys 中可以去掉这些关系运算符。

附录 A 的 PL/0 编译程序中,不同语法单位的错误处理过程没有统一标准,实现的精细程度有所不同。值得注意的是,一些错误恢复代码的加入影响到了整个代码的可读性,读者在阅读时应当将它们从主体代码中区分出来。

### 练习

1. 对文法  $G[S]$

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

$T \rightarrow T, S \mid S$

(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。

$$\begin{aligned} T &\rightarrow ST' \\ T' &\rightarrow ,ST' \mid \varepsilon \end{aligned}$$

- (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。

2. 对下面的文法 G:

$E \rightarrow TE$   
 $E \rightarrow +E|e$   
 $T \rightarrow FT$   
 $T \rightarrow T|e$   
 $F \rightarrow PF$   
 $F \rightarrow *F|e$   
 $P \rightarrow (E)|a|b|A$

计算这个文法的每个非终结符的 FIRST 集和 FOLLOW 集。

- (1) 计算这个文法的每个非终结符的 FIRST 集和 FOLLOW 集。  
 (2) 证明这个文法是 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。

3. 已知文法 G[S]:

$S \rightarrow MH|a$   
 $H \rightarrow LSo|e$   
 $K \rightarrow dM|e$   
 $L \rightarrow eHf$   
 $M \rightarrow K^1bLM$

判断 G 是否是 LL(1) 文法, 如果是, 构造 LL(1) 分析表。

4. 证明下述文法不是 LL(1) 的。

$S \rightarrow C\$$

$C \rightarrow bA|aB$

$A \rightarrow a|aC|bAA$

$B \rightarrow b|bC|aBB$

能否构造一等价的文法, 使其是 LL(1) 的? 并给出判断过程。

5. 文法 G 如下:

<程序>  $\rightarrow$  begin <语句表> end

<语句表>  $\rightarrow$  <语句> | <语句表> , <语句>

<语句>  $\rightarrow$  <无条件语句> | <条件语句>

<无条件语句>  $\rightarrow$  a

<条件语句>  $\rightarrow$  <如果语句> | <如果语句> else <语句>

<如果语句>  $\rightarrow$  <如果子句> <无条件语句>

<如果子句>  $\rightarrow$  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。

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

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

(2)  $S \rightarrow AB$   
 $A \rightarrow Ba$   
 $B \rightarrow Db$   
 $D \rightarrow d|$

