

特别说明

此资料来自豆丁网(<http://www.docin.com/>)

您现在所看到的文档是使用**下载器**所生成的文档

此文档的原件位于

<http://www.docin.com/p-6260550.html>

感谢您的支持

抱米花

<http://blog.sina.com.cn/lotusbaob>

第一章 引言

1. 解释下列名词

源程序, 目标程序, 翻译程序, 汇编程序, 编译程序, 遍

答: 源程序: 由汇编语言或高级程序语言编写的程序。

目标程序: 由目标语言编写的程序。

翻译程序: 把源程序转化为目标程序的程序。

汇编程序: 把由汇编语言编写的源程序转换成目标程序(由机器语言编写)的翻译程序。

编译程序: 把由高级程序语言编写的源程序转换为一台具体计算机的机器语言或汇编语言程序的翻译程序。

遍: 对源程序或源程序的中间形式从头到尾扫描一遍, 并做有关的加工处理, 生成新的源程序的中间形式或目标程序。

2. 典型的编译程序可划分为哪几个主要的逻辑部分? 各部分的主要功能是什么?

答: 典型的编译程序可划分 7 个主要的逻辑部分(模块)。

—词法分析 将字符串形式的源程序分解为具有独立语法语意的单词符号(Token);

—语法分析 从词法分析程序取得源程序, 并将一个或多个单词组合为语言的各种语法类。

—语义分析 确定源程序的意义(语义)。

—代码生成 将源程序的中间形式转化为汇编语言或机器语言。

—代码优化 获得更高效的目标程序。

—出错处理 对源程序的错误精确定位并能够跳过错误继续编译。

—符号表管理 保存每个标志符及其属性, 并保存过程名及其参数。

注意: 区分编译过程的 5 个阶段和编译程序的 7 个逻辑组成部分。

第二章 文法和语言的概念和表示

设关于句子的一组规则为:

〈句子〉:: = 〈主语〉〈谓语〉

〈主语〉:: = 〈名词短语〉

〈名词短语〉:: = 〈名词〉

〈名词短语〉:: = 〈形容词〉〈名词短语〉

〈形容词〉:: = big

〈形容词〉:: = brown

〈形容词〉:: = roasted

〈名词〉:: = John

〈名词〉:: = peanut

〈谓语〉:: = 〈动词〉〈直接宾语〉

〈动词〉:: = ate

〈直接宾语〉:: = 〈冠词〉〈名词短语〉

〈冠词〉:: = the

给出下述句子的推导, 并画出语法树。

解: (A) 〈句子〉=>〈主语〉〈谓语〉

=>〈主语〉〈动词〉〈直接宾语〉

=>〈主语〉〈动词〉〈冠词〉〈名词短语〉

=>〈主语〉〈动词〉〈冠词〉〈形容词〉〈名词短语〉

=>〈主语〉〈动词〉〈冠词〉〈形容词〉〈名词〉

=>〈主语〉〈动词〉〈冠词〉〈形容词〉peanut

=>〈主语〉〈动词〉〈冠词〉big peanut

=>〈主语〉〈动词〉the big peanut

=>〈主语〉ate the big peanut

=>〈名词短语〉ate the big peanut

=> 〈名词〉 ate the big peanut

=> John ate the big peanut

(语法树略)

(B) 〈句子〉=>〈主语〉〈谓语〉

=>〈名词短语〉〈谓语〉

=>〈名词〉〈谓语〉

=> John 〈谓语〉

=> John 〈动词〉〈直接宾语〉

=> John ate 〈直接宾语〉

=> John ate 〈冠词〉〈名词短语〉

=> John ate the 〈名词短语〉

=> John ate the 〈形容词〉〈名词短语〉

=> John ate the big 〈名词短语〉

=> John ate the big 〈形容词〉〈名词短语〉

=> John ate the big brown 〈名词短语〉

=> John ate the big brown 〈名词〉

=> John ate the big brown peanut

(语法树略)

(C) 〈句子〉=>〈主语〉〈谓语〉

=>〈名词短语〉〈谓语〉

=>〈名词〉〈谓语〉

=> John 〈谓语〉

=> John 〈动词〉〈直接宾语〉

=> John ate 〈直接宾语〉

=> John ate 〈冠词〉〈名词短语〉

=> John ate the 〈名词短语〉

=> John ate the 〈形容词〉〈名词短语〉

=> John ate the big 〈名词短语〉

=> John ate the big 〈形容词〉〈名词短语〉

=> John ate the big roasted 〈名词短语〉

=> John ate the big roasted 〈名词〉

=> John ate the big roasted peanut

(语法树略)

2. 利用规则 2-1, 除最左推导外, 给出句子 the big elephant ate the peanut 的另外两种推导 (其中一种为最右推导)。

答: (A) 最右推导

〈句子〉=>〈主语〉〈谓语〉

=>〈主语〉〈动词〉〈直接宾语〉

=>〈主语〉〈动词〉〈冠词〉〈名词〉

=>〈主语〉〈动词〉〈冠词〉 peanut

=>〈主语〉〈动词〉 the peanut

=>〈主语〉 ate the peanut

=>〈冠词〉〈形容词〉〈名词〉 ate the peanut

=>〈冠词〉〈形容词〉 elephant ate the peanut

=>〈冠词〉 big elephant ate the peanut

=> the big elephant ate the peanut

(B) 〈句子〉=>〈主语〉〈谓语〉

=>〈主语〉〈动词〉〈直接宾语〉

=>〈冠词〉〈形容词〉〈名词〉〈动词〉〈直接宾语〉

=>〈冠词〉〈形容词〉〈名词〉〈动词〉〈冠词〉〈名词〉

=> the 〈形容词〉〈名词〉〈动词〉〈冠词〉〈名词〉

=> the big 〈名词〉〈动词〉〈冠词〉〈名词〉

=> the big elephant 〈动词〉〈冠词〉〈名词〉

=> the big elephant ate 〈冠词〉〈名词〉

=>the big elephant ate the 〈名词〉

=>the big elephant ate the peanut

P19

1. 令 $A=\{\$, \}$, 又令 $Z=\$,$ 写出如下的符号串以及它们的长度:

$Z, ZZ, Z^2, Z^3, Z^0, A^*=?$

解: $Z=\$ \quad |Z|=1$
 $ZZ=\$ \$ \quad |ZZ|=2$
 $Z^2=ZZ=\$ \$ \quad |Z^2|=2$
 $Z^3=\$ \$ \$ \quad |Z^3|=3$
 $Z^0=\epsilon \quad |Z^0|=1$
 $A^*=\{\epsilon, \$, \$ \$, \$ \$ \$, \dots\} \quad |A^*|=\infty$

2. 令 $A=\{0, 1, 2\}$, 又令 $x=01, y=2, z=001$, 写出如下符号串及它们的长度:

$xy, xyz, x^4, (x^3)(y^2), (xy)^2$

解: $xy=012, |xy|=3 \quad xyz=012001, |xyz|=6$

$x^4=xxxx=01010101, |x^4|=8$

$(x^3)(y^2)=01010122, |(x^3)(y^2)|=8$

$(xy)^2=012012, |(xy)^2|=6$

令 $A=\{0, 1, 2\}$, 写出集合 A^+ 和集合 A^* 的 7 个最短的符号串。

解: A^+ 的 7 个最短的符号串为:

0, 1, 2, 00, 01, 02, 10 (答案不唯一)

A^* 的 7 个最短的符号串为:

$\epsilon, 0, 1, 2, 00, 01, 02,$ (答案不唯一)

4. 试证明: $A^+=AA^*=A^*A$

证: $\because A^*=A^0 \cup A^1, A^+=A^1 \cup A^2 \cup \dots \cup A^n \cup \dots$

得: $A^*=A^0 \cup A^1 \cup A^2 \cup \dots \cup A^n \cup \dots$

$\therefore AA^*=A(A^0 \cup A^1 \cup A^2 \cup \dots \cup A^n \cup \dots)$

$=AA^0 \cup AA^1 \cup AA^2 \cup \dots \cup AA^n \cup \dots$

$=A \cup A^2 \cup A^3 \cup A^{n+1} \cup \dots$

$=A^+$

同理可得: $A^*A=(A^0 \cup A^1 \cup A^2 \cup \dots \cup A^n \cup \dots)A$

$=A^0A \cup A^1A \cup A^2A \cup \dots \cup A^nA \cup \dots$

$=A \cup A^2 \cup A^3 \cup A^{n+1} \cup \dots$

$=A^+$

因此: $A^+=AA^*=A^*A$

P26

1. 设 $G[\langle \text{标志符} \rangle]$ 的规则是:

$\langle \text{标志符} \rangle ::= a|b|c|\langle \text{标志符} \rangle a|\langle \text{标志符} \rangle c|\langle \text{标志符} \rangle 0|\langle \text{标志符} \rangle 1$

试写出 V_T 和 V_N , 并对下列符号串 $a, ab0, a0c01, 0a, 11, aaa$ 给出可能的一些推导。

解: $V_T=\{a, b, c, 0, 1\}, V_N=\{\langle \text{标志符} \rangle\}$

不能推导出 $ab0, 11, 0a$

$\langle \text{标志符} \rangle \Rightarrow a$

$\langle \text{标志符} \rangle \Rightarrow \langle \text{标志符} \rangle 1$

$\Rightarrow \langle \text{标志符} \rangle 01$

$\Rightarrow \langle \text{标志符} \rangle c01$

$\Rightarrow \langle \text{标志符} \rangle 0c01$

$\Rightarrow a0c01$

(4) $\langle \text{标志符} \rangle \Rightarrow \langle \text{标志符} \rangle a$

$\Rightarrow \langle \text{标志符} \rangle aa$

$\Rightarrow aaa$

2. 写一文法, 使其语言是偶整数的集合。

解: $G[\langle \text{偶整数} \rangle]$:

$\langle \text{偶整数} \rangle ::= \langle \text{符号} \rangle \langle \text{偶数字} \rangle \mid \langle \text{符号} \rangle \langle \text{数字串} \rangle \langle \text{偶数字} \rangle$, (作用是一位和多位)

$\langle \text{符号} \rangle ::= + \mid - \mid \epsilon$

$\langle \text{数字串} \rangle ::= \langle \text{数字串} \rangle \langle \text{数字} \rangle \mid \langle \text{数字} \rangle$

$\langle \text{数字} \rangle ::= \langle \text{偶数字} \rangle \mid 1 \mid 3 \mid 5 \mid 7 \mid 9$

