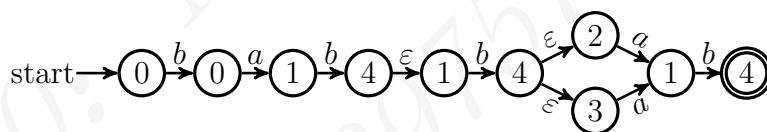


武汉大学计算机学院2012-2013学年第一学期  
2010级《编译原理》参考答案

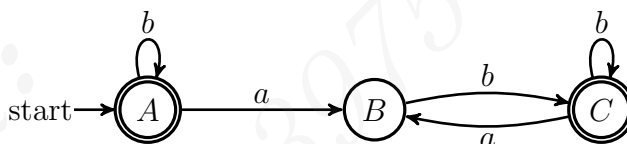
一、 (1)



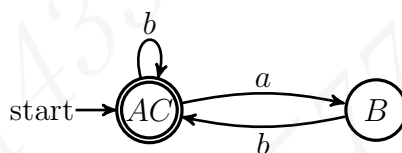
(2)

$$\begin{aligned} A &= \{0, 2, 3\} \\ B &= \{1\} \\ C &= \{1, 2, 3, 4, 5\} \end{aligned}$$

状态转换图为:



(3) 最小DFA如下所示:



(4) 空串或以 $b$ 结尾且没有连续的 $a$ .

(5)  $r = (b \mid ab)^*$ .

二、 (1) 最左推导如下:

$$\begin{aligned} E &\xRightarrow{lm} E[E] \\ &\xRightarrow{lm} id[E] \\ &\xRightarrow{lm} id[*E] \\ &\xRightarrow{lm} id[*id] \end{aligned}$$

(2) 消除左递归后的文法如下:

$$\begin{aligned} E &\rightarrow *ET \mid idT \\ T &\rightarrow [E]T \mid \varepsilon \end{aligned}$$

(3)  $First(E) = \{id, *\}; First(T) = \{[, \varepsilon\}.$   
 $Follow(E) = Follow(T) = \{[, ], \$\}.$

(4) LL(1)分析表如下所示:

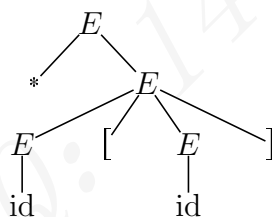
	id	*	[	]	\$
$E$	$E \rightarrow idT$	$E \rightarrow *ET$			
$T$			$E \rightarrow [E]T, E \rightarrow \varepsilon$	$E \rightarrow \varepsilon$	$E \rightarrow \varepsilon$

(5) 语句“id[\*id]”的分析过程如下所示:

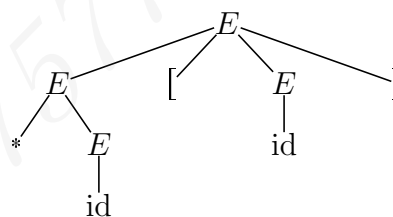
剩余串	分析栈	分析动作
id[*id]\$	$E\$$	$E \rightarrow idT$
id[*id]\$	idT\$	match-advance
[*id]\$	T\$	$T \rightarrow [E]T$
[*id]\$	[E]T\$	match-advance
*id]\$	E]T\$	$E \rightarrow *ET$
*id]\$	*ET]T\$	match-advance
id]\$	ET]T\$	$E \rightarrow idT$
id]\$	idTT]T\$	match-advance
]\$	TT]T\$	$T \rightarrow \varepsilon$
]\$	T]T\$	$T \rightarrow \varepsilon$
]\$	]T\$	match-advance
\$	T\$	$T \rightarrow \varepsilon$
\$	\$	分析成功

三、(1) “id[id]”的两颗不同的语法树:

语法树1:



语法树2:



(2) 无二义文法:

$$\begin{aligned}
 E &\rightarrow E + T \mid T \\
 T &\rightarrow *T \mid F \\
 F &\rightarrow F[E] \mid id
 \end{aligned}$$

四、(1) 识别活前缀的自动机在吃进  $E[**$  之后到达状态  $I_1$ , 其有效项目集为

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

识别活前缀的自动机在吃进  $*E[*E[$  之后到达状态  $I_5$ , 其有效项目集为:

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

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

(3) SLR分析表如下所示:

状态	action					goto
	id	*	[	]	\$	$E$
0	s3	s1				2
1	s3	s1				4
2			s5		acc	
3			r3	r3	r3	
4			s5	r1	r1	
5	s3	s1				6
6			s5	s7		
7			r2	r2	r2	

(4) “\*id[id]”的分析过程如下所示:

剩余串	分析栈	分析动作
*id[id]\$	0	shift
id[id]\$	0*1	shift
[id]\$	0*1id3	reduce $E \rightarrow \text{id}$
[id]\$	0*1E4	shift
id]\$	0*1E4[5	shift
]\$	0*1E4[5id3	reduce $E \rightarrow \text{id}$
]\$	0*1E4[5E6	shift
\$	0*1E4[5E6]7	reduce $E \rightarrow E[E]$
\$	0*1E4	reduce $E \rightarrow *E$
\$	0E2	分析成功

五、 (1)

产生式	语义规则
$E \rightarrow E_1 + E_2$	<pre> if <math>E_1.\text{type} == (\text{ptr}(T), 4)</math> and <math>E_2.\text{type} == (\text{ptr}(T_1), 4)</math> then error("pointer must operate with int") if <math>E_1.\text{type} == (\text{ptr}(T), 4)</math> then     <math>E.\text{type} = E_1.\text{type}</math>     <math>t = \text{newtemp}()</math>     <math>E.\text{temp} = \text{newtemp}()</math>     emit(<math>t + "=" + E_2.\text{temp} + "*" + \text{getsize}(T)</math>)     emit(<math>E.\text{temp} + "=" + E_1.\text{temp} + "+" + t</math>) else if <math>E_2.\text{type} == (\text{ptr}(T_1), 4)</math> then     <math>E.\text{type} = E_2.\text{type}</math>     <math>t = \text{newtemp}()</math>     <math>E.\text{temp} = \text{newtemp}()</math>     emit(<math>t + "=" + E_1.\text{temp} + "*" + \text{getsize}(T_1)</math>)     emit(<math>E.\text{temp} + "=" + E_2.\text{temp} + "+" + t</math>) else</pre>

	$E.type = (int, 4)$ $E.temp = newtemp()$ $emit(E.temp + "=" + E_1.temp + "+" + E_2.temp)$
$E \rightarrow *E_1$	$if\ E_1.type \neq (ptr(T), 4)$ <b>then</b> $error("*\text{ operates with no pointer}")$ $if\ E_1.type == (ptr(T), 4)\ \text{and}\ T == (array(T_1), n)$ <b>then</b> $E.type = (ptr(T_1), 4); E.temp = E_1.temp$ <b>else</b> $E.type = T; E.temp = newtemp()$ $emit(E.temp + "*" + E_1.temp)$
$E \rightarrow id$	$E.temp = id.lexeme$ $T = getsymb(id.lexeme)$ $if\ T == (char, 1)\ \text{then}\ E.type = (int, 4)$ $if\ T == (array(T_1), n)\ \text{then}\ E.type = (ptr(T_1), 4)$ <b>else</b> $E.type = T$

(2) 变量a的类型表达式为:

$array(array(array(ptr((int, 4), 4), 20), 80), 240)$

表达式“ $*a[x][y][z]$ ”为三地址码:

$t0 = x * 80$ $t1 = a + t0$ $t2 = y * 20$ $t3 = t1 + t2$	$t4 = z * 4$ $t5 = t3 + t4$ $t6 = *t5$
---	--

六、

<pre> 11: t0 := x + 1     x := t0     if (a&gt;b) goto 12     if (c&gt;d) goto 12     t1 := x + 2 </pre>	<pre>         x := t1 12: if (e&gt;f) goto 10         ifnot (g&gt;h) goto 13 10: ifnot (i&gt;k) goto 11 13: </pre>
--	--

七、程序1中，函数foo()的形参b的类型是指向长度为5的整型数组的指针，子表达式b[1](即\*(b + 1))的类型是指向长度为5的整形数组第一个元素的指针，因此其值还是(b + 1)，引用运算\*并没有起作用。所以foo(a) 能正确第访问a[1][3]。

但程序2中的子表达式b[1]的类型是指向整形的指针，因此\*(b + 1)中的引用运算\*将起作用。故对实参a, 子表达式\*((int \*\*) a + 1)的值为5, 类型是指针，这样b[1][3]将访问地址为5 + 3 \* 4上的值，超出了程序段的范围，因此报段错误。