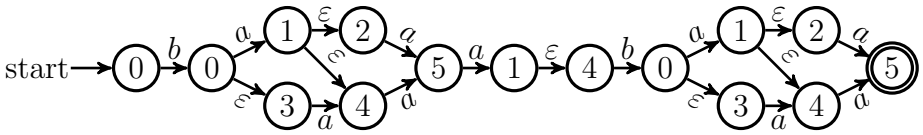
武汉大学计算机学院2009-2010学年第一学期
2007级《编译原理》参考答案

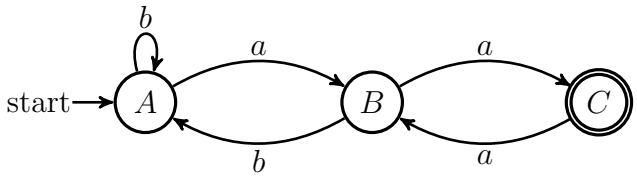
一、 (1)



(2)

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

状态转换图为：



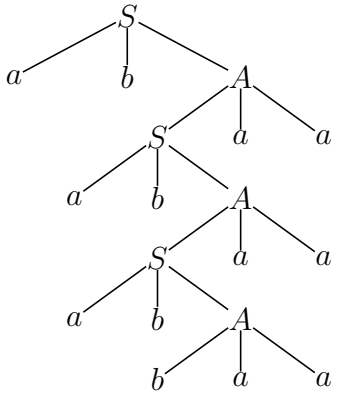
(3) $(b|a(aa)^*b)^*aa(aa)_*$ 或$b^*(aa|abb^*)^*aa$.

(4) 以偶数(非零)个a结尾，且连续的b之间及第一个b之前一定是奇数个a，.

二、 (1) 最右推导如下：

$$\begin{aligned} S &\underset{rm}{\Longrightarrow} abA \\ &\underset{rm}{\Longrightarrow} abSaa \\ &\underset{rm}{\Longrightarrow} ababAaa \\ &\underset{rm}{\Longrightarrow} ababSaaaa \\ &\underset{rm}{\Longrightarrow} abababAaaaa \\ &\underset{rm}{\Longrightarrow} abababbaaaaaa \end{aligned}$$

语法树：



(2) $\{\,(ab)^n ba^{2n} \mid n \in \mathbb{N}\,\}$.

(3) $\mathrm{First}(S) = \{\,a, b\,\}$; $\mathrm{First}(A) = \{\,a, b\,\}$.
$\mathrm{Follow}(S) = \{\,a, \$\,\}$; $\mathrm{Follow}(A) = \{\,a, \$\,\}$.

(4)

	a	b	\$
S	$S \to abA$		$S \to b$
A	$A \to Saa$	$A \to Saa \quad A \to bba$	

(5) 与$G(S)$等价的LL(1)文法：

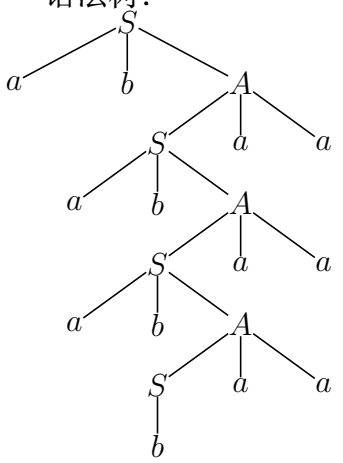
$$S \rightarrow abSaa \mid b;$$

(6) 与(1)不同的语句$abababbaaaaaa$的另一个最右推导和另一颗语法树为

最右推导如下：

$$\begin{aligned} S &\underset{rm}{\Longrightarrow} abA \\ &\underset{rm}{\Longrightarrow} abSaa \\ &\underset{rm}{\Longrightarrow} ababAaa \\ &\underset{rm}{\Longrightarrow} ababSaaaa \\ &\underset{rm}{\Longrightarrow} abababAaaaa \\ &\underset{rm}{\Longrightarrow} abababSaaaaaa \\ &\underset{rm}{\Longrightarrow} abababbaaaaaa \end{aligned}$$

语法树：



由此文法G是二义文法，而二义文法一定不是LR(k)文法。
也可答LR(1)分析器在移进$(ab)^n b$后进入状态$\{\langle S \rightarrow b\bullet, a/\$\rangle, \langle A \rightarrow b \bullet aa, a/\$\rangle\}$，面对输入$a$可移进到状态$\{\langle A \rightarrow ba \bullet a, a/\$\rangle\}$，也可用产生式$S \rightarrow b$归约，即有移进/归约冲突。

三、 (1) 面对输入“$id = id + id$”有两个不同的最左推导。

推导1：

$$\begin{aligned} E &\underset{lm}{\Longrightarrow} E + E \\ &\underset{lm}{\Longrightarrow} E = E + E \\ &\underset{lm}{\Longrightarrow} id = E + E \\ &\underset{lm}{\Longrightarrow} id = id + E \\ &\underset{lm}{\Longrightarrow} id = id + id \end{aligned}$$

推导2：

$$\begin{aligned} E &\underset{lm}{\Longrightarrow} E = E \\ &\underset{lm}{\Longrightarrow} id = E \\ &\underset{lm}{\Longrightarrow} id = E + E \\ &\underset{lm}{\Longrightarrow} id = id + E \\ &\underset{lm}{\Longrightarrow} id = id + id \end{aligned}$$

(2)

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

四、 (1) 识别活前缀的自动机在吃进 $E+**$ 之后到达状态I_1，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：

$$\overline{\{E \to * \bullet E\}} = \{\, E \to * \bullet E,\ E \to \bullet E = E,\\ E \to \bullet E + E, E \to \bullet * E,\ E \to \bullet id \,\}$$

