

## 第四章作业

### 第一次作业

4.2.1: 考虑上下文无关文法  $S \rightarrow S S + \mid S S * \mid a$  以及串  $aa+a*$

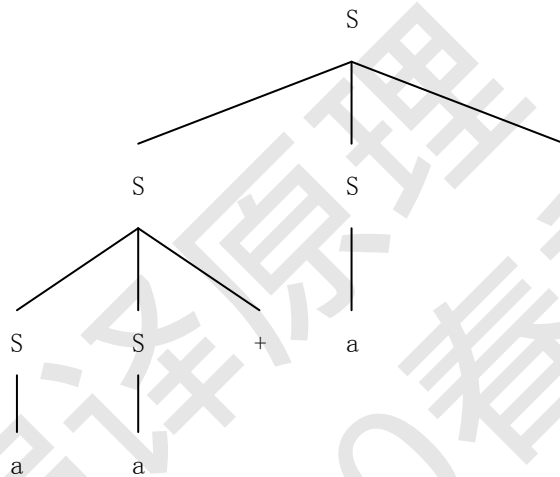
- 1) 给出这个串的一个最左推导；2) 给出这个串的一个最右推导；
- 3) 给出这个串的一颗语法分析树；4) 这个文法是否是二义性的？证明你的回答（选作）；5) 这个文法生成的语言是什么？

#### 练习 4.2.1

1)  $S \Rightarrow SS^* \Rightarrow SS+S^* \Rightarrow aS+S^* \Rightarrow aa+S^* \Rightarrow aa+a^*$

2)  $S \Rightarrow SS^* \Rightarrow Sa^* \Rightarrow SS+a^* \Rightarrow Sa+a^* \Rightarrow aa+a^*$

3)



4) 没有二义性。

(1) 先证明一个该文法产生串的长度的结论：设串的推导过程中使用产生式  $S \Rightarrow SS+$  和  $S \Rightarrow SS^*$  的次数为  $m$ ，则串的长度  $L=2*m+1$ ，且串中包含  $m$  个运算符(+或\*)和  $(m+1)$  个  $a$ 。

1) 当  $m=0$  时，仅有  $S \Rightarrow a$  一种情况， $L=1$ ，串由 1 个  $a$  和 0 个运算符构成，结论成立；

2) 设当  $m < k(k \geq 1)$  时结论成立，则当  $m=k$  时，第一步推导必然为

$$S \Rightarrow S_1 S_2 op$$

$op$  为+或\*， $S$  下标仅用于区分  $S$  的多次出现。设  $S_1 \xRightarrow{*} \alpha$ ， $S_2 \xRightarrow{*} \beta$ ， $\alpha$ 、 $\beta$  均为使用  $S \Rightarrow S_1 S_2 op$  少于  $k$  次得到的串，设二者推导过程中分别使用该产生式  $k_1$  和  $k_2$  次，根据假设有：

$$L(\alpha)=2*k_1+1, L(\beta)=2*k_2+1$$

串长度  $L=L(\alpha)+L(\beta)+1=2*(k_1+k_2+1)+1=2*k+1$ ；且串中  $a$  的个数为  $(k_1+1)+(k_2+1)=k+1$ ；运算符个数为  $k_1+k_2+1=k$ ，故结论成立。

(2) 下面证明该文法无二义性，对串的长度做归纳。由前述证明可知，该文法产生的串长  $L$  可为任意非负奇数。对由该文法得到的长度为  $L=2*k+1$  串  $\omega$ ：

1) 当  $k=0$  时， $L=2*0+1=1$ ，只有  $S \Rightarrow a$  一种情况，显然没有二义性。

2) 设当  $k < n$  时结论成立。  $S \xRightarrow{*} \omega$ , 根据  $\omega$  末尾运算符可确定第一步推导使用的产生式, 不妨设为:

$$S \Rightarrow S_1 S_2 +$$

从后向前处理串  $\omega$ , 除去末尾的运算符, 找到可以由  $S$  推导出的最短的串  $\alpha$ , 设  $\alpha$  长度为  $m_1$ , 由前述结论可知  $m_1 = 2 * k_1 + 1$ , 且  $\alpha$  包含  $k_1$  个运算符与  $(k_1 + 1)$  个  $a$ ,

由归纳假设可知  $\alpha$  无二义性, 存在唯一的最左推导  $S \xRightarrow[lm]{*} \alpha$ ;

