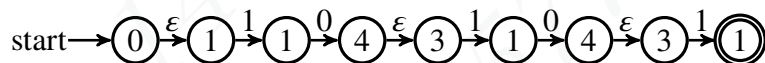


武汉大学计算机学院 2018 - 2019 学年第一学期
2016 级弘毅班《编译原理》(期末考试参考答案 A)

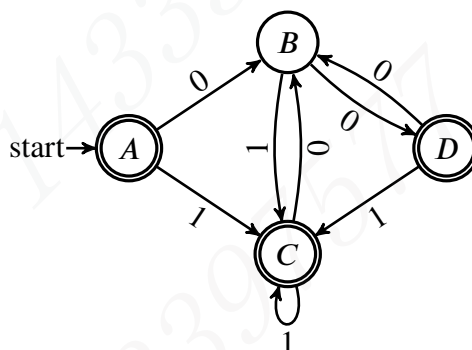
一、(1)



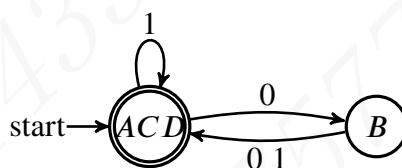
(2)

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

状态转换图为:



(3) 最小 DFA 如下所示:



(4) 连续的 0 出现偶数次, 奇数次出现必须接 1.

(5) $r = (00 \mid 01 \mid 1)^*$.

二、(1) 语句 “[a; a[a]]” 的最左推导如下:

$$\begin{array}{ll} L \xRightarrow{lm} \lambda a.L & \xRightarrow{lm} \lambda a.aL \\ \xRightarrow{lm} \lambda a.LL & \xRightarrow{lm} \lambda a.aa \end{array}$$

(2) 消除左递归后的文法如下:

$$\begin{array}{l} L \rightarrow \lambda a.LL' \mid aL' \\ L' \rightarrow LL' \mid \epsilon \end{array}$$

(3) $\text{First}(L) = \{a, \lambda\}$; $\text{First}(L') = \{a, \lambda, \epsilon\}$.

$\text{Follow}(L) = \text{Follow}(L') = \{a, \lambda, \$\}$.

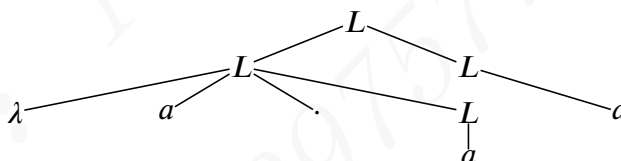
(4) LL(1) 分析表如下所示

	a	λ	.	\$
L	$L \rightarrow aL'$	$L \rightarrow \lambda a.LL'$		
L'	$L' \rightarrow LL' \mid \epsilon$	$L' \rightarrow LL' \mid \epsilon$		$L' \rightarrow \epsilon$

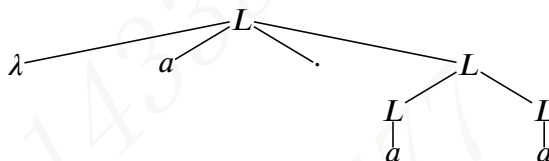
(5) 语句“ $\lambda a.a$ ”的分析过程如下所示:

剩余串	分析栈	分析动作
$\lambda a.a\$$	$L\$$	$L \rightarrow \lambda a.LL'$
$\lambda a.a\$$	$\lambda a.LL'\$$	match-advance
$a.a\$$	$a.LL'\$$	match-advance
$.a\$$	$.LL'\$$	match-advance
$a\$$	$LL'\$$	$L \rightarrow aL'$
$a\$$	$LL'L'\$$	match-advance
$\$$	$L'L'\$$	$L' \rightarrow \epsilon$
$\$$	$L'\$$	$L' \rightarrow \epsilon$
$\$$	$\$$	分析成功

三、 (1) 语句“ $\lambda a.aa$ ”的两颗不同的语法树为:
语法树 1:



语法树 2:



(2) 无二义文法:

$$\begin{aligned} L &\rightarrow \lambda a.L \mid A \\ A &\rightarrow Aa \mid a \end{aligned}$$

四、 (1) 状态 I_6 的 LR(0) 项目集为

$$\begin{aligned} &\overline{\{L \rightarrow L \bullet L, L \rightarrow \lambda a.L \bullet\}} \\ &= \{L \rightarrow L \bullet L, L \rightarrow \lambda a.L \bullet, L \rightarrow \bullet a, L \rightarrow \bullet \lambda a.L, L \rightarrow \bullet LL\}. \end{aligned}$$

- (2) 识别活前缀的自动在吃进 $(\lambda a.)^*L^*a$ 进入状态 I_2 , 而状态 I_2 无 L 出边, 因此仅 $(\lambda a.)^*L^*a$ 为活前缀, $(\lambda a.)^*L^*aL^+$ 不是活前缀.
- (3) $\text{Follow}(L) = \{\$, \lambda, a\}$, $\text{Follow}(L) = \{;, \text{]}\}$.
状态 I_4 和状态 I_7 有移进/归约冲突, 状态 I_4 面对 a 和 λ 选归约; 状态 I_7 选移进. 故 SLR 分析表如下所示:

	action				goto
状态	.	<i>a</i>	λ	\$	<i>L</i>
0		s2	s3		1
1		s2	s3	acc	4
2		r3	f3	r3	
3		s5			
4		r2	r2	r2	4
5	s6				
6		s2	s3		7
7		s2	s3	r1	4

(4) 语句“ $\lambda a.aa$ ”的分析过程如下所示：

剩余串	分析栈	分析动作
$\lambda a.aa\$$	0	shift
$a.aa\$$	0 λ 3	shift
$.aa\$$	0 λ 3 a 5	shift
$aa\$$	0 λ 3 a 5.6	shift
$a\$$	0 λ 3 a 5.6 a 2	reduce $L \rightarrow a$
$a\$$	0 λ 3 a 5.6 L 7	shift
$a\$$	0 λ 3 a 5.6 L 7 a 2	reduce $L \rightarrow a$
$a\$$	0 λ 3 a 5.6 L 7 L 4	reduce $L \rightarrow LL$
$a\$$	0 λ 3 a 5.6 L 7	reduce $L \rightarrow \lambda a.L$
$a\$$	0 L 1	分析成功

五、 (1)

产生式	语义规则
$L' \rightarrow L$	$L.is_abs = \text{False}$
$L \rightarrow \lambda a.L_1$	$L_1.is_abs = \text{True}$ if ($L_1.last_op = \text{App}$) then $L.code = [\text{PUT}(L_1.code ++ [\text{TAIL}])]$ else $L.code = [\text{PUT}(L_1.code ++ [\text{RET}])]$ $L.last_op = \text{Abs}$
$L \rightarrow L_1 L_2$	$L_1.is_abs = \text{False}$ $L_2.is_abs = \text{False}$ if ($L.is_abs$) then $L.code = L_1.code ++ L_2.code$ else $L.code = L_1.code ++ L_2.code ++ [\text{APPLY}]$ $L.last_op = \text{App}$
$L \rightarrow (L_1)$	$L_1.is_abs = L.is_abs$ $L.code = L_1.code$ $L.last_op = L_1.last_op$
$L \rightarrow a$	$L.code = [\text{GET } a.lexval]$ $L.last_op = \text{Atom}$

(2) $[\text{PUT}[\text{PUT}[\text{GET } x;\text{RET}];\text{GET } y;\text{TAIL}];\text{PUT}[\text{GET } y;\text{RET}];\text{APP}]$

六、

```

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

```

七、 计算求和函数，局部尾递归函数，优化为循环.