识别活前缀的自动机在吃进 $*E = E$ 之后到达状态I_8, 不能再接受任何非终结符，所对应的项目集：

$$\begin{aligned}&\overline{\{E \to E = E\bullet,\ E \to E \bullet + E,\ E \to E\bullet = E\}}\\ =&\{E \to E = E\bullet,\ E \to E \bullet + E,\ E \to E\bullet = E\}\end{aligned}$$

(2) 状态I_3, I_7和I_8面对‘=’和‘+’有移进/归约冲突。

(3) $\mathrm{First}(E) = \{\, id,\ * \,\}$, $\mathrm{Follow}(E) = \{\, +,\ =, \$ \,\}$, SLR分析表如下所示：

	action					goto
状态	$*$	$+$	$=$	id	\$	E
0	s1			s3		2
1	s1			s3		4
2		s5	s6		acc	
3		r4	r4		r4	
4		r3	r3		r3	
5	s1			s3		7
6	s1			s3		8
7		r1	r1		r1	
8		s5	s6		r2	

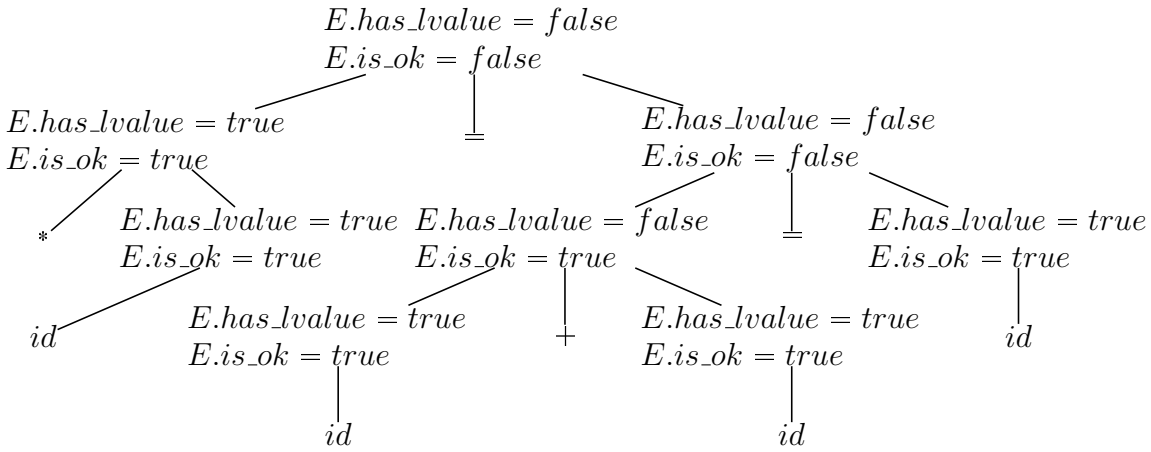
(4) “$id = *id + id$”的分析过程如下所示：

剩余串	分析栈	分析动作
$id = *id + id\$$	0	shift
$= *id + id\$$	$0id3$	reduce $E \to id$
$= *id + id\$$	$0E2$	shift
$*id + id\$$	$0E2 = 6$	shift
$id + id\$$	$0E2 = 6 * 1$	shift
$+id\$$	$0E2 = 6 * 1id3$	reduce $E \to id$
$+id\$$	$0E2 = 6 * 1E4$	reduce $E \to *E$
$+id\$$	$0E2 = 6E8$	shift
$id\$$	$0E2 = 6E8 + 5$	shift
\$	$0E2 = 6E8 + 5id3$	reduce $E \to id$
\$	$0E2 = 6E8 + 5E7$	reduce $E \to E + E$
\$	$0E2 = 6E8$	reduce $E \to E = E$
\$	$0E2$	分析成功

五、 (1)

产生式	语义规则
$E \to E_1 + E_2$	$E.has\_lvalue = false$ $E.is\_ok = E_1.is\_ok \wedge E_2.is\_ok$
$E \to E_1 = E_2$	$E.has\_lvalue = false$ $E.is\_ok = E_1.has\_lvalue \wedge E_1.is\_ok \wedge E_2.is\_ok$
$E \to *E_1$	$E.has\_lvalue = true$ $E.is\_ok = E_1.is\_ok$
$E \to id$	$E.has\_lvalue = true$ $E.is\_ok = true$

(2) $*id = id + id = id$的附注语法树:



六、

```
l6: t0 := x + 1
    x := t0
    if (x = 10) goto l0
    goto l2
l0: if (x = 20) goto l1
    goto l2
l1: t1 := x + 10
    x := t1
    goto l4
```

```
l2: goto l7
l4: if (x = 100) goto l3
    goto l7
l3: goto l8
l7: if (x < 0) goto l8
    goto l5
l5: if (x > 110) goto l8
    goto l6
l8:
```

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

address	memory	note
	`......`	
x	11	← a
x-4	22	← b
x-8	11	← 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“`*(cp - 2) -= 4;`”将把地址`x -16`上的值减去4，即函数`main()`的frame pointer减少了4个字节的偏移量，这样`foo()`返回后恢复`main()`的运行环境时，其`ebp`寄存器就比调用`foo()`前少了4个字节，所以当`main()`在调用`foo()`之后通过`ebp + offset`访问`a`时，实际上是取得是比原`a`地址低4个字节上的值，即`b`，所以在`main()`中执行的语句“`printf("a = %d\n", a);`”输出`a = 22`.