$\langle \text{偶数字} \rangle ::= 0 \mid 2 \mid 4 \mid 6 \mid 8$

3. 写一文法, 使其语言是偶整数的集合, 但不允许有以 0 开头的偶整数。

解: $G[\langle \text{偶整数} \rangle]$:

$\langle \text{偶整数} \rangle ::= \langle \text{符号} \rangle \langle \text{单偶数} \rangle \mid \langle \text{符号} \rangle \langle \text{首数字} \rangle \langle \text{数字串} \rangle \langle \text{尾偶数} \rangle$

$\langle \text{符号} \rangle ::= + \mid - \mid \epsilon$

$\langle \text{单偶数} \rangle ::= 2 \mid 4 \mid 6 \mid 8$

$\langle \text{尾偶数} \rangle ::= 0 \mid \langle \text{单偶数} \rangle$

$\langle \text{首数字} \rangle ::= 1 \mid 3 \mid 5 \mid 7 \mid 9 \mid \langle \text{单偶数} \rangle$

$\langle \text{数字串} \rangle ::= \langle \text{数字串} \rangle \langle \text{数字} \rangle \mid \langle \text{数字} \rangle \mid \epsilon$

$\langle \text{数字} \rangle ::= 0 \mid \langle \text{首数字} \rangle$

4. 设文法 G 的规则是:

$\langle A \rangle ::= b\langle A \rangle \mid cc$

试证明: $cc, bcc, bbcc, bbbcc \in L[G]$

证: (1) $\langle A \rangle \Rightarrow cc$

(2) $\langle A \rangle \Rightarrow b \langle A \rangle \Rightarrow bcc$

(3) $\langle A \rangle \Rightarrow b \langle A \rangle \Rightarrow bb \langle A \rangle \Rightarrow bbcc$

(4) $\langle A \rangle \Rightarrow b \langle A \rangle \Rightarrow bb \langle A \rangle \Rightarrow bbb \langle A \rangle \Rightarrow bbbcc$

又 $\because cc, bcc, bbcc, bbbcc \in Vt^*$

\therefore 由语言定义, $cc, bcc, bbcc, bbbcc \in L[G]$

5. 试对如下语言构造相应文法:

$\{ a(b^n)a \mid n=0, 1, 2, 3, \dots \}$, 其中左右圆括号 of 终结符。

$\{ (a^n)(b^n) \mid n=1, 2, 3, \dots \}$

解: (1) 文法 $[G \langle S \rangle]$:

$S ::= a(B)a$

$B ::= bB \mid b \mid \epsilon$

(2) 文法 $[G \langle S \rangle]$:

$S ::= (A)(B)$

$A ::= aA \mid a$

$B ::= bB \mid b$

6. 文法 $G_3[\langle \text{表达式} \rangle]$:

$\langle \text{表达式} \rangle ::= \langle \text{项} \rangle \mid \langle \text{表达式} \rangle + \langle \text{项} \rangle \mid \langle \text{表达式} \rangle - \langle \text{项} \rangle$

$\langle \text{项} \rangle ::= \langle \text{因子} \rangle \mid \langle \text{项} \rangle * \langle \text{因子} \rangle \mid \langle \text{项} \rangle / \langle \text{因子} \rangle$

$\langle \text{因子} \rangle ::= (\langle \text{表达式} \rangle) \mid i$

试给出下列符号串的推导:

$i, (i), i*i, i*i+i, i*(i+i)$

解: (1) $\langle \text{表达式} \rangle \Rightarrow \langle \text{项} \rangle$

$\Rightarrow \langle \text{因子} \rangle$

$\Rightarrow i$

(2) $\langle \text{表达式} \rangle \Rightarrow \langle \text{项} \rangle$

$\Rightarrow \langle \text{因子} \rangle$

$\Rightarrow (\langle \text{表达式} \rangle)$

$\Rightarrow (\langle \text{项} \rangle)$

$\Rightarrow (\langle \text{因子} \rangle)$

$\Rightarrow (i)$

(3) $\langle \text{表达式} \rangle \Rightarrow \langle \text{项} \rangle$

$\Rightarrow \langle \text{项} \rangle * \langle \text{因子} \rangle$

$\Rightarrow \langle \text{项} \rangle * i$

$\Rightarrow \langle \text{因子} \rangle * i$

$\Rightarrow i * i$

(4) $\langle \text{表达式} \rangle \Rightarrow \langle \text{表达式} \rangle + \langle \text{项} \rangle$

$\Rightarrow \langle \text{表达式} \rangle + \langle \text{因子} \rangle$

$\Rightarrow \langle \text{表达式} \rangle + i$

$\Rightarrow \langle \text{项} \rangle + i$

$\Rightarrow \langle \text{项} \rangle * \langle \text{因子} \rangle + i$

$\Rightarrow \langle \text{项} \rangle * i + i$

$\Rightarrow \langle \text{因子} \rangle * i + i$

$\Rightarrow i * i + i$

(5) $\langle \text{表达式} \rangle \Rightarrow \langle \text{项} \rangle$

$\Rightarrow \langle \text{项} \rangle * \langle \text{因子} \rangle$

$\Rightarrow \langle \text{项} \rangle * (\langle \text{表达式} \rangle)$

$\Rightarrow \langle \text{项} \rangle * (\langle \text{表达式} \rangle + \langle \text{项} \rangle)$

$\Rightarrow \langle \text{项} \rangle * (\langle \text{表达式} \rangle + \langle \text{因子} \rangle)$

$\Rightarrow \langle \text{项} \rangle * (\langle \text{表达式} \rangle + i)$

$\Rightarrow \langle \text{项} \rangle * (\langle \text{项} \rangle + i)$

$\Rightarrow \langle \text{项} \rangle * (\langle \text{因子} \rangle + i)$

$\Rightarrow \langle \text{项} \rangle * (i + i)$

$\Rightarrow \langle \text{因子} \rangle * (i + i)$

$\Rightarrow i * (i + i)$

7. 对文法 $G_3[\langle \text{表达式} \rangle]$ (见上题), 列出句型 $\langle \text{表达式} \rangle + \langle \text{项} \rangle * \langle \text{因子} \rangle$ 的所有短语和简单短语。

解: 句型 $\langle \text{表达式} \rangle + \langle \text{项} \rangle * \langle \text{因子} \rangle$ 的短语有:

$\langle \text{表达式} \rangle + \langle \text{项} \rangle * \langle \text{因子} \rangle$ 和 $\langle \text{项} \rangle * \langle \text{因子} \rangle$

简单短语是: $\langle \text{项} \rangle * \langle \text{因子} \rangle$

注意: 求短语, 简单短语和句柄尽量用语法树。

8. 文法 $V ::= aaV \mid bc$ 的语言是什么?

解: $L(G[V]) = \{a^{2n}bc \mid n=0, 1, 2, \dots\}$

9. 设文法 G 为:

$N ::= D \mid ND$

$D ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

G 的语言是什么?