设串  $\omega$  剩余部分为  $\beta$ , 设  $\beta$  长度为  $m_2$ , 同理可知  $m_2 = 2 * k_2 + 1$ ,  $\beta$  包含  $k_2$  个运算符与  $(k_2 + 1)$  个  $a$ , 存在唯一最左推导  $S \xRightarrow[lm]{*} \beta$ , 且满足  $k = k_1 + k_2$ 。

此时串  $\omega$  可表示成如下形式:

$$\omega = \beta \alpha +$$

故存在唯一的最左推导:

$$S \Rightarrow S S + \xRightarrow[lm]{*} \beta S + \xRightarrow[lm]{*} \beta \alpha +$$

此时仍不存在二义性。

综上所述, 该文法不具有二义性。

5) 由字符  $a$  与运算符  $+$ 、 $*$  构成的后缀表达式。

### 练习 4.2.3

为 4.2.3 的第一题的语言设计文法: 所有由 0 和 1 组成的并且每个 0 之后都至少跟着一个 1 的串的集合

$$S \rightarrow 01S \mid 1S \mid \varepsilon$$

### 练习 4.3.1

1) 该文法无左公因子

2) 不能, 因为有左递归存在

3)  $\text{rexpr} \rightarrow \text{rterm rexpr}'$

$$\text{rexpr}' \rightarrow + \text{rterm rexpr}' \mid \varepsilon$$

$$\text{rterm} \rightarrow \text{rfactor rterm}'$$

$$\text{rterm}' \rightarrow \text{rfactor rterm}' \mid \varepsilon$$

$$\text{rfactor} \rightarrow \text{rprimary rfactor}'$$

$$\text{rfactor}' \rightarrow * \text{rfactor}' \mid \varepsilon$$

$$\text{rprimary} \rightarrow a \mid b$$

4) 适合

4.3.1: 下面是一个只包含符号  $a$  和  $b$  的正则表达式的文法, 其中用  $+$  替代表示并运算的字符  $|$ , 以避免和文法中作为元符号使用竖线混淆

$$\text{rexpr} \rightarrow \text{rexpr} + \text{rterm} \mid \text{rterm}$$

$$\text{rterm} \rightarrow \text{rterm rfactor} \mid \text{rfactor}$$

$$\text{rfactor} \rightarrow \text{rfactor}^* \mid \text{rprimary}$$

$$\text{rprimary} \rightarrow a \mid b$$

1) 对该文法提取左公因子; 2) 提取左公因子的变换能是这个文法适用于自顶向下的语法分析技术吗? 3) 将提取了左公因子的文法继续消除左递归; 4) 此时得到的文法适用于自顶向下的语法分析吗?

## 第二次作业

### 1. 使得文法的预测分析产生回溯的原因是什么？仅使用 FIRST 集合可以避免回溯吗？为什么？

即使当非终结符用某个产生式匹配成功，但是这种成功可能只是暂时的，因为没有足够的信息来唯一地确定可能的产生式，所以分析过程就会产生回溯。不可以。例如对于产生式  $A \rightarrow \alpha \mid \beta$ ,  $FIRST(\alpha)$  与  $FIRST(\beta)$  交集为空集，但  $\epsilon$  是其中某个 FIRST 集合的元素，不是一般性，假设  $\epsilon \in FIRST(\alpha)$ ，想要避免回溯，则还需要考虑  $FOLLOW(A)$  与  $FIRST(\beta)$  的情况

### 2.

#### 考虑文法：

$lexp \rightarrow atom \mid list$   
 $atom \rightarrow number \mid identifier$   
 $list \rightarrow (lexp-seq)$   
 $lexp-seq \rightarrow lexp-seq \ lexp \mid lexp$

#### a) 消除左递归

$lexp \rightarrow atom \mid list$   
 $atom \rightarrow number \mid identifier$   
 $list \rightarrow (lexp-seq)$   
 $lexp-seq \rightarrow lexp \ lexp-seq'$   
 $lexp-seq' \rightarrow lexp \ lexp-seq' \mid \epsilon$

