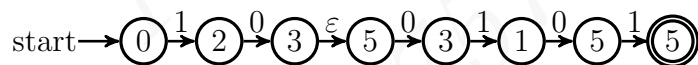
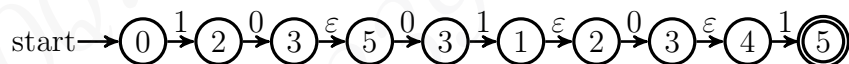


武汉大学计算机学院2008-2009学年第二学期  
2006级《编译原理》参考答案

一、 (1)



或



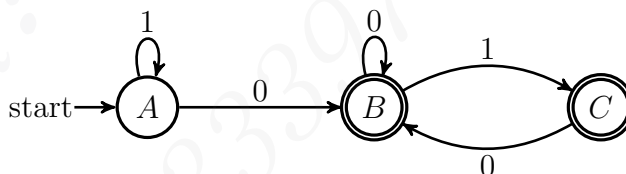
(2)

$$A = \{0, 1, 2\}$$

$$B = \{3, 4, 5\}$$

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

状态转换图为:



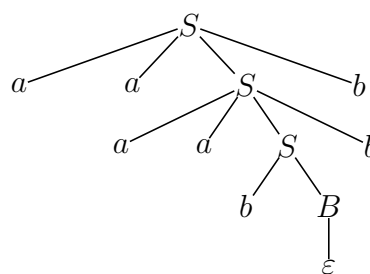
(3)  $1^*(01|0)^*(01|0)$  或  $1^*(01|0)^+$  或  $1^*0(10|0)^*(1|\varepsilon)$ .

(4) 以一个或多个连续的1或以0开始并不再有两个或多个连续的1为子串的非空字符串集合。  
或由0和1组成且至少含有一个0并不含“011”子串的非空字符串集合。

二、 (1) 最右推导如下:

$$\begin{aligned} S &\xRightarrow{rm} aaSb \\ &\xRightarrow{rm} aaaaSbb \\ &\xRightarrow{rm} aaaabBbb \\ &\xRightarrow{rm} aaaabbbb \end{aligned}$$

语法树:



(2)  $\{a^{2m}b^n \mid m, n \in \mathbb{N} \wedge 0 \leq 2m \leq 2n\}$ .

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

(4)

	$a$	$b$	$\$$
$S$	$S \rightarrow aaSb$	$S \rightarrow B$	$S \rightarrow B$
$B$		$B \rightarrow bB, B \rightarrow \varepsilon$	$B \rightarrow \varepsilon$

(5) 与 $G(S)$ 等价的LL(1)文法:

$$\begin{aligned} S &\rightarrow AB \\ A &\rightarrow aaAb \mid \varepsilon \\ B &\rightarrow bB \mid \varepsilon \end{aligned}$$

(6) 由上最右推导得知, LR分析器识别语句 $a^{2m}b^p b^m (p \geq 0)$ 的步骤应该是: 在移进 $2m$ 个 $a$ 之后, 先把前 $p$ 个多余的 $b$ 归约为 $B$ , 再将 $B$ 归约为 $S$ 得到活前缀 $a^{2m}S$ , 最后将剩余的 $m$ 个 $b$ 用产生式 $S \rightarrow aaSb$  逐个归约。由于多余的 $b$ 可以任意多, 且LR分析从左到右的扫描机制, 及只能向前查看固定次数的符号, 因此分析器无法知道有多少个 $b$ 是多余的, 从而无法解决何时停止将多余的 $b$ 归约为 $B$ 。故不是LR(k)文法。  
或者简答为: LR分析器由于无法知道有多少个 $b$ , 因此不能判断多少个 $b$ 归约为 $B$ , 多少个 $b$ 用于平衡 $a$ 。

三、(1) 面对输入“ $n + n + n$ ”有两个不同的最左推导。

推导1:

$$\begin{aligned} E &\xRightarrow{lm} E + E \\ &\xRightarrow{lm} n + E \\ &\xRightarrow{lm} n + E + E \\ &\xRightarrow{lm} n + n + E \\ &\xRightarrow{lm} n + n + n \end{aligned}$$

推导2:

$$\begin{aligned} E &\xRightarrow{lm} E + E \\ &\xRightarrow{lm} E + E + E \\ &\xRightarrow{lm} n + E + E \\ &\xRightarrow{lm} n + n + E \\ &\xRightarrow{lm} n + n + n \end{aligned}$$

(2)

$$\begin{aligned} E &\rightarrow T + E \mid T \\ T &\rightarrow \sqrt{T} \mid F \\ F &\rightarrow F^2 \mid n \end{aligned}$$

四、(1) 识别活前缀的自动机在吃进 $\sqrt{E + E +}$ 之后到达状态 $I_2$ , 因此它是活前缀, 其对应的有效项目集即是 $I_2$ 所对应的项目集:

$$\begin{aligned} \overline{\{E \rightarrow E + \bullet E\}} &= \{E \rightarrow E + \bullet E, E \rightarrow \bullet E + E, \\ &\quad E \rightarrow \bullet \sqrt{E}, E \rightarrow \bullet E^2, E \rightarrow \bullet n\} \end{aligned}$$