给出句子 0125 和 783 的最右推导 (规范推导) 和最左推导。

解: (1) $L(G) = \{\text{所有无符号整数}\}$

(2) $N \Rightarrow ND$

$\Rightarrow N5$

$\Rightarrow ND5$

$\Rightarrow N25$

$$\Rightarrow ND25$$

$$\Rightarrow N125$$

$$\Rightarrow 0125 \quad (\text{最右推导 (规范推导)})$$

$$N \Rightarrow ND$$

$$\Rightarrow NDD$$

$$\Rightarrow DDD$$

$$\Rightarrow 7DD$$

$$\Rightarrow 78D$$

$$\Rightarrow 783 \quad (\text{最左推导})$$

P33

1. 给定文法 $G[E]$ 为:
$$E ::= RP \mid P$$

$$P ::= (E) \mid i$$

$$R ::= RP+ \mid RP* \mid P+ \mid P*$$
证明 (1) $i+i*(i+i)$ 是文法 G 的句子。

(2) 画出该句子的语法树。

证: (1) $E \Rightarrow RP$

$$\Rightarrow RP* P$$

$$\Rightarrow RP* (E)$$

$$\Rightarrow RP* (RP)$$

$$\Rightarrow RP* (P+P)$$

$$\Rightarrow RP* (P+i)$$

$$\Rightarrow RP* (i+i)$$

$$\Rightarrow P+P* (i+i)$$

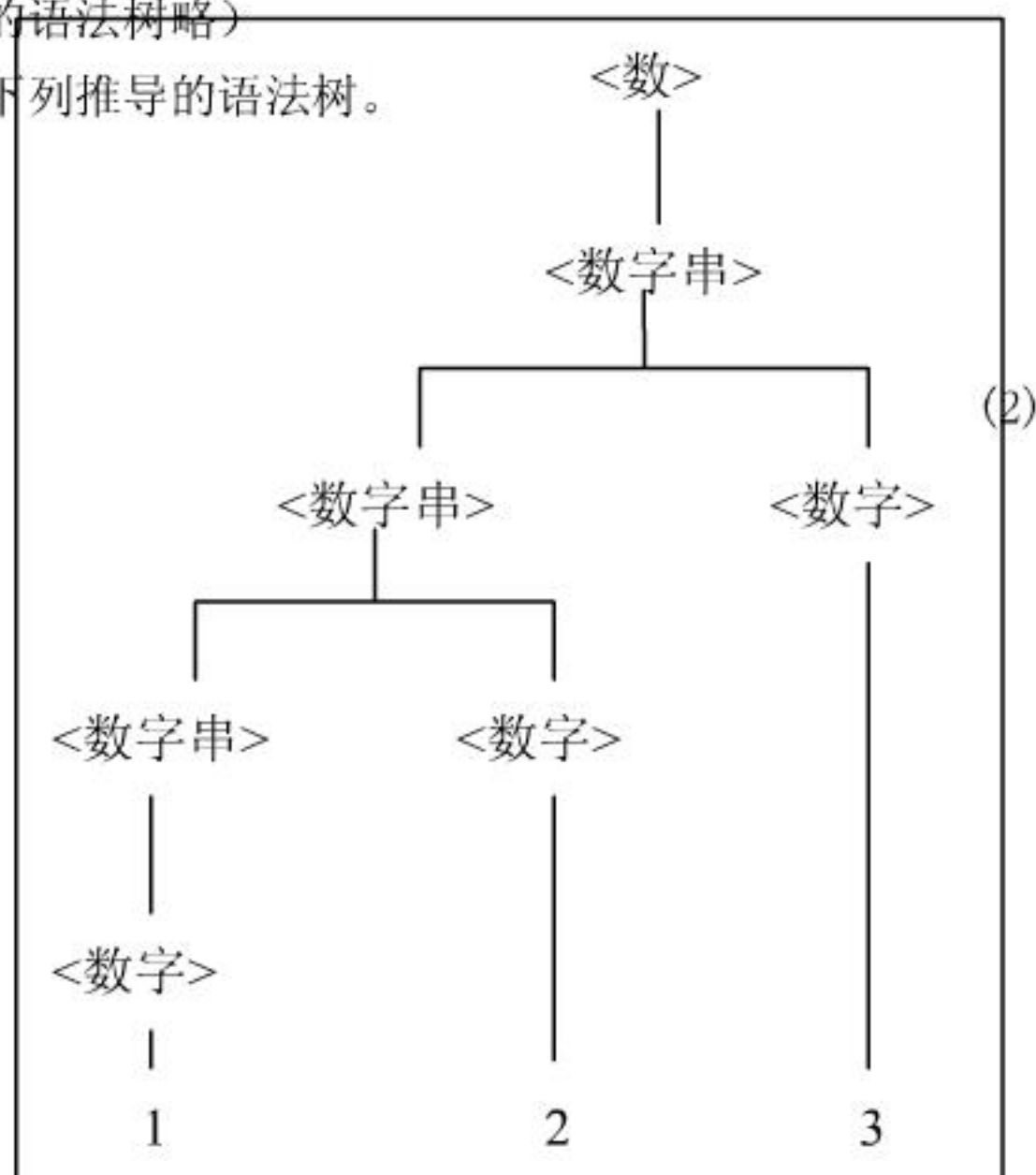
$$\Rightarrow P+i* (i+i)$$

$$\Rightarrow i+i* (i+i) \in Vt^*$$
 $\therefore i+i*(i+i)$ 是文法 G 的句子

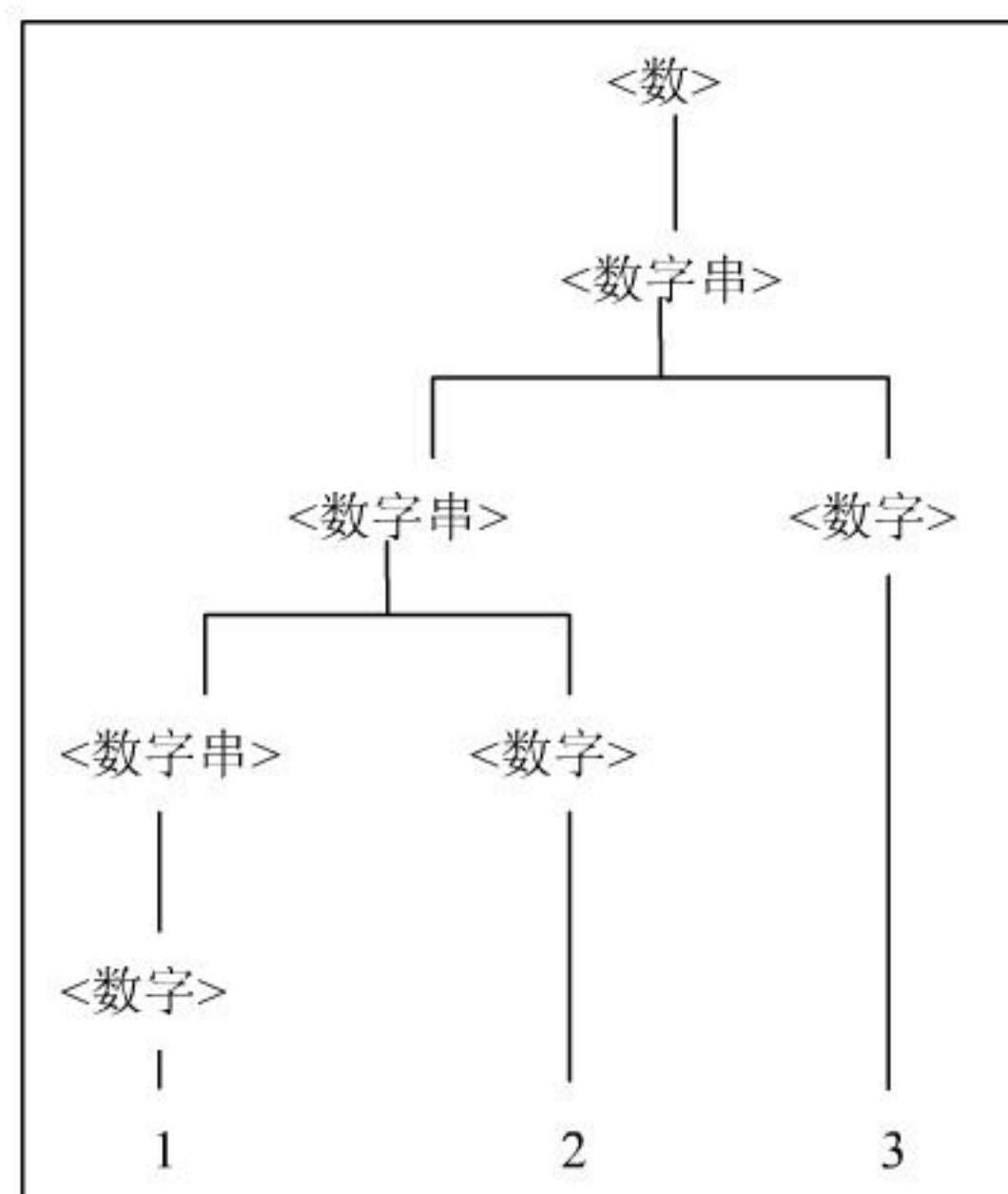
(对应的语法树略)

2. 画出下列推导的语法树。

(1)



(2)



由上图可知两种推导的语法树是相同的。

4. 由下述语法树构造推导: (语法树书上 P34 上, 未画)。

解: (1) $\langle \text{表达式} \rangle \Rightarrow \langle \text{项} \rangle$

$$\Rightarrow \langle \text{项} \rangle * \langle \text{因子} \rangle$$

$$\Rightarrow \langle \text{因子} \rangle * \langle \text{因子} \rangle$$

$$\Rightarrow i * \langle \text{因子} \rangle$$

$\Rightarrow i * i$
 $(2) \langle \text{表达式} \rangle \Rightarrow \langle \text{项} \rangle$
 $\Rightarrow \langle \text{项} \rangle * \langle \text{因子} \rangle$
 $\Rightarrow \langle \text{因子} \rangle * \langle \text{因子} \rangle$
 $\Rightarrow i * \langle \text{因子} \rangle$
 $\Rightarrow i * (\langle \text{表达式} \rangle)$
 $\Rightarrow i * (\langle \text{表达式} \rangle + \langle \text{项} \rangle)$
 $\Rightarrow i * (\langle \text{项} \rangle + \langle \text{项} \rangle)$
 $\Rightarrow i * (\langle \text{因子} \rangle + \langle \text{项} \rangle)$
 $\Rightarrow i * (i + \langle \text{项} \rangle)$
 $\Rightarrow i * (i + \langle \text{因子} \rangle)$
 $\Rightarrow i * (i + i)$

5. 已知文法 $G[E]$:

$E ::= ET+ \mid T$

$T ::= TF* \mid F$

$F ::= FP \uparrow \mid P$

$P ::= (E) \mid i$

有句型 $TF*PP \uparrow +$, 问此句型的短语, 简单短语, 和句柄是什么?

解: 此句型的短语有: $TF*PP \uparrow +$, $TF*$, $PP \uparrow$, P

简单短语有: $TF*$, P

句柄是: $TF*$

注意: 求短语, 简单短语和句柄尽量用语法树。

6. 分别对 $i+i*i$ 和 $i+i+i$ 中的每一个句子构造两棵语法树, 从而证明下列文法 $G[\langle \text{表达式} \rangle]$ 是二义的。

$\langle \text{表达式} \rangle ::= i \mid (\langle \text{表达式} \rangle) \mid \langle \text{表达式} \rangle \langle \text{运算符} \rangle \langle \text{表达式} \rangle$

$\langle \text{运算符} \rangle ::= + \mid - \mid * \mid /$

解: 以 $i+i*i$ 为例:

8. 证明下面的文法 G 是二义的:

$S ::= iSeS \mid iS \mid i$

证: 由文法可知 $iiiei$ 是该文法的句子, 又由文法可知 $iiiei$ 有两棵不同的语法树。

所以该文法是二义性文法。

7. 文法 $G_3[\langle \text{表达式} \rangle]$ 为:

$\langle \text{表达式} \rangle ::= \langle \text{项} \rangle \mid \langle \text{表达式} \rangle + \langle \text{项} \rangle \mid \langle \text{表达式} \rangle - \langle \text{项} \rangle$

$\langle \text{项} \rangle ::= \langle \text{因子} \rangle \mid \langle \text{项} \rangle * \langle \text{因子} \rangle \mid \langle \text{项} \rangle / \langle \text{因子} \rangle$

$\langle \text{因子} \rangle ::= (\langle \text{表达式} \rangle) \mid i$

证明此文法的句子 $i+i*i$ 和 $i+i+i$ 是无二义性的。在句子 $i+i*i$ 中, 哪种运算符是优先的? 在句子 $i+i+i$ 中, 又是哪种运算符?