#### a. 消除左递归

#### b. 求得该文法的 FIRST 集合和 FOLLOW 集合

#### c. 说明所得的文法是 LL(1) 文法

#### d. 为所得的文法构造 LL(1) 分析表

#### e. 对输入串 (a (b (2)) (c)) 给出相应得 LL(1) 分析程序的动作

#### b) 求该文法的 FIRST 集合和 FOLLOW 集合

$FIRST(lexp) = \{ number, identifier, ( \}$

$FIRST(atom) = \{ number, identifier \}$

$FIRST(list) = \{ ( \}$

$FIRST(lexp-seq) = \{ number, identifier, ( \}$

$FIRST(lexp-seq') = \{ \epsilon, number, identifier, ( \}$

$FOLLOW(lexp) = \{ \$, ), number, identifier, ( \}$

$FOLLOW(atom) = \{ \$, ), number, identifier, ( \}$

$FOLLOW(list) = \{ \$, ), number, identifier, ( \}$

$FOLLOW(lexp-seq) = \{ ) \}$

$FOLLOW(lexp-seq') = \{ ) \}$

#### c) 说明所得的文法是 LL(1) 文法

可以根据 LL(1) 文法的定义来证明

因为对于：1)  $lexp$  为左部的产生式，有  $FIRST(atom) \cap FIRST(list) = \phi$ ，且

2)  $atom$  为左部的产生式， $FIRST(number) \cap FIRST(identifier) = \phi$ ，且

3)  $lexp$  为左部的产生式， $FIRST(lexp \ lexp-seq') \cap FIRST(\epsilon) = FIRST(lexp)$

$\cap FIRST(\epsilon) = \phi$ ，且  $FIRST(lexp-seq') \cap FOLLOW(lexp-seq') = \phi$

所以该文法是 LL(1) 文法

#### d) 为所得的文法构造 LL(1) 分析表

非终结符	输入符号				
	Number	Identifier	(	)	\$
lexp	lexp $\rightarrow$ atom	lexp $\rightarrow$ atom	lexp $\rightarrow$ list		
atom	atom $\rightarrow$ number	atom $\rightarrow$ identifier			
list			list $\rightarrow$ ( lexp-seq )		
lexp-seq	lexp-seq $\rightarrow$ lexp lexp-seq'	lexp-seq $\rightarrow$ lexp lexp-seq'	lexp-seq $\rightarrow$ lexp lexp-seq'		
lexp-seq'	lexp-seq' $\rightarrow$ lexp lexp-seq'	lexp-seq' $\rightarrow$ lexp lexp-seq'	lexp-seq' $\rightarrow$ lexp lexp-seq'	lexp-seq' $\rightarrow$ $\epsilon$	

e) 对输入串(a (b (2)) (c))给出相应得 LL(1)分析程序的动作

记 lexp 为 E, list 为 L, atom 为 A, lexp-seq 为 S, lexp-seq' 为 S', number 为 num, identifier 为 id, 则分析过程如下:

栈	输入	动作
\$ E	(a(b(2))(c))\$	E $\rightarrow$ L
\$ L	(a(b(2))(c))\$	L $\rightarrow$ (S)
\$ ) S(	(a(b(2))(c))\$	match
\$ ) S	a(b(2))(c))\$	S $\rightarrow$ E S'
\$ ) S'E	a(b(2))(c))\$	E $\rightarrow$ A
\$ ) S'A	a(b(2))(c))\$	A $\rightarrow$ id
\$ ) S'id	a(b(2))(c))\$	match
\$ ) S'	(b(2))(c))\$	S' $\rightarrow$ E S'
\$ ) S'E	(b(2))(c))\$	E $\rightarrow$ L
\$ ) S'L	(b(2))(c))\$	L $\rightarrow$ (S)
\$ ) S') S (	(b(2))(c))\$	match
\$ ) S') S	b(2))(c))\$	S $\rightarrow$ E S'
\$ ) S') S'E	b(2))(c))\$	E $\rightarrow$ A
\$ ) S') S'A	b(2))(c))\$	A $\rightarrow$ id
\$ ) S') S'id	b(2))(c))\$	match
\$ ) S') S'	(2))(c))\$	S' $\rightarrow$ E S'
\$ ) S') S'E	(2))(c))\$	E $\rightarrow$ L
\$ ) S') S'L	(2))(c))\$	L $\rightarrow$ (S)
\$ ) S') S') S (	(2))(c))\$	match
\$ ) S') S') S	2))(c))\$	S $\rightarrow$ E S'
\$ ) S') S') S'E	2))(c))\$	E $\rightarrow$ A
\$ ) S') S') S'A	2))(c))\$	A $\rightarrow$ num
\$ ) S') S') S'num	2))(c))\$	match
\$ ) S') S') S'	))(c))\$	S' $\rightarrow$ $\epsilon$
\$ ) S') S')	))(c))\$	match

