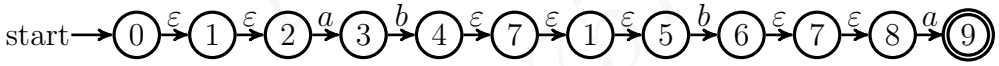
武汉大学计算机学院2007-2008学年第二学期
2005级《编译原理》参考答案

一、 (1)

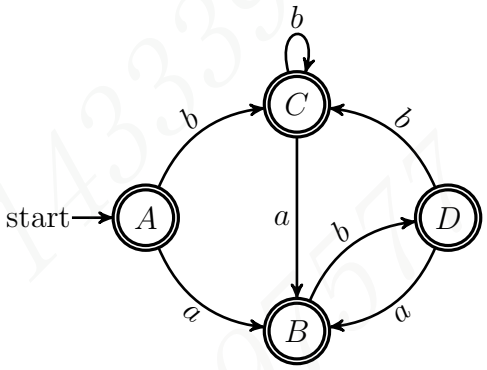


(2) $(ab|b)^*(a|\varepsilon)$

(3)

$$\begin{aligned} A &= \{0,1,2,5,8,9\} \\ B &= \{3,9\} \\ C &= \{1,2,5,6,7,8,9\} \\ D &= \{1,2,4,5,7,8,9\} \end{aligned}$$

状态转换图为：

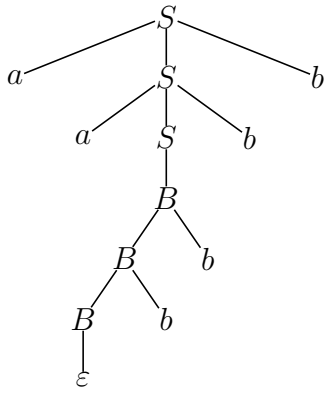


(4) 没有aa子串；或没有连续的a。

二、 (1) 最右推导如下：

$$\begin{aligned} S &\underset{rm}{\Longrightarrow} aSb \\ &\underset{rm}{\Longrightarrow} aaSbb \\ &\underset{rm}{\Longrightarrow} aaBbb \\ &\underset{rm}{\Longrightarrow} aaBbbb \\ &\underset{rm}{\Longrightarrow} aaBbbbb \\ &\underset{rm}{\Longrightarrow} aabbbb \end{aligned}$$

语法树：



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

(3) 由上最右推导得知，LR分析器识别语句$a^m b^p b^m (p \geqslant 0)$的步骤应该是：在移进$n$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$a^n S$，最后将剩余的同$a$的次数平衡的$m$个$b$用产生式 $S \to aSb$ 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规约为B，多少个b用于平衡a。

(4) 修改a和b匹配的机制即可，让在前面出现的b先与a匹配:

$$\begin{aligned} S &\to TB \\ T &\to aTb \mid \varepsilon \\ B &\to Bb \mid \varepsilon \end{aligned}$$

三、 (1) $\text{First}(S) = \{ a \}$; $\text{First}(T) = \{ +, \varepsilon \}$
$\text{Follow}(S) = \{ +, \$ \}$; $\text{Follow}(T) = \{ +, \$ \}$

(2)

	$+$	a	\$
S		$S \to aT$	
T	$T \to +ST \quad T \to \varepsilon$		$T \to \varepsilon$

这样面对栈顶符号T和输入单词+可选两个不同的产生展开，故不是LL(1)文法。

(3) 实际上上述文法是二义文法，生成的语言是仅有加法运算的表达式，其LL(1)文法如下：

$$\begin{aligned} S &\to aT \\ T &\to +aT \mid \varepsilon \end{aligned}$$

四、 (1) 面对输入“$e \mapsto ee$”有两个不同的最左推导。

推导1：

$$\begin{aligned} E &\underset{lm}{\Longrightarrow} EE \\ &\underset{lm}{\Longrightarrow} e \mapsto EE \\ &\underset{lm}{\Longrightarrow} e \mapsto eE \\ &\underset{lm}{\Longrightarrow} e \mapsto ee \end{aligned}$$

推导2：

$$\begin{aligned} E &\underset{lm}{\Longrightarrow} e \mapsto E \\ &\underset{lm}{\Longrightarrow} e \mapsto EE \\ &\underset{lm}{\Longrightarrow} e \mapsto eE \\ &\underset{lm}{\Longrightarrow} e \mapsto ee \end{aligned}$$

(2)

$$\begin{aligned} E &\rightarrow e \mapsto E \mid T \\ T &\rightarrow TF \mid F \\ F &\rightarrow e \mid (\, E\,) \end{aligned}$$

五、 (1) 识别活前缀的自动机在吃进 $(E((Ee \mapsto E$ 之后到达状态I_8，因此它是活前缀，其对应的有效项目集即是I_8所对应的项目集：

$\overline{\{E \rightarrow e \mapsto E\bullet,\ E \rightarrow E \bullet E\}} = \{\, E \rightarrow e \mapsto E\bullet,\ E \rightarrow E \bullet E,$
$E \rightarrow \bullet e \mapsto E,\ E \rightarrow \bullet EE,\ E \rightarrow \bullet(E),\ E \rightarrow \bullet e\}$