证: (容易证明文法规则右边出现有选折项时, 各项首符号集不相交, 且该文法不是左递归文法)。

在句子 $i+i*i$ 中, 运算符 $*$ 是优先的

在句子 $i+i+i$ 中, 第一个 $+$ 运算符是优先的

9. 有文法 $G[N]$:

$N ::= SE \mid E$

$E ::= 0 \mid 2 \mid 4 \mid 6 \mid 8 \mid 10$

$S ::= SD \mid D$

$D ::= 0 \mid 1 \mid 2 \mid \dots \mid 9$

举例说明文法 $G[N]$ 是二义的。此文法描述的语言是什么? 试写出另一文法 G' ,

使 $L(G') = L(G)$, 且 G' 是无二义性的。

解: 由文法可知 10 是该文法的句子, 又由文法可知 10 有两棵不同的语法树。

此文法描述的语言 $L(G) = \{\text{无符号偶数}\}$

所求文法 $G' [\langle \text{无符号偶数} \rangle]$:

$\langle \text{无符号偶数} \rangle ::= \langle \text{偶数字} \rangle \mid \langle \text{数字串} \rangle \langle \text{偶数字} \rangle$

$\langle \text{数字串} \rangle ::= \langle \text{数字串} \rangle \langle \text{数字} \rangle \mid \langle \text{数字} \rangle$

$\langle \text{数字} \rangle ::= \langle \text{偶数字} \rangle \mid 1 \mid 3 \mid 5 \mid 7 \mid 9$

$\langle \text{偶数字} \rangle ::= 0 \mid 2 \mid 4 \mid 6 \mid 8$

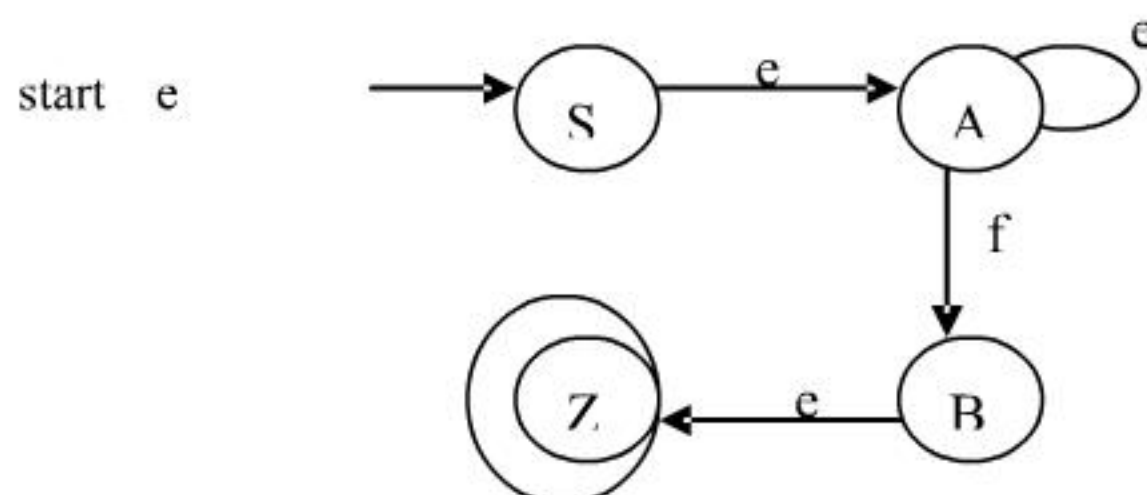
(****上述提供文法中的第二条规则右部的 10 选折项去掉似乎就符合要求了行****)

第三章

P67

1. 解:

其状态图为:



由状态图知:

$f, eeff$ 不是合法的句子

$eeffe$ 是该文法的句子。

2. 解:

(1) 正则文法为 $G[Z]$:

$Z ::= A1 \mid 0 \quad A ::= A0 \mid 0$

(2) 该文法的 $V = \{Z, A, 1, 0\} \quad V_n = \{Z, A\} \quad V_t = \{1, 0\}$

(3) 该文法所确定的语言为: $L(G[Z]) = \{0^n 1 \text{ 或 } 0 \mid n \geq 1\}$

5. 证明:

(1) $A \mid A^* = \{x \mid x \in LA \text{ 或 } x \in LA^*\} = \{x \mid x \in LA\} = A$

(2) $(A^*)^* = (A^*)^0 \cup (A^*)^1 \cup (A^*)^2 \cup \dots \cup (A^*)^n$

$= \epsilon \cup (A^0 \cup A^1 \cup A^2 \cup \dots \cup A^n) \cup (A^1 \dots) = \epsilon \cup A^0 \cup A^1 \cup A^2 \cup \dots \cup A^n$

$= A$

(3) $\epsilon \mid AA^*$ 所表示的语言是 $\{\epsilon\} \cup LA \cdot LA^* = LA^0 \cup LA (LA^0 \cup LA^1 \cup LA^2 \cup \dots)$

$= LA^0 \cup LA^1 \cup LA^2 \cup \dots = LA^*$

故 $\epsilon \mid AA^* = A^*$

(4) $(LALB)^* LA = (\{\epsilon\} \cup LALB \cup LALBLALB \cup LALBLALBLALB \cup \dots) LA$

$= LA \cup LALBLA \cup LALBLALBLA \cup LALBLALBLALBLA \cup LA \dots$

$= LA \cup (\{\epsilon\} \cup LBLA \cup LBLALBLA \cup \dots)$

$= LA(LBLA)$

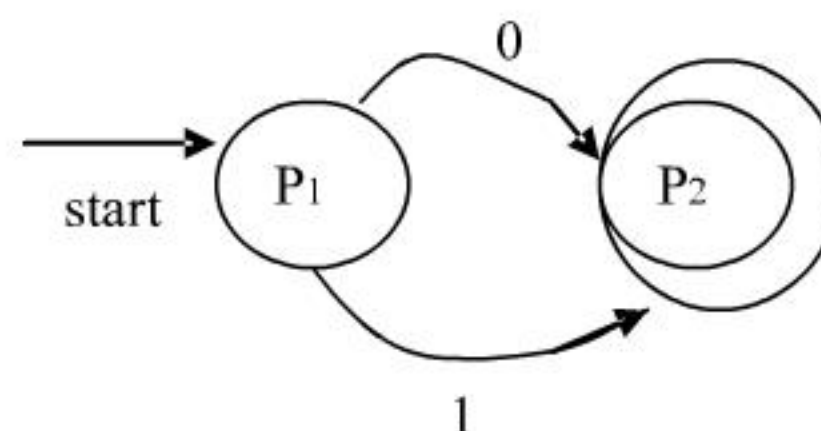
$\therefore (AB)^* = A(AB)^*$

(5) 三个表达式所描述的语言都是 $LALA$ 中任意组合

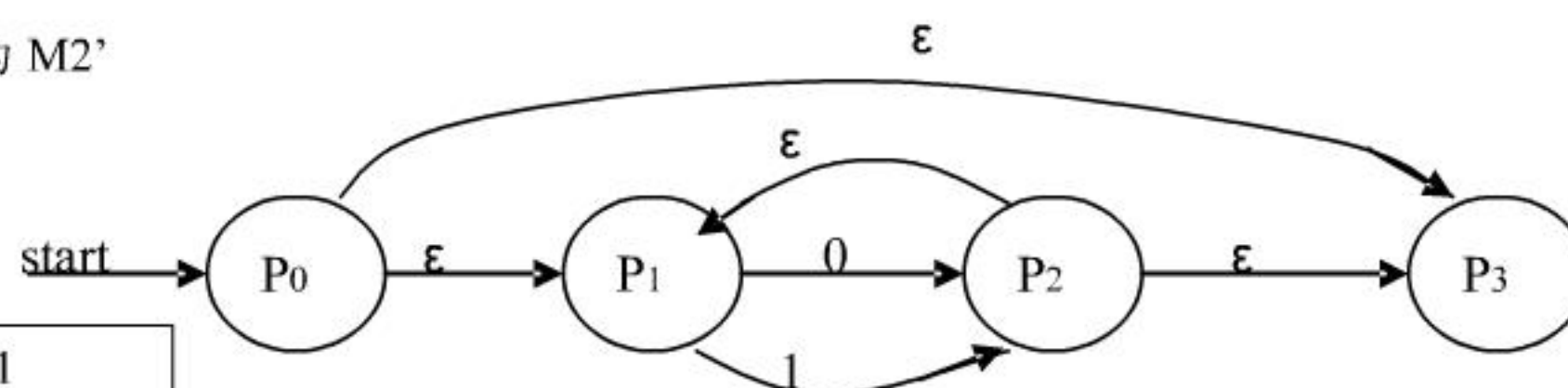
$\therefore (A|B)^* = (A^*B^*) = (A^*|B^*)^*$

6. 解:

(1) $R1 = 0 \mid 1$ 对应的自动机为



$(R1)^* = (0|1)^*$ 对应的自动机为 $M2'$



将 $M2'$ 确定化

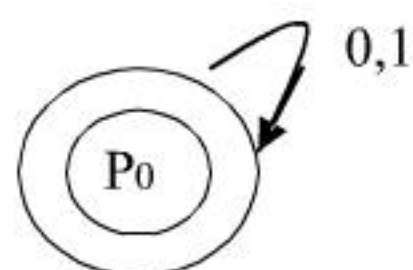
状态 \ 输入	ϵ	0	1
P0	$\{P1, P3\}$	\varnothing	\varnothing
P1	\varnothing	$\{P2\}$	$\{P2\}$
P2	$\{P1, P3\}$	\varnothing	\varnothing
P3	\varnothing	\varnothing	\varnothing

I	I0	I1	(空)	状态 入	输 0	1
{P0,P1,P3}	{P1,P2,P3}	{P1,P2,P3}		P0'	P1'	P1'
{P1,P2,P3}	{P1,P2,P3}	{P1,P2,P3}		P1'	P1'	P1'

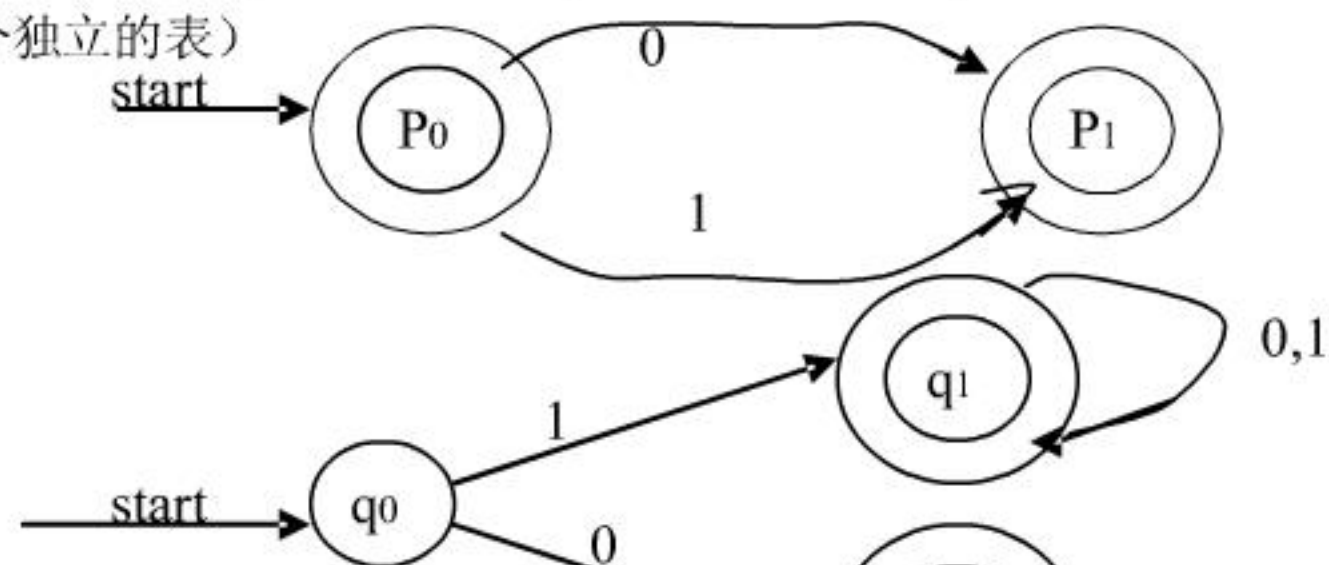
(注：上图中的(空)列表示没有该列。即上图两个独立的表)