(3)  $S \rightarrow ad$   
 $A \rightarrow S$   
 $B \rightarrow b$

(4)  $S \rightarrow i$   
 $E \rightarrow i$

(5)  $S \rightarrow$   
 $A \rightarrow$   
 $B \rightarrow$

(6)  $M \rightarrow$   
 $H \rightarrow$

7. 对下面的文法

(1)  $A \rightarrow$   
 $E \rightarrow$

(2)  $A \rightarrow$   
 $E \rightarrow$

(3)  $A \rightarrow$   
 $E \rightarrow$

(4)  $A \rightarrow$   
 $E \rightarrow$

(5)  $A \rightarrow$   
 $E \rightarrow$

(6)  $A \rightarrow$   
 $E \rightarrow$

(7)  $A \rightarrow$   
 $E \rightarrow$

(8)  $A \rightarrow$   
 $E \rightarrow$

(9)  $A \rightarrow$   
 $E \rightarrow$

(10)  $A \rightarrow$   
 $E \rightarrow$

(11)  $A \rightarrow$   
 $E \rightarrow$

(12)  $A \rightarrow$   
 $E \rightarrow$

(13)  $A \rightarrow$   
 $E \rightarrow$

(14)  $A \rightarrow$   
 $E \rightarrow$

(15)  $A \rightarrow$   
 $E \rightarrow$

(16)  $A \rightarrow$   
 $E \rightarrow$

(17)  $A \rightarrow$   
 $E \rightarrow$

(18)  $A \rightarrow$   
 $E \rightarrow$

(19)  $A \rightarrow$   
 $E \rightarrow$

(20)  $A \rightarrow$   
 $E \rightarrow$

(21)  $A \rightarrow$   
 $E \rightarrow$

(22)  $A \rightarrow$   
 $E \rightarrow$

(23)  $A \rightarrow$   
 $E \rightarrow$

(24)  $A \rightarrow$   
 $E \rightarrow$

(25)  $A \rightarrow$   
 $E \rightarrow$

(26)  $A \rightarrow$   
 $E \rightarrow$

(27)  $A \rightarrow$   
 $E \rightarrow$

(28)  $A \rightarrow$   
 $E \rightarrow$

(29)  $A \rightarrow$   
 $E \rightarrow$

(30)  $A \rightarrow$   
 $E \rightarrow$

(31)  $A \rightarrow$   
 $E \rightarrow$

(32)  $A \rightarrow$   
 $E \rightarrow$

(33)  $A \rightarrow$   
 $E \rightarrow$

(34)  $A \rightarrow$   
 $E \rightarrow$

$$(1) S \rightarrow A|B$$

$$A \rightarrow aA|a$$

$$B \rightarrow bB|b$$

$$(2) S \rightarrow AB$$

$$A \rightarrow Ba|\epsilon$$

$$B \rightarrow Db|D$$

$$D \rightarrow d|\epsilon$$

$$(3) S \rightarrow aAaB|bAbB$$

$$A \rightarrow S|db$$

$$B \rightarrow bB|a$$

$$(4) S \rightarrow |(E)$$

$$E \rightarrow E+S|E-S|S$$

$$(5) S \rightarrow SaA|bB$$

$$A \rightarrow aB|c$$

$$B \rightarrow bB|d$$

$$(6) M \rightarrow MaH|H$$

$$H \rightarrow b(M)|(M)|b$$

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。

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

$$B \rightarrow Abb|a$$

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

$$B \rightarrow Bb|d$$

$$(3) S \rightarrow Aa|b$$

$$A \rightarrow SB$$

$$B \rightarrow ab$$

$$(4) S \rightarrow AS|b$$

$$A \rightarrow SA|a$$

$$(5) S \rightarrow Ab|Ba$$

$$A \rightarrow aA|a$$

$$B \rightarrow a$$

$$(6) S \rightarrow aSbS|bSaS|\epsilon$$

8. 按照本章介绍的消除一切左递归算法消除下面文法中的左递归(要求依非终结符的两种排序方式 S、Q、P 和 Q、P、S 分别执行该算法):

$$S \rightarrow PQ|a$$

$$P \rightarrow QS|b$$

$$Q \rightarrow SP|c$$



9. 对下面的文法:

```
<bexpr> → <bexpr> or <bterm> | <bterm>  
<bterm> → <bterm> and <bfactor> | <bfactor>  
<bfactor> → not <bfactor> | (<bexpr>) | true | false
```

构造一个预测分析器。

10. 试为语言  $L$  写一 LL(1) 文法, 其中  $L = \{w \mid w \in (a|b)^* \text{ 且 } w \text{ 中 } a, b \text{ 的个数相等}\}$ 。

11. 在阅读附录 A 中 PL/0 编译程序的基础上,
- (1) 熟悉该编译程序的整体架构, 识别出各语法单位对应的子程序;
  - (2) 对应表 4.3 给出的语法规则, 读懂各子程序之间的调用关系;
  - (3) 读懂词法分析程序是如何接入到编译程序主体的;
  - (4) 进一步阅读 PL/0 编译程序, 读懂出错处理相关的代码。

12. 对附录 A 中 PL/0 编译程序源码进行裁减和改造, 使其仅包含词法和语法分析过程。该分析程序读入 PL/0 语言源程序, 如果没有发现词法或语法错误, 再输出其相应的语法分析树。语法分析树的显示格式可自行设计, 建议采用缩进的文本表示形式。

## 第五章

$$\begin{aligned} D &\rightarrow D(T) | H \\ H &\rightarrow a | (S) \\ T &\rightarrow T+S | S \end{aligned}$$

其中  $V_N = \{S, D, T, H\}$ ,  $V_T = \{+, (, ), a, +\}$ ,  $S$  为开始符号。  
对应的算符优先关系矩阵如表 5.14 所示。