识别活前缀的自动机在吃进 $\sqrt{\sqrt{E + E^2}}$ 之后到达状态 $I_4$ , 不能再接受任何非终结符, 因此 $\sqrt{\sqrt{E + E^2}}$ 不是活前缀。

(2) 状态 $I_3$ 和 $I_7$ 面对 $+$ 有移进/归约冲突。

(3)  $\text{First}(E) = \{n, \sqrt{\quad}\}$ ,  $\text{Follow}(E) = \{+, ^2 \$\}$ , SLR分析表如下所示:

	action					goto
状态	+	$n$	$\sqrt{\quad}$	$^2$	$\$$	$E$
0		s5	s6			1
1	s2			s4	acc	
2		s5	s6			3
3	r1			s4	r1	
4	r3			r3	r3	
5	r4			r4	r4	
6		s5	s6			7
7	r2			s4	r2	

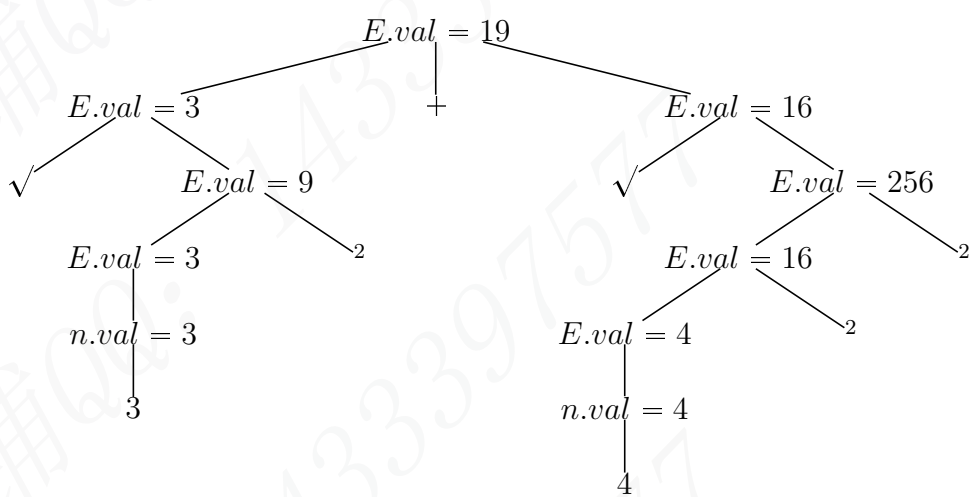
(4) “ $\sqrt{n + n^2}$ ”的分析过程如下所示:

剩余串	分析栈	分析动作
$\sqrt{n + n^2}\$$	0	shift
$n + n^2\$$	0 $\sqrt{6}$	shift
$+n^2\$$	0 $\sqrt{6}n5$	reduce $E \rightarrow n$
$+n^2\$$	0 $\sqrt{6}E7$	reduce $E \rightarrow \sqrt{E}$
$+n^2\$$	0E1	shift
$n^2\$$	0E1 + 2	shift
$^2\$$	0E1 + 2n5	reduce $E \rightarrow n$
$^2\$$	0E1 + 2E3	shift
$\$$	0E1 + 2E3 $^24$	reduce $E \rightarrow E^2$
$\$$	0E1 + 2E3	reduce $E \rightarrow E + E$
$\$$	0E1	分析成功

五、 (1)

产生式	语义规则
$E \rightarrow E_1 + E_2$	$E.val = E_1.val + E_2.val$
$E \rightarrow \sqrt{E_1}$	$E.val = \text{sqrt}(E_1.val)$
$E \rightarrow E_1^2$	$E.val = E_1.val * E_1.val$
$E \rightarrow n$	$E.val = n.val$

(2)  $\sqrt{3^2 + \sqrt{4^2}^2}$ 的附注语法树:



```

六、 16: if (a > b) goto 10      x := t0
      goto 18                  goto 15
10: if (c < d) goto 17          13: t1 := x + 2
      goto 18                  x := t1
17: if (c > 1) goto 12         15: if (x > 100) goto 14
      goto 11                  goto 16
11: if (d < 10) goto 12        14: goto 18
      goto 13                  goto 16
12: t0 := x + 1               18:

```

七、 程序在调用outputc()后的内存格局如下(little endian):

address	memory	note
	.....	
x	0	← a
x-1	0x30	
x-2	0x31	
x-3	0x32	
x-4	0x33	← b
x-5	0x30	
x-6	0x31	
x-7	0x32	
x-8	0x33	← 实参
x-12	ret add	
x-16	old fp	
x-20	x-8	← cp
	.....	

这时指针cp指向实参的首地址，语句“while (\*cp) printf("%c", \*cp++);”将以字节为单位并按内存地址增长方向连续输出内存对应的ASCII字符，直到内存值为0为止。故反向以字符方式两次输出整数0x30313233。