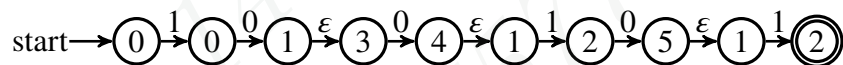


武汉大学计算机学院 2018 - 2019 学年第一学期
2016 级《编译原理》(期末考试参考答案 A)

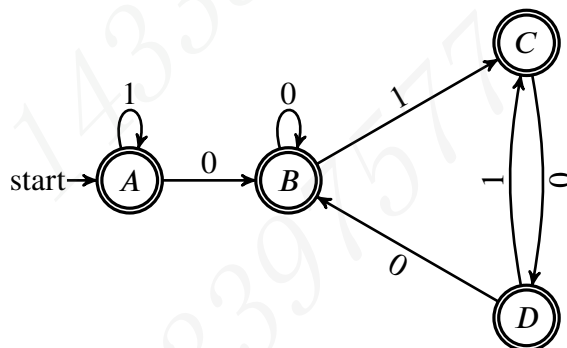
一、(1)



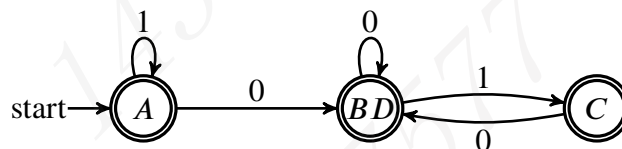
(2)

$$A = \{0, 3\}, B = \{1, 3, 4\}, C = \{2\}, D = \{1, 3, 4, 5\}.$$

状态转换图为:



(3) 最小 DFA 如下所示:



(4) 无 011 子串.

(5) $r = 1^*(0+1)^*0^*$.

二、(1) 语句 “[a; a[a]]” 的最左推导如下:

$$\begin{array}{ll}
 S & \xRightarrow{lm} [L] \\
 & \xRightarrow{lm} [L; L] \\
 & \xRightarrow{lm} [a; L] \\
 & \xRightarrow{lm} [a; [L]] \\
 & \xRightarrow{lm} [a; [a]]
 \end{array}$$

(2) 消除左递归后的文法如下:

$$\begin{array}{l}
 S \rightarrow [L] \\
 L \rightarrow SL' \mid aL' \\
 L' \rightarrow ;LL' \mid \epsilon
 \end{array}$$

(3) $\text{First}(S) = \{ [\}; \text{First}(L) = \{ [, a \}; \text{First}(L') = \{ ;, \epsilon \}.$
 $\text{Follow}(S) = \{ ;,], \$ \}; \text{Follow}(L) = \text{Follow}(L') = \{ ;,] \}.$

(4) LL(1) 分析表如下所示

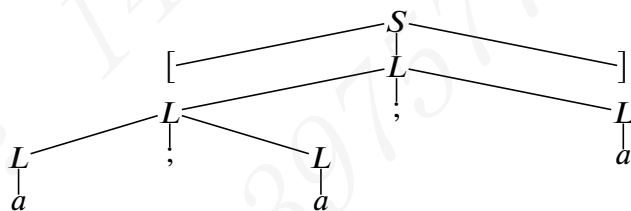
	a	$;$	$[$	$]$	$\$$
S			$S \rightarrow [L]$		
L	$L \rightarrow aL'$		$L \rightarrow SL'$		
L'		$L' \rightarrow ;LL' \mid \epsilon$		$L' \rightarrow \epsilon$	

