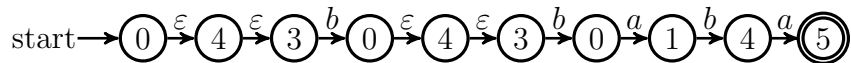


武汉大学计算机学院2010-2011学年第一学期
2008级《编译原理》参考答案

一、 (1)



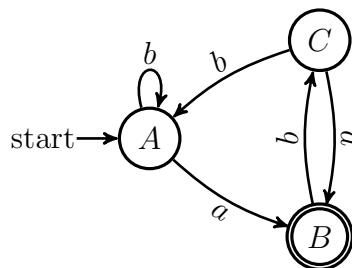
(2)

$$A = \{0, 3, 4\}$$

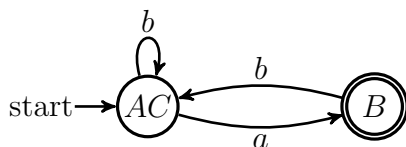
$$B = \{1, 2, 5\}$$

$$C = \{1, 3, 4\}$$

状态转换图为：



(3) 最小状态自动机如下：



(4) 以 a 结尾，且没有连续的 a .

二、 (1) 最左推导如下：

$$\begin{aligned}
 S &\xRightarrow{lm} aAaBb \\
 &\xRightarrow{lm} aaAaBb \\
 &\xRightarrow{lm} aaaBb \\
 &\xRightarrow{lm} aaabBb \\
 &\xRightarrow{lm} aaabb
 \end{aligned}$$

(2) $\text{First}(S) = \{a, b\}$; $\text{First}(A) = \{\varepsilon, a\}$; $\text{First}(B) = \{\varepsilon, b\}$.
 $\text{Follow}(S) = \{\$ \}$; $\text{Follow}(A) = \{a\}$; $\text{Follow}(B) = \{b\}$.

(3)

	a	b	$\$$
S	$S \rightarrow aAaBb$	$S \rightarrow bBbAa$	
A	$A \rightarrow aA; \quad A \rightarrow \varepsilon$		
B		$b \rightarrow bB; \quad B \rightarrow \varepsilon$	

(4) $r = aaa^*bb^*|bbb^*aa^*$ 或 $r = aa^*ab^*b|bb^*ba^*a$.

(5) 与 $G(S)$ 等价的 LL(1) 文法:

$$\begin{aligned} S &\rightarrow A \mid D \\ A &\rightarrow aaB \\ B &\rightarrow aB \mid bC \\ C &\rightarrow bC \mid \varepsilon \\ D &\rightarrow bbE \\ E &\rightarrow bE \mid aF \\ F &\rightarrow aF \mid \varepsilon \end{aligned}$$

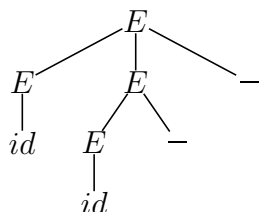
或与 $G(S)$ 等价的 LL(1) 文法:

$$\begin{aligned} S &\rightarrow aaAbB \mid bbBaA \\ A &\rightarrow aA \mid \varepsilon \\ B &\rightarrow bB \mid \varepsilon \end{aligned}$$

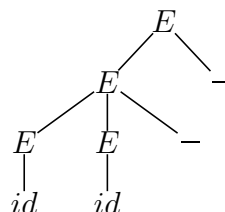
(6) SLR(1) 分析器在移进 a 后进入状态 $\{S \rightarrow a \bullet AaBb, A \rightarrow \bullet aA, A \rightarrow \bullet\}$, 面对输入 a 可移进到状态 $\{A \rightarrow a \bullet A, A \rightarrow \bullet aA, A \rightarrow \bullet\}$; 而 $a \in \text{Follow}(A)$, 所以也可以用产生式 $A \rightarrow \varepsilon$ 归约, 即有移进/归约冲突。

三、(1) “ $id\ id\ -\ -$ ” 的两颗不同的语法树:

语法树1:



语法树2:



(2)

$$\begin{aligned} E &\rightarrow E - \mid M \\ M &\rightarrow MM - \mid id \end{aligned}$$

四、(1) 识别活前缀的自动机在吃进 $EEE+$ 之后到达状态 I_4 , 不能再接受任何符号, 因此 $EEE + E$ 不是活前缀; 而识别活前缀的自动机在吃进 $EEEE$ 之后到达状态 I_2 , 所对应的 LR(0) 项目集即是其有效项目集:

$$\begin{aligned} &\overline{\{E \rightarrow EE \bullet +, E \rightarrow EE \bullet *, E \rightarrow E \bullet E +, E \rightarrow E \bullet E *\}} \\ = &\{E \rightarrow EE \bullet +, E \rightarrow EE \bullet *, E \rightarrow E \bullet E +, E \rightarrow E \bullet E *, \\ &E \rightarrow \bullet EE +, E \rightarrow \bullet EE *, E \rightarrow \bullet id\} \end{aligned}$$

(2) $\text{First}(E) = \{id\}$, $\text{Follow}(E) = \{+, *, id, \$\}$.