\$ ) S' ) S'	)(c))\$	$S' \rightarrow \epsilon$
\$ ) S')	)(c))\$	match
\$ ) S'	(c))\$	$S' \rightarrow E S$
\$ ) S'E	(c))\$	$E \rightarrow L$
\$ ) S'L	(c))\$	$L \rightarrow (S)$
\$ ) S')S(	(c))\$	match
\$ ) S')S	c))\$	$S \rightarrow E S'$
\$ ) S')S'E	c))\$	$E \rightarrow A$
\$ ) S')S'A	c))\$	$A \rightarrow id$
\$ ) S')S'id	c))\$	match
\$ ) S')S'	))\$	$S' \rightarrow \epsilon$
\$ ) S')	))\$	match
\$ ) S'	)\$	$S' \rightarrow \epsilon$
\$ )	)\$	Match
\$	\$	Accept

#### 练习 4.5.2

- 4.5.2 对于文法  $S \rightarrow S S + \mid S S * \mid a$  和下面的最右句型，指出其归约时使用的到句柄

1.  $SS+$
2.  $SS+$
3. 第一个  $a$

- ①  $SSS + a * +$
- ②  $SS + a * a +$
- ③  $aaa * a + +$

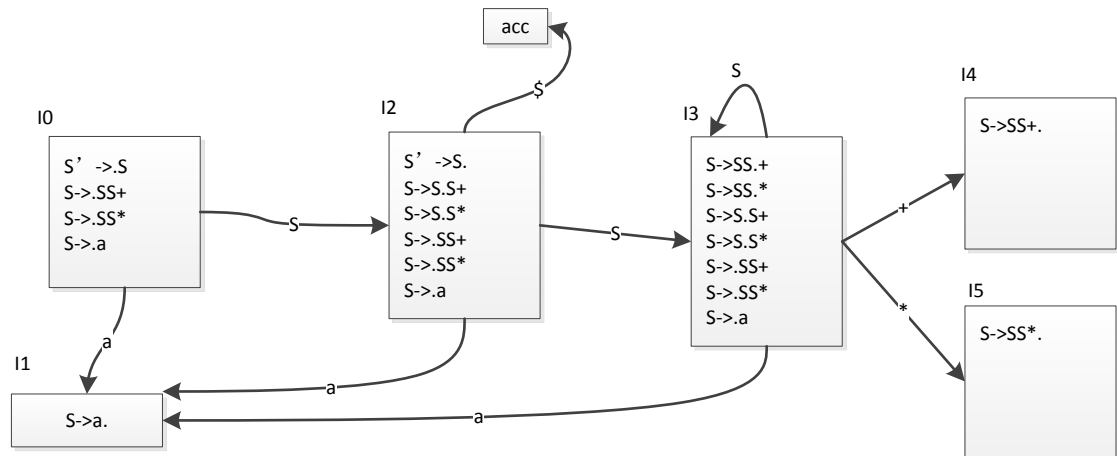
#### 练习 4.6.2

- 4.6.2 对于文法  $S \rightarrow S S + \mid S S * \mid a$ ，增广该文法构造SLR项目集。计算这些项目集的GOTO函数，给出这个文法的语法分析表。这个文法是SLR文法吗？

增广文法如下：

- 1)  $S' \rightarrow S$
- 2)  $S \rightarrow SS+$
- 3)  $S \rightarrow SS*$
- 4)  $S \rightarrow a$

GOTO 函数见下图：



$FOLLOW(S) = \{ a, \$, +, * \}$

语法分析表如下：

状态	ACTION				GOTO
	a	+	*	\$	S
0	S1				2
1	R4	R4	R4	R4	
2	S1			acc	3
3	S1	S4	S5		3
4	R2	R2	R2	R2	
5	R3	R3	R3	R3	

因为没有冲突，所以是 SLR 文法。

#### 练习 4.7.1

- 4.7.1 对于文法  $S \rightarrow S S + \mid S S * \mid a$ ，构造

增广文法如下：

- 1)  $S' \rightarrow S$
- 2)  $S \rightarrow SS+$
- 3)  $S \rightarrow SS*$
- 4)  $S \rightarrow a$