状态转换图：

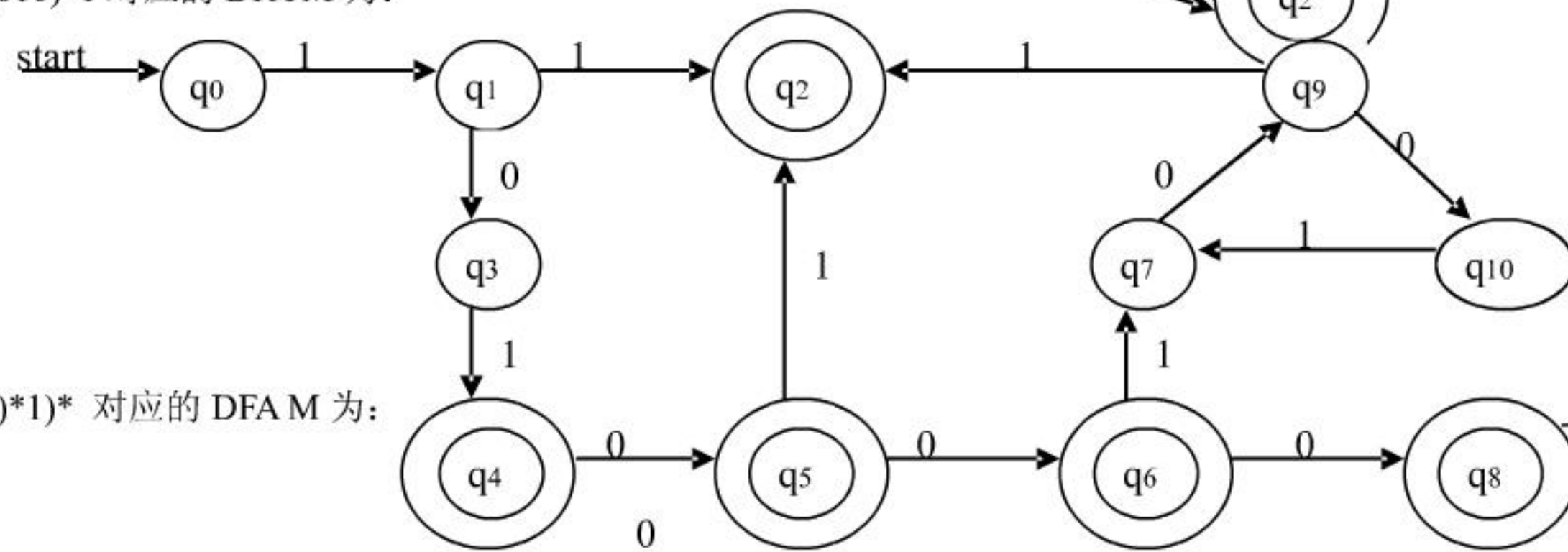
简化为：



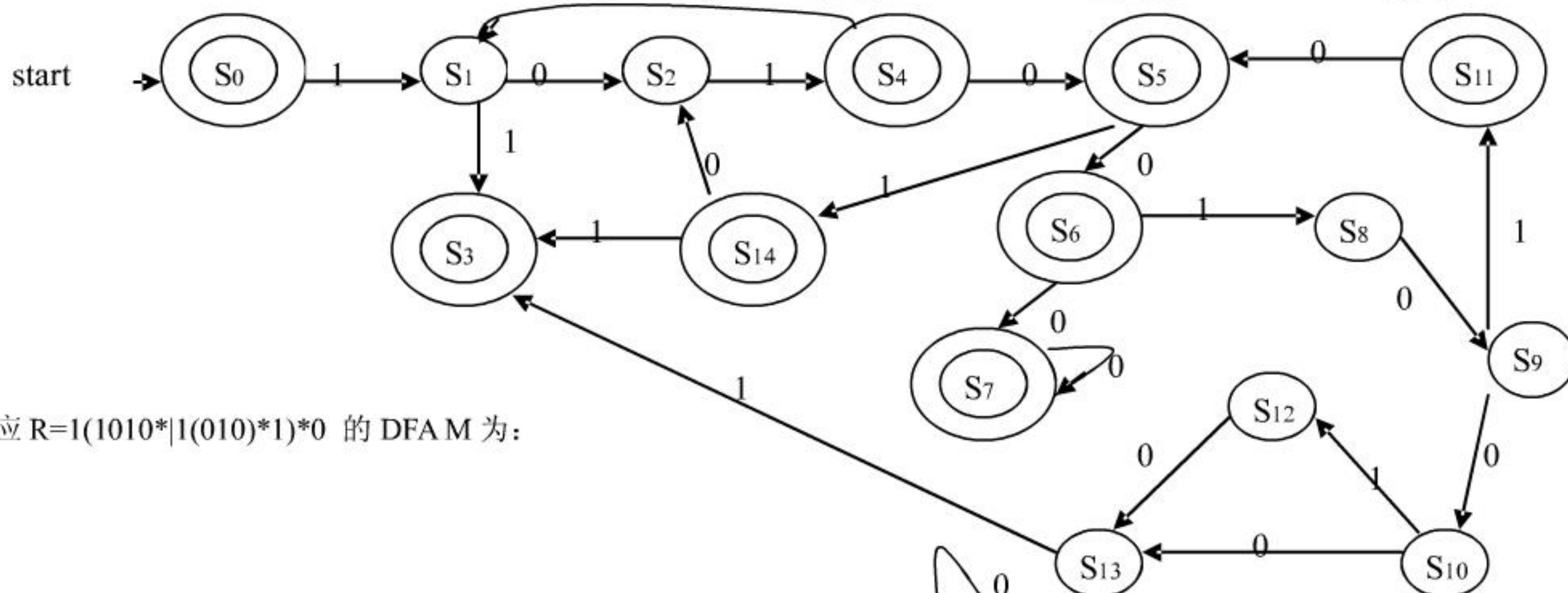
$\therefore R=1(0|1)^*0$ 的 DFAM 为：



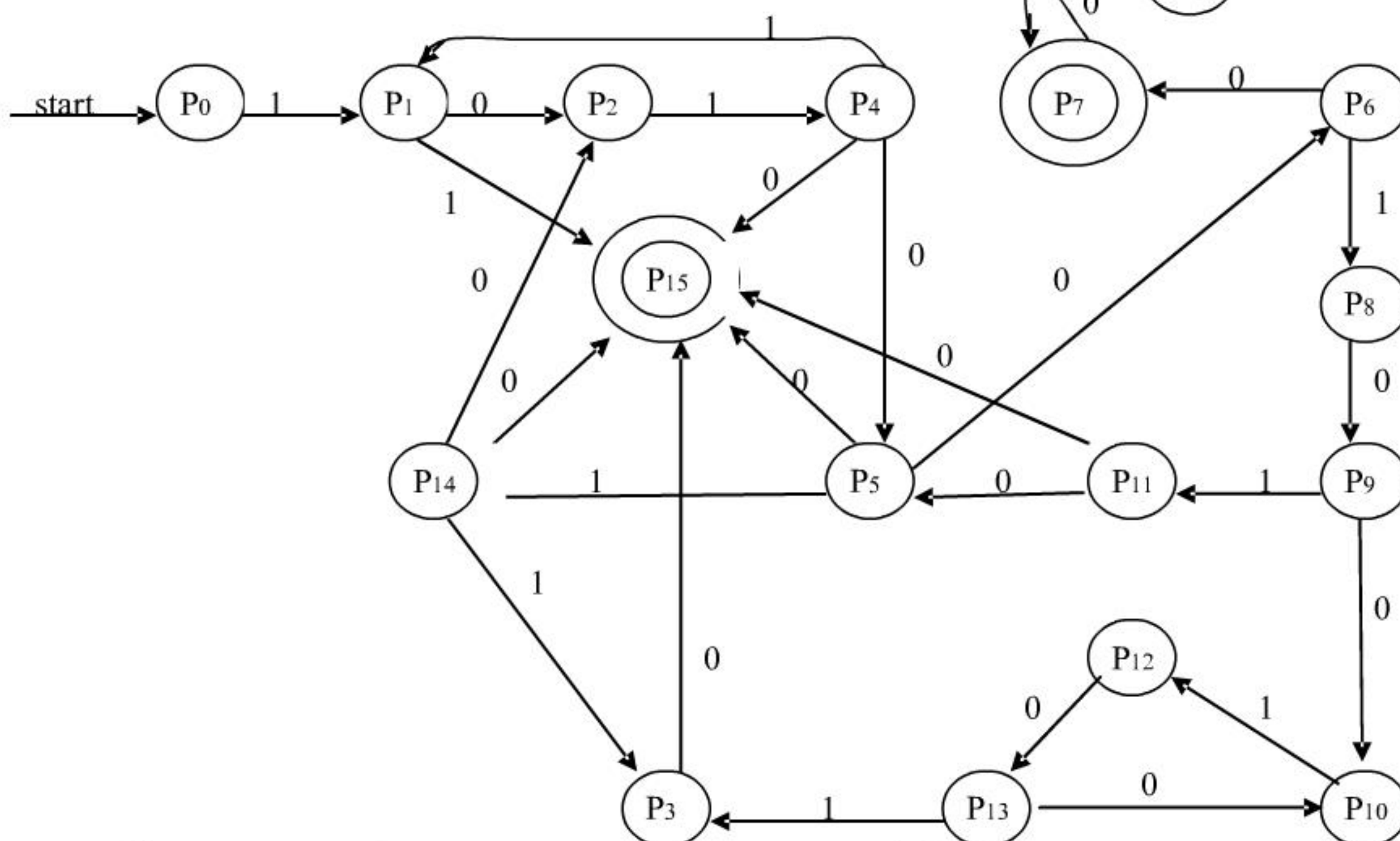
(2) $R1=1010^*1(010)^*1$ 对应的 DFAM 为：



$R2=(1010^*1(0|0)^*1)^*$ 对应的 DFAM 为：



对应 $R=1(1010^*1(010)^*1)^*0$ 的 DFAM 为：



8.解:

(a)