表 5.14 算符优先关系矩阵表

	$+$	$($	$)$	$a$	$+$	$\#$
$+$	$\triangleright$	$\triangleleft$	$\triangleright$	$\triangleleft$	$\triangleright$	$\triangleright$
$($	$\triangleleft$	$\triangleleft$	$\equiv$	$\triangleleft$	$\triangleleft$	$\triangleright$
$)$	$\triangleright$	$\triangleright$	$\triangleright$	$\triangleright$	$\triangleright$	$\triangleright$
$a$	$\triangleright$	$\triangleright$	$\triangleright$	$\triangleright$	$\triangleright$	$\triangleright$
$+$	$\triangleleft$	$\triangleleft$	$\triangleright$	$\triangleleft$	$\triangleright$	$\triangleright$
$\#$	$\triangleleft$	$\triangleleft$		$\triangleleft$		$\equiv$

读者自己可以用算符优先分析法对输入串  $(a+a)\#$  进行分析, 不难发现, 它可以完全正确地进行归约, 然而  $(a+a)\#$  却不是该文法能推导出的句子。此外, 通常一个适用语言的文法也很难满足算符优先文法的条件, 因而算符优先分析法仅适用于表达式的语法分析。

## 练 习

1. 已知文法  $G[S]$  为:

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

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

(1) 计算  $G[S]$  的 FIRSTVT 和 LASTVT。

(2) 构造  $G[S]$  的算符优先关系表并说明  $G[S]$  是否为算符优先文法。

(3) 计算  $G[S]$  的优先函数。

(4) 给出输入串  $(a,a)\#$  和  $(a,(a,a))\#$  的算符优先分析过程。

2. 对题 1 的  $G[S]$ :

(1) 给出  $(a,(a,a))$  和  $(a,a)$  的最右推导和规范归约过程。

(2) 将 (1) 和题 1 中的 (4) 进行比较, 说明算符优先归约和规范归约的区别。

3. 有文法  $G[S]$ :

$$S \rightarrow V$$

$$V \rightarrow T | V_1 T$$

$$T \rightarrow F \mid T + F$$

$$F \rightarrow V \mid ($$

- (1) 给出  $(+i)$  的规范推导。
- (2) 指出句型  $F + Fi$  的短语、句柄和素短语。
- (3)  $G[S]$  是否 OPG, 若是, 给出 (1) 中句子的分析过程。

4. 已知文法  $G[S]$  为

$$S \rightarrow S; G \mid G$$

$$G \rightarrow G(T) \mid H$$

$$H \rightarrow a \mid (S)$$

$$T \rightarrow T + S \mid S$$

- (1) 构造  $G[S]$  的算符优先关系表, 并判断  $G[S]$  是否为算符优先文法。
- (2) 给出句型  $a(T+S); H; (S)$  的短语、句柄、素短语和最左素短语。
- (3) 给出  $a; (a+a)$  和  $(a+a)$  的分析过程, 说明它们是否为  $G[S]$  的句子。
- (4) 给出 (3) 中输入串的最右推导, 分别说明两个输入串是否为  $G[S]$  的句子。
- (5) 由 (3) 和 (4) 说明了算符优先分析的哪些缺点?
- (6) 算符优先分析过程和规范归约过程都是最右推导的逆过程吗?

5. 已知布尔表达式文法  $G[B]$  为

$$B \rightarrow B \circ T \mid T$$

$$T \rightarrow T a F \mid F$$

$$F \rightarrow n F \mid (B) \mid i \mid f$$

- (1)  $G[B]$  是算符优先文法吗?
- (2) 若  $G[B]$  是算符优先文法, 请给出输入串  $ntofat\#$  的分析过程。

## 第六章



## 练习

### 1. 已知文法

$$A \rightarrow aAd \mid aAb \mid \epsilon$$

判断该文法是否是 SLR(1) 文法, 若是, 请构造相应分析表, 并对输入串  $ab$  给出分析过程。

### 2. 若有定义二进制的文法如下:

$$S \rightarrow L \mid L \mid L$$

$$L \rightarrow LB \mid B$$

$$B \rightarrow 0 \mid 1$$

(1) 试为该文法构造 LR 分析表, 并说明属哪类 LR 分析表。

(2) 给出输入串 101.110 的分析过程。

### 3. 考虑文法

$$S \rightarrow AS \mid b$$

$$A \rightarrow SA \mid a$$

(1) 列出该文法的所有 LR(0) 项目。

(2) 按 (1) 列出的项目构造识别该文法活前缀的 NFA, 把这个 NFA 确定化为 DFA, 说明这个 DFA 的所有状态全体构成该文法的 LR(0) 规范族。

(3) 这个文法是 SLR(1) 的吗? 若是, 构造出它的 SLR 分析表。

(4) 这个文法是 LALR(1) 或 LR(1) 的吗?