(3) SLR分析表如下所示:

	action				goto
状态	*	+	<i>id</i>	\$	<i>E</i>
0			s5		1
1			s5	acc	2
2	s3	s4	s5		2
3	r2	r2	r2	r2	
4	r1	r1	r1	r1	
5	r3	r3	r3	r3	

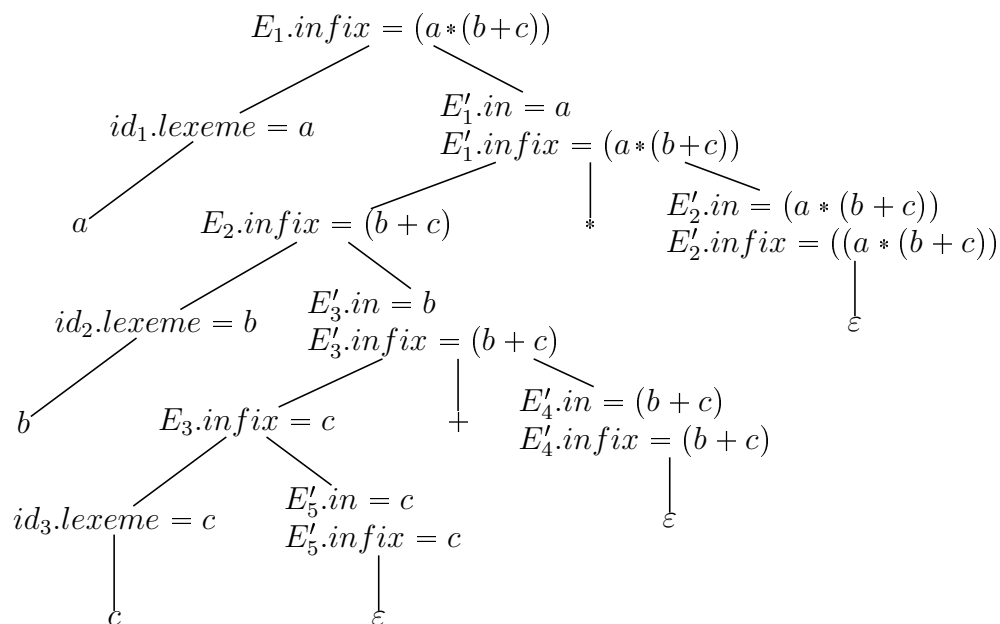
(4) “*id id + id**”的分析过程如下所示:

剩余串	分析栈	分析动作
<i>id id + id * \$</i>	0	shift
<i>id + id * \$</i>	0id5	reduce $E \rightarrow id$
<i>id + id * \$</i>	0E1	shift
<i>+id * \$</i>	0E1id5	reduce $E \rightarrow id$
<i>+id * \$</i>	0E1E2	shift
<i>id * \$</i>	0E1E2 + 4	reduce $E \rightarrow EE +$
<i>id * \$</i>	0E1	shift
<i>*\$</i>	0E1id5	reduce $E \rightarrow id$
<i>*\$</i>	0E1E2	shift
<i>\$</i>	0E1E2 * 3	reduce $E \rightarrow EE*$
<i>\$</i>	0E1	分析成功

五、 (1)

产生式	语义规则
$E \rightarrow id E'$	$E.infix = E'.infix$ $E'.in = id.lexeme$
$E' \rightarrow E + E'_1$	$E'.infix = E'_1.infix$ $E'_1.in = "(" + E'.in + "+" + E.infix + ")"$
$E' \rightarrow E * E'_1$	$E'.infix = E'_1.infix$ $E'_1.in = "(" + E'.in + "*" + E.infix + ")"$
$E' \rightarrow \varepsilon$	$E'.infix = E'.in$

(2) “*abc + **”的附注语法树:



六、

```

17:  if (a > b) goto 10          goto 15
      goto 19                  13:  goto 19
10:  if (c > d) goto 18          15:  if (m > n) goto 14
      goto 19                  goto 16
18:  if (e > f) goto 11        14:  goto 17
      goto 12                  16:  t1 := y + 2
11:  if (i > j) goto 12        y := t1
      goto 13                  goto 17
12:  t0 := x + 1              19:
      x := t0

```

七、程序在调用foo()后的内存格局如下：

address	memory	note
	
x	0	← i
x-4	20	← a[1]
x-8	10	← a[0]
	

当for语句执行到i=2时，循环条件满足，这时数组已经越界，a[2]实际上是i的地址。语句“a[i] -= 3;”把i修改为-1，当程序执行到i++时，i又被修改为0。循环条件再次成立，如此反复，循环不能退出。