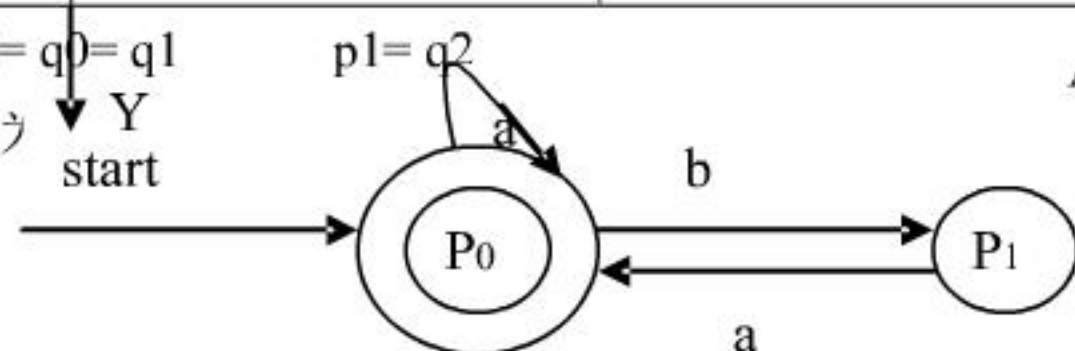
I	Ia	Ib
{0}	{0,1}	{1}
{0,1}	{0,1}	{1}
{1}	{0}	\varnothing

\therefore DFA 的状态转换矩阵为:

状态 \ 输入	a	b
q0	q1	q2
q1	q1	q2
q2	q0	\varnothing

化简: $p0 = q0 = q1$

状态图



A Y

Y | Y N

10. 解: 它所识别的语言为: $(1010)^*$

第四章 语法分析

P80

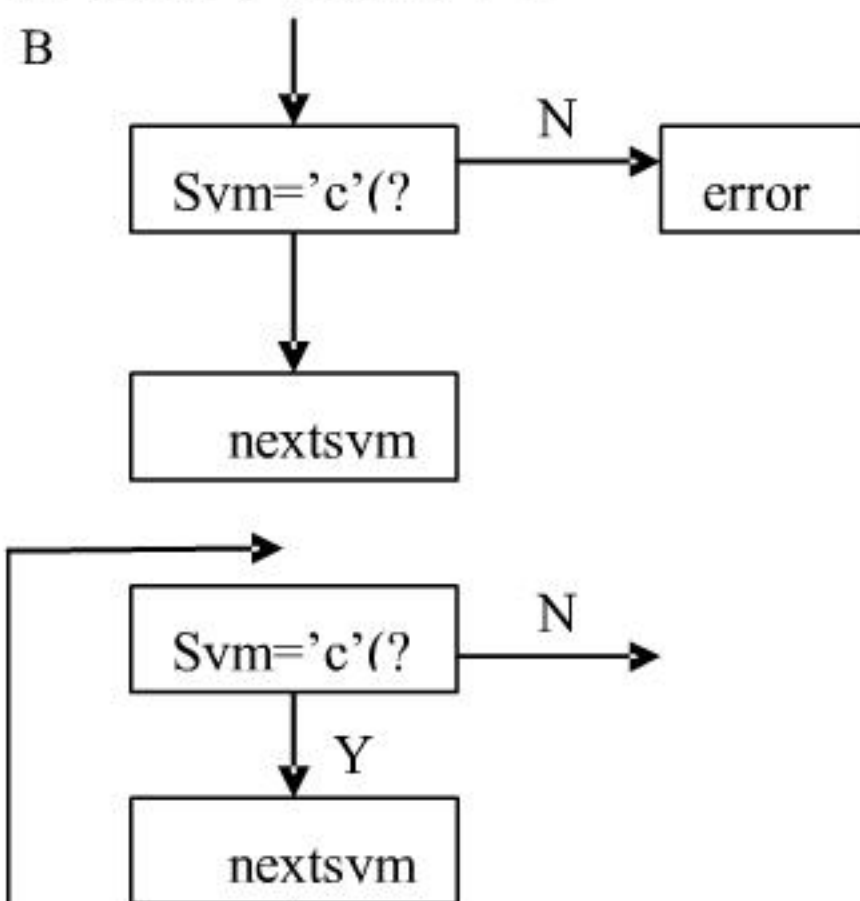
1. 解: (见课本)

2. 解:

文法有左递归, 故先改写文法:

$A ::= (B) | dB e$ $B ::= c \{c\}$

下面给出两个分析子程序的框图:

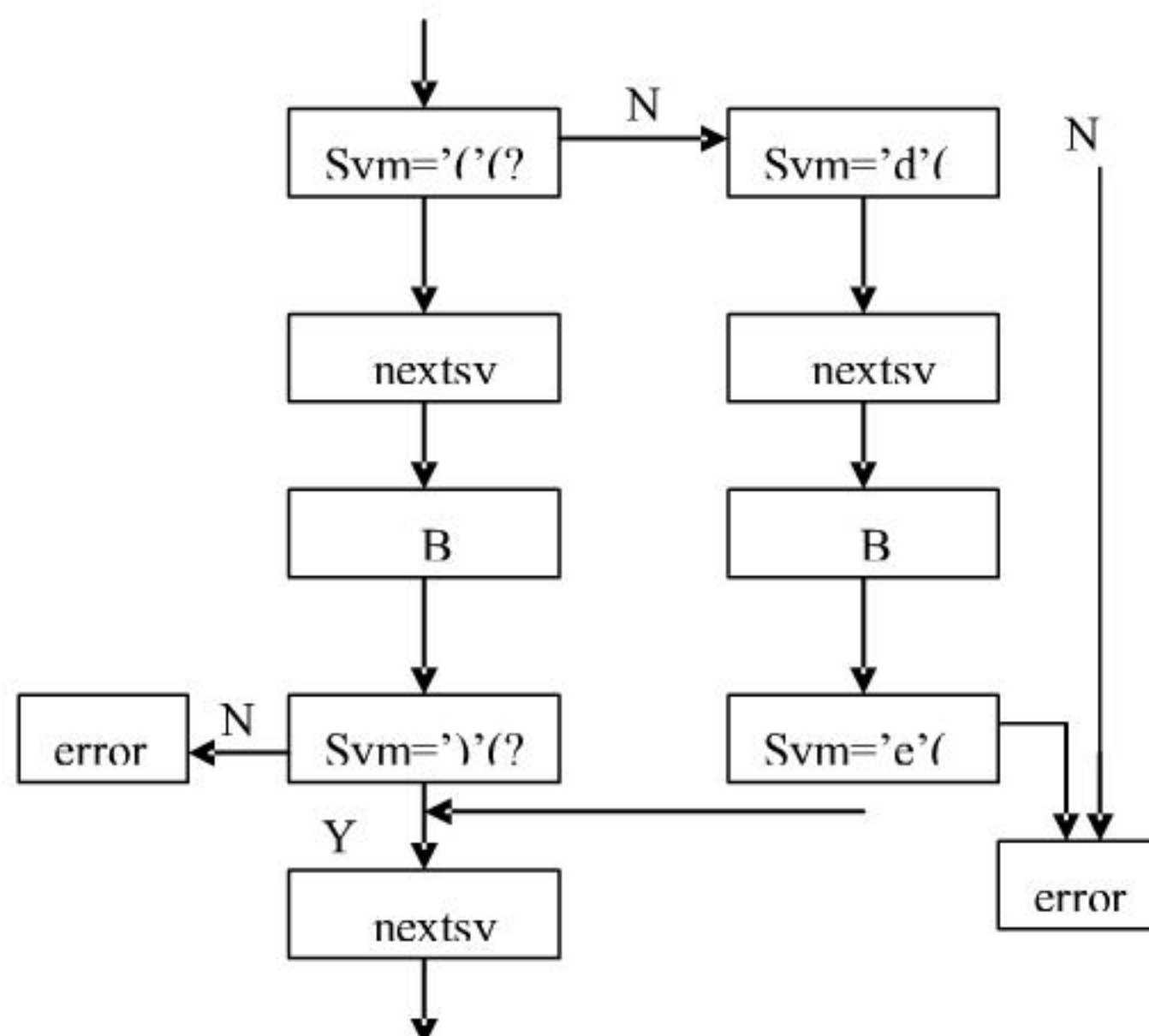


两分析子程序如下:

```

procedure A;
  IF sym='(' THEN
    Begin
      Nextsym;
      B;
      If sym=')' Then
        Nextsym;
      Else
        Error
    End
  End

```



<pre> If sym='a' then Begin B; If sym='d' then Nextsym; Else Error End; Else Error; Procedure A; If sym='c' then Begin Nextsym; </pre>	<pre> While sym='a' do B; End; Else Error; Procedure B; If sym='a' then Begin Nextsym; If sym='c' then A; End; Else Error; </pre>
--	--

4. 解:

首先改写文法如下:

〈语句〉 ::= 〈变量〉 ::= 〈表达式〉 | if 〈表达式〉 then 〈语句〉 [else 〈语句〉]
 〈变量〉 ::= i((〈表达式〉)|ε)
 〈表达式〉 ::= 〈项〉 {+ 〈项〉 }
 〈项〉 ::= 〈因子〉 { * 〈因子〉 }
 〈因子〉 ::= 〈变量〉 | (〈表达式〉)

语法分析程序(pascal)共有五个子程序 statement,expr,item,factor,var

<pre> procedure statement; if sym='if' then begin nextsym; expr; if sym='then' then begin nextsym; statement; if sym='else' then begin nextsym; statement; end; end; else error; end; else begin var; if sym=':=' then begin nextsym; expr; </pre>	<pre> end; else error; end; procedure expr; item; while sym='+' do begin nextsym; item; end; procedure item; factor; while sym='*' do begin nextsym; factor; end; procedure var; if sym='i' then begin nextsym; if sym='(' then </pre>
---	---

<pre> begin nextsym; expr; if sym=')' then nextsym; else error; end; end; else error; </pre>	<pre> procedure factor; if sym='(' then begin nextsym; expr; if sym=')' then nextsym; else error; end; else var; </pre>
--	---

P87

1.解:

<p>(1). $FIRST(P) = \{ (, a, b, \wedge \}$</p> <p>$FIRST(F') = \{ *, \epsilon \}$</p> <p>$FIRST(F) = \{ (, a, b, \wedge \}$</p> <p>$FIRST(T') = \{ (, a, b, \wedge, \epsilon \}$</p> <p>$FIRST(T) = \{ (, a, b, \wedge \}$</p> <p>$FIRST(E') = \{ +, \epsilon \}$</p> <p>$FIRST(E) = \{ (, a, b, \wedge \}$</p>	<p>$FOLLOW(E) = \{ \#, \}$</p> <p>$FOLLOW(E') = \{ \#, \}$</p> <p>$FOLLOW(T) = \{ \#, \}, + \}$</p> <p>$FOLLOW(T') = \{ \#, \}, + \}$</p> <p>$FOLLOW(F) = \{ (, a, b, \wedge \}, +, \# \}$</p> <p>$FOLLOW(F') = \{ (, a, b, \wedge \}, +, \# \}$</p> <p>$FOLLOW(P) = \{ (, a, b, \wedge \}, +, \#, * \}$</p>
--	---

(2)证明: 对于 $E' \rightarrow +E | \epsilon$

$FIRST(+E) = \{ + \}$ $FIRST(\epsilon) = \{ \epsilon \}$ $FOLLOW(E') = \{ \#, \}$

$\therefore FIRST(+E) \cap FIRST(\epsilon) = \Phi$ $FIRST(+E) \cap FOLLOW(E') = \Phi$

对于 $T' \rightarrow T | \epsilon$ $FIRST(T) \cap FOLLOW(T') = \Phi$ 对于 $F' \rightarrow *F' | \epsilon$ $FIRST(*F') \cap FOLLOW(F') = \Phi$ 对于 $P \rightarrow (E) | a | b | \wedge$ $FIRST((E)) = \{ (\}$ $FIRST(a) = \{ a \}$ $FIRST(b) = \{ b \}$ $FIRST(\wedge) = \{ \wedge \}$ \therefore 根据 LL(1)文法的充要条件可以断定该文法是 LL(1)文法.