(5) 语句 “[a;a]” 的分析过程如下所示：

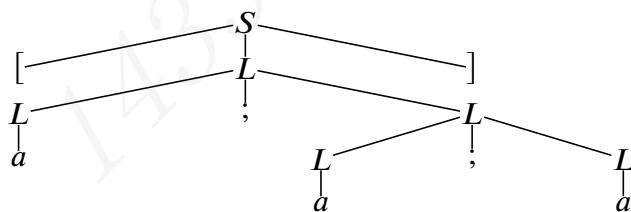
剩余串	分析栈	分析动作
[a;a]\$	$S\$$	$S \rightarrow [L]$
[a;a]\$	$[L]\$$	match-advance
a;a]\$	$L]\$$	$L \rightarrow aL'$
a;a]\$	$aL']\$$	match-advance
;a]\$	$L']\$$	$L' \rightarrow ;LL'$
;a]\$	$;LL']\$$	match-advance
a]\$	$LL']\$$	$L \rightarrow aL'$
a]\$	$aL'L']\$$	match-advance
]\$	$L'L']\$$	$L' \rightarrow \epsilon$
]\$	$L']\$$	$L' \rightarrow \epsilon$
]\$	$]\$$	match-advance
\$	$\$$	分析成功

三、 (1) 语句 “a[a,a,a]” 的两颗不同的语法树为：

语法树 1:



语法树 2:



(2) 无二义文法:

$$\begin{aligned}
 S &\rightarrow [L] \\
 L &\rightarrow S;L \mid a;L \mid S \mid a
 \end{aligned}$$

四、 (1) 状态 I_6 的 LR(0) 项目集为

$$\overline{\{L \rightarrow L; \bullet L\}} \\ = \{L \rightarrow L; \bullet L, L \rightarrow \bullet L; L, L \rightarrow \bullet S, L \rightarrow \bullet a, S \rightarrow \bullet [L]\}.$$

(2) 识别活前缀的自动在吃进 $[(L;)^*a$ 进入状态 I_5 , 而状态 I_5 无] 出边, 因此不是活前缀.

(3) $\text{Follow}(S) = \{ \$, ;,] \}, \text{Follow}(L) = \{ ;,] \}.$

状态 I_8 有移进/归约冲突, 右结合选移进. 故 SLR 分析表如下所示:

状态	action					goto	
	;	[]	a	\$	L	S
0		s2					1
1					acc		
2		s2		s5		3	4
3	s6		s7				
4	r3		r3				
5	r4		r4				
6		s2		s5		8	4
7	r1		r1		r1		
8	s6		r2				

(4) 语句 “[a;a]” 的分析过程如下所示:

剩余串	分析栈	分析动作
[a;a]\$	0	shift
a;a]\$	0[2	shift
;a]\$	0[2a5	reduce $L \rightarrow a$
;a]\$	0[2L3	shift
a]\$	0[2L3;6	shift
]\$	0[2L3;6a5	reduce $L \rightarrow a$
]\$	0[2L3;6L8	reduce $L \rightarrow L; L$
]\$	0[2L3	shift
\$	0[2L3]7	reduce $S \rightarrow [L]$
\$	0S1	分析成功

五、 (1)

产生式	语义规则
$S \rightarrow [L]$	if ($L.is_list$) then $S.tree = L.tree$ else $S.tree = "\Lambda(" + L.tree, \perp)"$
$L \rightarrow L_1; L_2$	if ($L_2.is_list$) then $L.tree = "\Lambda(" + L_1.tree + ", " + L_2.tree + ")"$ else $L.tree = "\Lambda(" + L_1.tree + ", \Lambda(" + L_2.tree + ", \perp)"$ $L.is_list = True$
$L \rightarrow S$	$L.tree = S.tree$ $L.is_list = False$
$L \rightarrow a$	$L.tree = a.lexval$ $L.is_list = False$

(2) $\Lambda(\Lambda(\Lambda(a, \perp), \perp),$
 $\Lambda(b, \Lambda(\Lambda(c, \perp), \perp)))$

六、

```

L1: if (a > b) goto L2      |      if (i > j) goto L3
    if (c > d) goto L3      | L0: t1 := y + 2
    ifnot (e > f) goto L3   |      x := t1
L2: t0 := x + 1            |      goto L1
    x := t0                 | L3:
    if (g > h) goto L0      |

```

七、 传值是 C 语言唯一的参数传递方式，main() 中调用 foo(p) 并未能对 p 初始化，因此输出乱码。