- ① 规范 LR 项目集族，并构建其语法分析表
- ② LALR 项目集族

1. 正规 LR 项目集族如下：

$I_0$

$S' \rightarrow \cdot S, \$$

$S \rightarrow \cdot SS+, \$/a$

$S \rightarrow \cdot SS*, \$/a$

$S \rightarrow \cdot a, \$/a$

$I_1$

$S \rightarrow a \cdot, \$/a$

$I_2$

$S' \rightarrow S., \$$   
 $S \rightarrow S.S+, \$/a$   
 $S \rightarrow S.S*, \$/a$   
 $S \rightarrow .SS+, +/* /a$   
 $S \rightarrow .SS*, +/* /a$   
 $S \rightarrow .a, +/* /a$

**I<sub>3</sub>**  
 $S \rightarrow a., +/* /a$

**I<sub>4</sub>**  
 $S \rightarrow SS., \$/a$   
 $S \rightarrow SS.*, \$/a$   
 $S \rightarrow S.S+, +/* /a$   
 $S \rightarrow S.S*, +/* /a$   
 $S \rightarrow .SS+, +/* /a$   
 $S \rightarrow .SS*, +/* /a$   
 $S \rightarrow .a, +/* /a$

**I<sub>5</sub>**  
 $S \rightarrow SS+., \$/a$

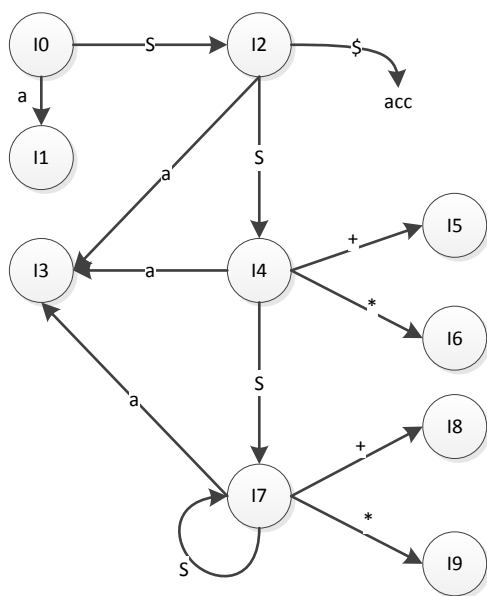
**I<sub>6</sub>**  
 $S \rightarrow SS*., \$/a$

**I<sub>7</sub>**  
 $S \rightarrow SS., +/* /a$   
 $S \rightarrow SS.*, +/* /a$   
 $S \rightarrow S.S+, +/* /a$   
 $S \rightarrow S.S*, +/* /a$   
 $S \rightarrow .SS+, +/* /a$   
 $S \rightarrow .SS*, +/* /a$   
 $S \rightarrow .a, +/* /a$

**I<sub>8</sub>**  
 $S \rightarrow SS+., +/* /a$

**I<sub>9</sub>**  
 $S \rightarrow SS*., +/* /a$

状态转换图如下：



语法分析表如下：

状态	ACTION				GOTO
	a	+	*	\$	S
0	S1				2
1	R4			R4	
2	S3			acc	4
3	R4	R4	R4		
4	S3	S5	S6		7
5	R2			R2	
6	R3			R3	
7	S3	S8	S9		7
8	R2	R2	R2		
9	R3	R3	R3		

2. 归并得到 LALR 项目集族如下：

**I0**

$S' \rightarrow .S, \$$

$S \rightarrow .SS+, \$/a$

$S \rightarrow .SS*, \$/a$

$S \rightarrow .a, \$/a$

**I13**

$S \rightarrow a., \$/+/* /a$

**I2**

$S' \rightarrow S., \$$

$S \rightarrow S.S+, \$/a$



$S \rightarrow S.S^*, \$ / a$   
 $S \rightarrow .SS+, + / * / a$   
 $S \rightarrow .SS^*, + / * / a$   
 $S \rightarrow .a, + / * / a$

#### I47

$S \rightarrow SS+.+, + / * / a / \$$   
 $S \rightarrow SS.*+, + / * / a / \$$   
 $S \rightarrow S.S+, + / * / a$   
 $S \rightarrow S.S^*, + / * / a$   
 $S \rightarrow .SS+, + / * / a$   
 $S \rightarrow .SS^*, + / * / a$   
 $S \rightarrow .a, + / * / a$

#### I58

$S \rightarrow SS+.+, + / * / a / \$$

#### I69

$S \rightarrow SS*., + / * / a / \$$