(3).构造其分析表如下:

	a	b	\wedge	()	*	+	#
E	$E \rightarrow TE'$	$E \rightarrow TE'$	$E \rightarrow TE'$	$E \rightarrow TE'$				
E'					$E' \rightarrow \epsilon$		$E' \rightarrow +E$	$E' \rightarrow \epsilon$
T	$T \rightarrow FT'$	$T \rightarrow FT'$	$T \rightarrow FT'$	$T \rightarrow FT'$				
T'	$T' \rightarrow T$	$T' \rightarrow T$	$T' \rightarrow T$	$T' \rightarrow T$	$T' \rightarrow \epsilon$		$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow PF'$	$F \rightarrow PF'$	$F \rightarrow PF'$	$F \rightarrow PF'$				
F'	$F' \rightarrow \epsilon$	$F' \rightarrow \epsilon$	$F' \rightarrow \epsilon$	$F' \rightarrow \epsilon$	$F' \rightarrow \epsilon$	$F' \rightarrow *F'$	$F' \rightarrow \epsilon$	$F' \rightarrow \epsilon$
P	$P \rightarrow a$	$P \rightarrow b$	$P \rightarrow \wedge$	$P \rightarrow (E)$				

(注: 空白处均为 ERROR)

2 解:

(2) $FIRST(aABbcd) = \{ a \}$; $FIRST(\epsilon) = \{ \epsilon \}$;

$FIRST(Asd) = \{ a, d \}$; $FIRST(SAh) = \{ a, h \}$;

$FIRST(eC) = \{ e \}$; $FIRST(Sf) = \{ a, f \}$;

$FIRST(Cg) = \{ g, a, f \}$; $FIRST(aBD) = \{ a \}$;

(3) 对于 $C \rightarrow Sf \mid Cg$, $FIRST(Sf) \cap FIRST(Cg) = \{ a, f \}$

此文法不是 LL(1) 文法。 ~~✗~~

解： : * *

本题错误。

6. 解：

一个文法是文法的充要条件是： $\forall A \in V_n$, A 的任何两条不同的规则 $A::=\alpha \mid \beta$ 有下列条件成立：

$$\textcircled{1} \text{FIRST}(\alpha) \cap \text{FIRST}(\beta) = \Phi$$

$$\textcircled{2} \beta \Rightarrow \epsilon, \text{ 则 } \text{FIRST}(\alpha) \cap \text{FOLLOW}(A) = \Phi$$

证明：充分性：

对于任意非终结符 A 若 $A::=\alpha \mid \beta$ 满足上述条件，取分析表项 $M[A, a]$, $a \in V_t$

若 $A \Rightarrow a \dots$ 假设 $\alpha \Rightarrow a \dots$ 即 $a \in \text{FIRST}(\alpha)$

$$\therefore \text{FIRST}(\alpha) \cap \text{FIRST}(\beta) = \Phi$$

$$\therefore a \in \text{FIRST}(\beta) \quad \text{分析表项 } M[A, a] = A::=\alpha$$

$$\text{若 } \beta \Rightarrow \epsilon, \text{ 且 } a \in \text{FOLLOW}(A) \quad a \in \text{FIRST}(\alpha)$$

$$M[A, a] = A::=\epsilon \quad \text{否则 } M[A, a] = \text{error}$$

综上，分析表的元素无多重定义，符合 LL(1) 文法定义，是 LL(1) 文法
必要性：

令对于 LL(!) 文法 G 的 $\forall A \in V_n$, $A::=\alpha \mid \beta$ 条件不成立

$$\text{FIRST}(\alpha) \cap \text{FIRST}(\beta) = B \neq \Phi \quad \text{FIRST}(\alpha) \cap \text{FIRST}(\beta) = C \neq \Phi$$

若 $a \in B$, 则 $M[A, a]$ 中，可同时存在 $A::=\alpha$ 及 $A::=\beta$

若 $\beta \Rightarrow \epsilon$ 则 $M[A, a]$ 中，可同时存在 $A::=\alpha$ $A::=\epsilon$ 两条规则

这与定义相矛盾，假设错误

必要性得证。

P100

3. 解：

```

procedure INSERT(u, a)
    if not L[u, a] then
        begin
            L[u, a] := true;
            将 (u, a) 压入栈 STACK
        end;
program MAIN;
begin
    for 每个非终结符 U 和终结符 a do
        L[u, a] := false;
    For 每条形如  $U::=\dots a$  或  $U::=\dots aV$  规则 do
        INSERT(U, a);
    While STACK 非空 DO
        Begin
            将 STACK 栈顶弹出，记为 (V, a);
            for 每条形如  $U::=\dots V$  规则 do
                INSERT(U, a);
        End;
    End
end

```

P109

2. 解：

$$IO = \text{closure}(\{S \rightarrow \cdot E\}) = \{S \rightarrow \cdot E\#, E \rightarrow \cdot wX, E \rightarrow \cdot xY\}$$

$I1 = \text{goto}(I0, E) = \text{closure}(\{S \rightarrow E. \# \}) = \{S \rightarrow E. \# \}$

$I2 = \text{goto}(I0, w) = \text{closure}(\{E \rightarrow w. X \}) = \{E \rightarrow w. X, X \rightarrow. yX, X \rightarrow. z \}$

$I3 = \text{goto}(I0, x) = \text{closure}(\{E \rightarrow x. Y \}) = \{E \rightarrow x. Y, Y \rightarrow. yY, Y \rightarrow. z \}$

$I4 = \text{goto}(I1, \#) = \text{closure}(\{S \rightarrow E\# \}) = \{S \rightarrow E\# \}$

$I5 = \text{goto}(I2, X) = \text{closure}(\{E \rightarrow wX \}) = \{E \rightarrow wX \}$

$I6 = \text{goto}(I2, y) = \text{closure}(\{X \rightarrow y. X \}) = \{X \rightarrow y. X, X \rightarrow. yX, X \rightarrow. z \}$

$I7 = \text{goto}(I2, z) = \text{closure}(\{X \rightarrow. z \}) = \{X \rightarrow. z \}$

$I8 = \text{goto}(I3, Y) = \text{closure}(\{E \rightarrow xY \}) = \{E \rightarrow xY \}$

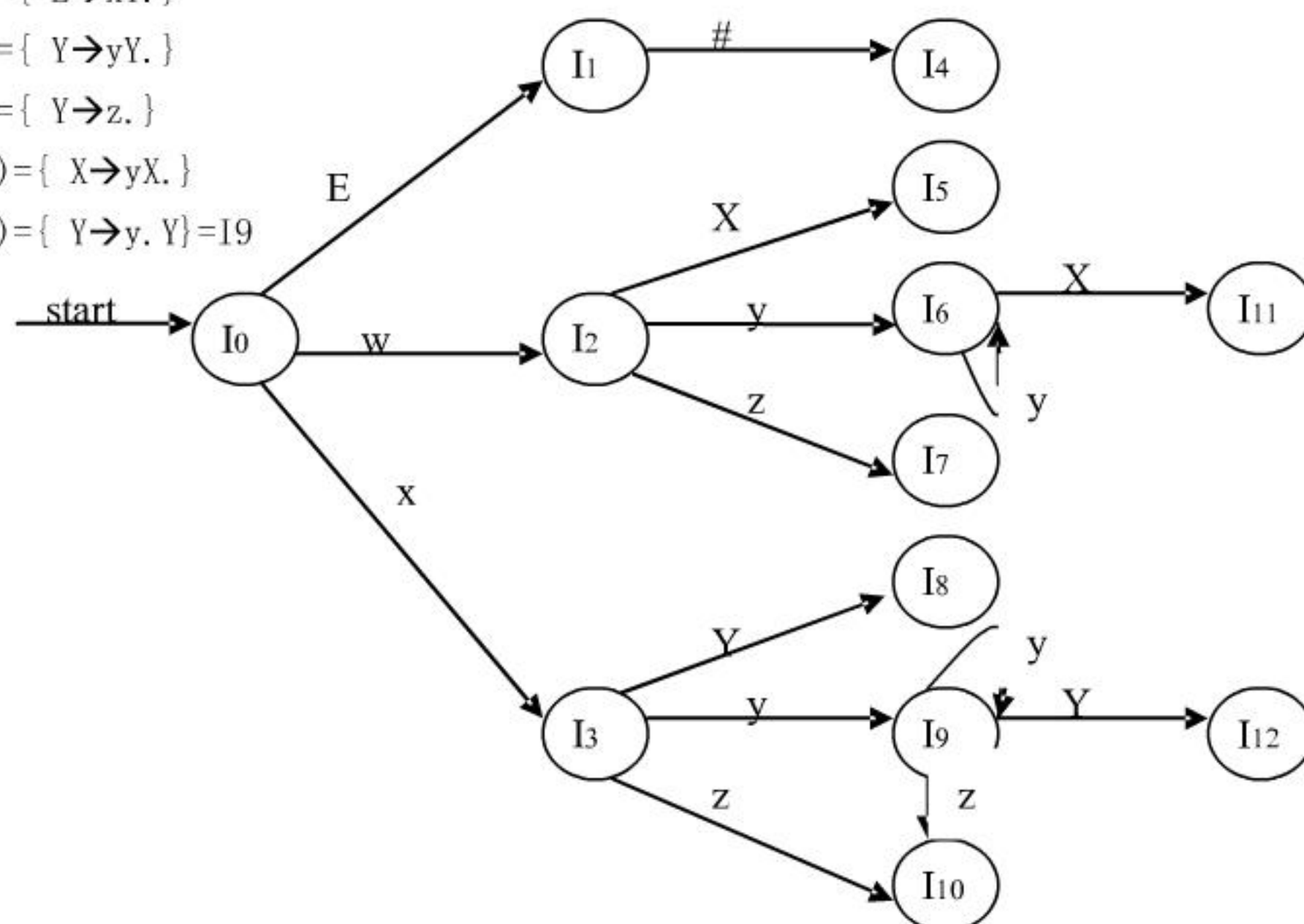
$I9 = \text{goto}(I3, y) = \text{closure}(\{Y \rightarrow y. Y \}) = \{Y \rightarrow y. Y \}$