4. 下面是一个描述  $\Sigma = \{a, b\}$  上的正规式的 LALR(1) 文法 (实际上也是 SLR(1) 文法), 只不过用  $\mid$  代替  $|$ 。

$$E \rightarrow E \mid T \mid T$$

$$T \rightarrow TF \mid F$$

$$F \rightarrow F * (E) \mid a \mid b$$

构造该文法的 LALR(1) 项目集和分析表。

### 5. 一个类 ALGOL 的文法如下:

$$\langle \text{Program} \rangle \rightarrow \langle \text{Block} \rangle$$

$$\langle \text{Block} \rangle \rightarrow \langle \text{Block head} \rangle \langle \text{Compound Statement} \rangle$$

$$\langle \text{Block head} \rangle \rightarrow \text{begin } d$$

$$\langle \text{Block head} \rangle \rightarrow \langle \text{Block head} \rangle ; d$$

$$\langle \text{Compound Tail} \rangle \rightarrow S \text{ end}$$

$$\langle \text{Compound Tail} \rangle \rightarrow S ; \langle \text{Compound Tail} \rangle$$

$$\langle \text{Compound Statement} \rangle \rightarrow \text{begin} \langle \text{Compound Tail} \rangle$$

试构造其 LR(0) 分析表。

6. 文法  $G = (\{U, T, S\}, \{a, b, c, d, e\}, P, S)$ , 其中  $P$  为

$$S \rightarrow UTa \mid Tb$$

$T \rightarrow S | Sc | d$   
 $U \rightarrow US | e$

- (1) 判断  $G$  是 LR(0), SLR(1), LALR(1) 还是 LR(1) 的, 说明理由。  
 (2) 构造相应的分析表。

7. 证明下面文法不是 LR(0) 而是 SLR(1)。

$S \rightarrow A$   
 $A \rightarrow Ab | bBa$   
 $B \rightarrow aAc | a | aAb$

8. 证明文法 (其中  $\$$  相当于 #)

$S \rightarrow A \$$   
 $A \rightarrow BaBb | DbDa$   
 $B \rightarrow e$   
 $D \rightarrow e$

是 LR(1) 而不是 SLR(1) 的。

9. 证明下面文法是 LR(1) 而不是 LALR(1) 的。

$S \rightarrow Aa | bAe | Be | bBa$   
 $A \rightarrow d$   
 $B \rightarrow d$

10. 判断下列 6 个文法是否为 LR 类文法, 若是, 请说明是 LR(0), SLR(1), LALR(1) 或 LR(1) 的哪一种, 并构造相应的分析表; 若不是, 请说明理由。

- (1)  $S \rightarrow AB$

$A \rightarrow aBa | e$   
 $B \rightarrow bAb | e$

- (2)  $S \rightarrow D; B | B$

$D \rightarrow d | e$   
 $B \rightarrow B; a | a | e$

- (3)  $S \rightarrow aAd | eBd | aBr | eAr$

$A \rightarrow a$   
 $B \rightarrow a$

- (4)  $A \rightarrow AbBa | B$

$B \rightarrow a | e$

- (5)  $A \rightarrow aB | e$

$B \rightarrow Ab | a$

- (6)  $S \rightarrow (SR | a$

$R \rightarrow , SR | )$

11. 设文法  $G[S]$  为

$S \rightarrow AS | e$   
 $A \rightarrow aA | b$

- (1) 证明  $G[S]$  是 LR(1) 文法。



- (2) 构造它的 LR(1) 分析表。  
 (3) 给出输入符号串  $abab\#$  的分析过程。  
 12. 证明任何 SLR(1) 文法都是 LR(1) 文法。  
 13. 证明任何 SLR(1) 文法都是 LALR(1) 文法。  
 14. 一个文法若是 SLR(1) 文法, 那么它的 SLR(1) 和 LALR(1) 识别活前缀的 DFA 的状态数目相同, 但 LALR(1) 对一些错误的发现可能比 SLR(1) 要早。请说明理由。

15. 已知文法为:

$S \rightarrow a \mid \Lambda \mid (T)$

$T \rightarrow T, S \mid S$

(1) 构造它的 LR(0), LALR(1) 和 LR(1) 分析表。

(2) 给出对输入符号串  $(a\#)$  和  $(a,a\#)$  的分析过程。

(3) 说明(1)中 3 种分析表发现错误的时刻和输入串的出错位置有何区别。

16. 给定文法:

$S \rightarrow do S \text{ or } S \mid do S \mid S \mid acc$

(1) 构造识别该文法活前缀的 DFA。

(2) 该文法是 LR(0) 的吗? 是 SLR(1) 的吗? 说明理由。

(3) 若对一些终结符的优先级以及算符的结合规则规定如下:

① or 优先级大于 do。

② 服从左结合。

③ 优先级大于 do。

④ 优先级大于 or。

请构造该文法的 LR 分析表。

17. 已知某文法  $G[S]$  的 LALR(1) 分析表如表 6.19 所示。

表 6.19 对表达式二义性文法的 LR 分析表

状态	ACTION					GOTO
	a	t	g	c	#	
0	$S_{11}$	$S_1$				S
1						
2				$S_1$		1
3	$S_{11}$	$S_1$	$S_1$	$S_1$	acc	
4	$S_1$			$S_1$		
5	$S_1$					
6						16
7						
8			$r_1$	$S_1$		
9			$S_1$	$r_1$		
10						
11	$S_{11}$	$S_1$		$S_{11}$	$r_1$	
12	$S_{11}$	$S_1$		$S_1$		
13	$S_{11}$	$S_1$	$S_{11}$	$S_1$		
14				$S_1$		14
15			$r_1$	$S_1$		12
16			$r_1$	$S_1$	$r_1$	15

并且已知各规

规则编号

右部长度

左部符号

写出使用  
的  $r_i, s_j$  序列

18. 给定

$S \rightarrow if S$

$S \rightarrow if S$

$S \rightarrow a$

为文法

识别活前缀

(1)

(2)

(3)

(4)

(5)

态冲突

(6)

然而,

经过的

数不超

种分

并且已知各规则右边语法符号的个数以及左边的非终结符如下：

规则编号	1	2	3	4
右部长度	4	4	4	4
左部符号	S	S	S	S

写出使用上述 LALR(1) 分析器分析下面的串的过程(只需写出前 10 步, 列出所有可能的  $r, m$  序列, 注意先后次序):

acaaccgtgccacgatgccaa...

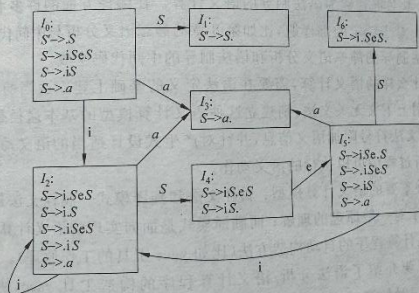
18. 给定如下文法  $G[S]$ :

$S \rightarrow \underline{i} S \text{ else } S$

$S \rightarrow \underline{i} S$

$S \rightarrow a$

为文法  $G[S]$  增加产生式  $S' \rightarrow S$ , 得到增广文法  $G'[S]$ , 下图是相应的 LR(0) 项目集和识别活前缀的 DFA (i 表示 if, e 表示 else):



- 指出该 DFA 中的全部冲突状态及其冲突类型, 以说明文法  $G[S]$  不是 LR(0) 文法。
- 文法  $G[S]$  也不是 SLR(1) 文法。为什么?
- 给出文法  $G[S]$  的 LR(1) 项目集和相应的 DFA。
- 指出(3)中 DFA 的全部冲突状态, 以说明文法  $G[S]$  也不是 LR(1) 文法。
- 若规定 else 优先匹配左边靠近它的未匹配的 if, 则可以解决上述两个 DFA 中的状态冲突。试给出文法  $G[S]$  在给定这一规则情况下的 SLR(1) 分析表和 LR(1) 分析表。
- 对于文法  $G[S]$  中正确的句子, 基于(5)中两个分析表均可以成功进行 LR 分析。然而, 对于不属于文法  $G[S]$  中的句子, 两种分析过程发现错误的速度不同, 即发现错误时所经过的移进-归约总步数有差异。试给出一个长度不超过 10 的句子(即所包含的终结符个数不超过 10), 使得两种分析过程发现错误的速度不同。哪一个更快? 对于你给的例子, 两种分析过程分别到达哪个状态时会发现错误?