识别活前缀的自动机在吃进 $((e \mapsto E)$ 之后到达状态I_4, 不能再接受任何非终结符，因此 $((e \mapsto E)E$ 不是活前缀。

(2) 状态I_2和I_8有移进/归约冲突。

(3)

	action					goto
状态	$\mapsto$	(	)	e	\$	E
0		s3		s6		1
1		s3		s6	acc	2
2		r2	r2	r2	r2	2
3		s3		s6		4
4		s3	s5	s6		2
5		r3	r3	r3	r3	
6	s7	r4	r4	r4	r4	
7		s3		s6		8
8		s3	r1	s6	r1	2

(4)

剩余串	分析栈	分析动作
$e \mapsto ee\$$	0	shift
$\mapsto ee\$$	$0e6$	shift
$ee\$$	$0e6 \mapsto 7$	shift
$e\$$	$0e6 \mapsto 7e6$	reduce $E \to e$
$e\$$	$0e6 \mapsto 7E8$	shift
$\$$	$0e6 \mapsto 7E8e6$	reduce $E \to e$
$\$$	$0e6 \mapsto 7E8E2$	reduce $E \to EE$
$\$$	$0e6 \mapsto 7E8$	reduce $E \to e \mapsto E$
$\$$	$0S1$	分析成功

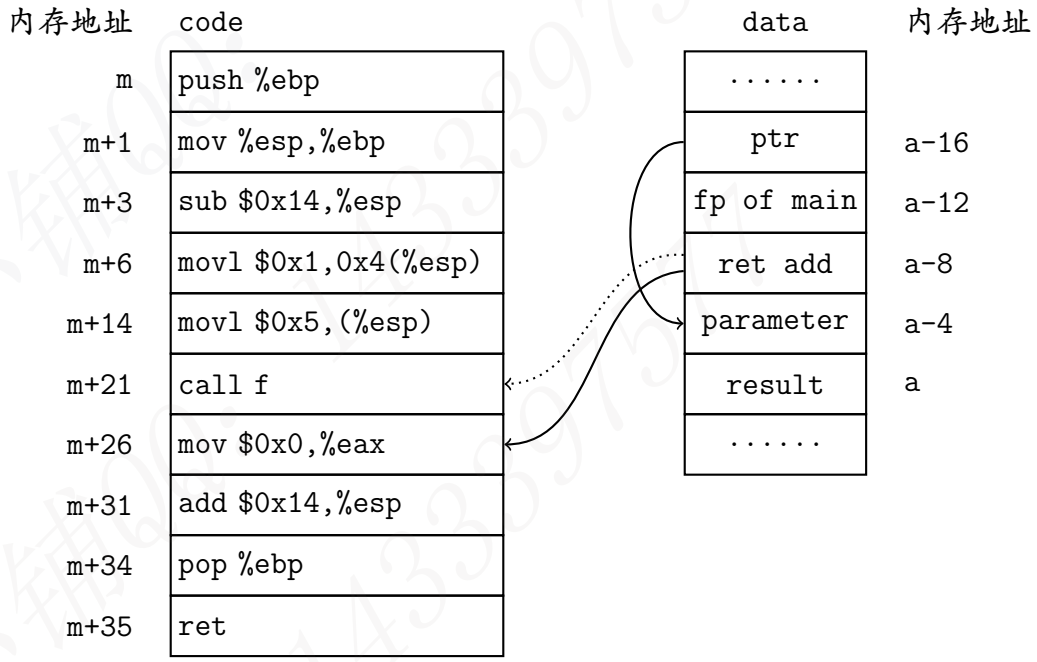
六、 (1)

产生式	语义规则
$P \to S$	$S.break = \varnothing$; $S.continue = \varnothing$
$S \to \textbf{if}\,(B)\ S_1$	$S_1.break = S.break$; $S_1.continue = S.continue$
$S \to \textbf{while}\ (B)\ S_1$	$begin = newlabel()$ $S_1.break = S.next$; $S_1.continue = begin$
$S \to S_1\ S_2$	$S_1.break = S.break$; $S_1.continue = S.continue$ $S_2.break = S.break$; $S_2.continue = S.continue$
$S \to \{\ S_1\ \}$	$S_1.break = S.break$; $S_1.continue = S.continue$
$S \to \textbf{break}$	**if** $(S.break == \varnothing)$ **then** $error$(“break not within while”) **else** gen('goto' $S.break$)
$S \to \textbf{continue}$	**if** $(S.continue == \varnothing)$ **then** $error$(“continue not within while”) **else** gen('goto' $S.continue$)

(2) 对应的三地址码：

```
l4: if (x > 0) goto l0          | l1: goto l4
    goto l6                     | l2: t1 := x * 2
l0: if (x < 100) goto l5        |     x := t1
    goto l6                     |     if (x = 100) goto l3
l5: t0 := x + 1                 |     goto l4
    x := t0                     | l3: goto l6
    if (x = 10) goto l1         |     goto l4
    goto l2                     | l6:
```

七、 程序在调用f()后，内存的格局如下图所示：



语句“*(ptr - 1) = *(ptr - 1) - 5;”把内存“ret add”的值修改成“call f”指令所在的地址，这样函数f()返回后将继续“call f” 直到从“if (parameter == 1)”为真出口返回。

或简答：语句“*(ptr - 1) = *(ptr - 1) - 5;”修改了返回地址。