$I10 = \text{goto}(I3, z) = \text{closure}(\{Y \rightarrow. z \}) = \{Y \rightarrow. z \}$

$I11 = \text{goto}(I6, X) = \text{closure}(\{X \rightarrow yX \}) = \{X \rightarrow yX \}$

$I12 = \text{goto}(I9, Y) = \text{closure}(\{Y \rightarrow y. Y \}) = \{Y \rightarrow y. Y \} = I9$

LR(0)机器为:



3.解:

步骤	栈内容	输入串	活前缀	句柄
1	0	wyyz#		
2	0w2	yyz#	w	
2	0w2y6	yz#	wy	
3	0w2y6y6	z#	wyy	
4	0w2y6y6z7	#	wyyz	z
5	0w2y6y6X11	#	wyyx	yx
6	0w2y6X11	#	wyx	yx
7	0w2X5	#	wx	wx
8	0E1	#	E	
9	0E1#4	#	E#	E#
10	0	accept		

步骤	栈内容	输入串	活前缀	句柄
0	0	xyyyz#		
1	0x3	yyyz#	x	
2	0x3y9	yyz#	xy	
3	0x3y9y9	yz#	xyy	
4	0x3y9y9y9	z#	xyyy	
5	0x3y9y9y9z10	#	xyyyz	z
6	0x3y9y9y9Y12	#	xyyyY	yY
7	0x3y9y9Y12	#	xyyY	yY

8	0x3y9Y12	#	xyY	yY
9	0x3Y8	#	xY	xY
10	0E1	#	E	
11	0E1#4	#	E#	E#
12	0	accept		

P116

1. 给定下列产生式的文法:

$S \rightarrow E\# \quad E \rightarrow T \mid E+T \quad T \rightarrow P \mid T*P \quad P \rightarrow F \mid F \uparrow P \quad F \rightarrow i \mid (E)$

为该文法构造 LR(0) 机器。

证明该文法是 SLR(1)。

为该文法构造 SLR(1) 分析表。

解: (a) 构造 LR(0) 机器

C0: $\{S \rightarrow \cdot E\#, E \rightarrow \cdot T, E \rightarrow \cdot E+T, T \rightarrow \cdot P, T \rightarrow \cdot T*P, P \rightarrow \cdot F, P \rightarrow \cdot F \uparrow P, F \rightarrow \cdot i, F \rightarrow \cdot (E)\}$

C1: $\{S \rightarrow E \cdot \#, E \rightarrow E \cdot +T\}$

C2: $\{E \rightarrow T \cdot, T \rightarrow T \cdot *P\}$

C3: $\{T \rightarrow P \cdot\}$

C4: $\{P \rightarrow F \cdot, P \rightarrow F \cdot \uparrow P\}$

C5: $\{F \rightarrow i \cdot\}$

C6: $\{F \rightarrow (\cdot E), E \rightarrow \cdot T, E \rightarrow \cdot E+T, T \rightarrow \cdot P, T \rightarrow \cdot T*P, P \rightarrow \cdot F, P \rightarrow \cdot F \uparrow P, F \rightarrow \cdot i, F \rightarrow \cdot (E)\}$

C7: $\{S \rightarrow E\# \cdot\}$

C8: $\{E \rightarrow E+ \cdot T, T \rightarrow \cdot P, T \rightarrow \cdot T*P, P \rightarrow \cdot F, P \rightarrow \cdot F \uparrow P, F \rightarrow \cdot i, F \rightarrow \cdot (E)\}$

C9: $\{T \rightarrow T* \cdot P, P \rightarrow \cdot F, P \rightarrow \cdot F \uparrow P, F \rightarrow \cdot i, F \rightarrow \cdot (E)\}$

C10: $\{P \rightarrow F \uparrow \cdot P, P \rightarrow \cdot F, P \rightarrow \cdot F \uparrow P, F \rightarrow \cdot i, F \rightarrow \cdot (E)\}$

C11: $\{F \rightarrow (E) \cdot, E \rightarrow E \cdot +T\}$

C12: $\{E \rightarrow E+T \cdot, T \rightarrow T \cdot *P\}$

C13: $\{T \rightarrow T*P \cdot\}$

C14: $\{P \rightarrow F \uparrow P \cdot\}$

C15: $\{F \rightarrow (E) \cdot\}$

(b) 对于状态 C2: $\{E \rightarrow T \cdot, T \rightarrow T \cdot *P\}$

$\text{Follow}(E) = \{\#, +, >\}$ *不属于 $\text{Follow}(E)$

C4: $\{P \rightarrow F \cdot, P \rightarrow F \cdot \uparrow P\}$

$\text{Follow}(P) = \{\#, +, >, *\}$ ↑ 不属于 $\text{Follow}(P)$

C12: $\{E \rightarrow E+T \cdot, T \rightarrow T \cdot *P\}$

$\text{Follow}(E) = \{\#, +, >\}$ *不属于 $\text{Follow}(E)$

∴ 该文法构成的分析表不会出现多重定义出口。它是 SLR(1) 文法。

(c) 构造分析表为:

状态	S	E	T	P	F	+	*	↑	i	()	#
C0		C1	C2	C3	C4				S5	S6		
C1						S8						S7
C2						r1	S9				r1	r1
C3						r3	r3				r3	r3
C4						r5	r5	S10			r5	r5
C5						r7	r7	r7			r7	r7
C6		C11	C2	C3	C4				S5	S6		
C7												A
C8			C12	C3	C4				S5	S6		
C9				C13	C4				S5	S6		

C10				C14	C4				S5	S6		
C11						S8						
C12						r2	S9				S15	
C13						r4	r4				r4	r4
C14						r6	r6				r6	r6
C15						r8	r8	r8			r8	r8

(d) 分析 $i+i \uparrow (i*i)\#$

步骤	栈内容	输入串	活前缀	句柄
0	0	$i+i \uparrow (i*i)\#$		
1	0i5	$+i \uparrow (i*i)\#$	I	i
2	0F4	$+i \uparrow (i*i)\#$	F	F
3	0P3	$+i \uparrow (i*i)\#$	P	P
4	0T2	$+i \uparrow (i*i)\#$	T	T
5	0E1	$+i \uparrow (i*i)\#$	E	
6	0E1+8	$i \uparrow (i*i)\#$	E+	i
7	0E1+8i5	$\uparrow (i*i)\#$	E+i	
8	0E1+8F4	$\uparrow (i*i)\#$	E+F	
9	0E1+8F4 \uparrow 10	$(i*i)\#$	E+F \uparrow	
10	0 E1+8F4 \uparrow 10(6	$i*i)\#$	E+F \uparrow (
11	0 E1+8F4 \uparrow 10(6i5	$*i)\#$	E+F \uparrow (i	i
12	0 E1+8F4 \uparrow 10(6F4	$*i)\#$	E+F \uparrow (F	F
13	0 E1+8F4 \uparrow 10(6P3	$*i)\#$	E+F \uparrow (P	P
14	0 E1+8F4 \uparrow 10(6T2	$*i)\#$	E+F \uparrow (T	
15	0 E1+8F4 \uparrow 10(6T2*9	$i)\#$	E+F \uparrow (T*	
16	0 E1+8F4 \uparrow 10(6T2*9i5)#	E+F \uparrow (T*i	i
17	0 E1+8F4 \uparrow 10(6T2*9F4)#	E+F \uparrow (T*F	F
18	0 E1+8F4 \uparrow 10(6T2*9P13)#	E+F \uparrow (T*P	T*P
19	0 E1+8F4 \uparrow 10(6T2)#	E+F \uparrow (T	T
20	0 E1+8F4 \uparrow 10(6E11)#	E+F \uparrow (E	
21	0 E1+8F4 \uparrow 10(6E11)15	#	E+F \uparrow (E)	(E)
22	0 E1+8F4 \uparrow 10F4	#	E+F \uparrow F	F
23	0 E1+8F4 \uparrow 10P14	#	E+F \uparrow P	F \uparrow P
24	0 E1+8P3	#	E+P	P
25	0 E1+8T12	#	E+T	E+T
26	0E1	#	E	
27	0E1#7		E#	E#
28	0S0	accept		

分析 $(i*i) \uparrow (i+i)$

步骤	栈内容	输入串	活前缀	句柄
0	0	$(i*i) \uparrow (i+i)\#$		
1	0(6	$i*i) \uparrow (i+i)\#$	(
2	0(6i5	$*i) \uparrow (i+i)\#$	(i	i
3	0(6F5	$*i) \uparrow (i+i)\#$	(F	F
4	0(6P3	$*i) \uparrow (i+i)\#$	(P	P
5	0(6T2	$*i) \uparrow (i+i)\#$	(T	
6	0(6T2*9	$i) \uparrow (i+i)\#$	(T*	

7	0(6T2*9i5) ↑ (i+i)#	(T*i	i
8	0(6T2*9F4) ↑ (i+i)#	(T*F	F
9	0(6T2*9P13) ↑ (i+i)#	(T*P	T*P
10	0(6T2) ↑ (i+i)#	(T	T
11	0(6TE11) ↑ (i+i)#	(E	
12	0(6TE11)5	↑ (i+i)#	(E)	(E)
13	0F4	↑ (i+i)#	F	
14	0F4 ↑ 10	(i+i)#	F ↑	
15	0 F4 ↑ 10(6	i+i)#	F ↑ (i
16	0 F4 ↑ 10(6i5	+i)#	F ↑ (i	F
17	0 F4 ↑ 10(6F4	+i)#	F ↑ (F	P
18	0 F4 ↑ 10(6P3	+i)#	F ↑ (P	T
19	0 F4 ↑ 10(6T2	+i)#	F ↑ (T	
20	0 F4 ↑ 10(6E11	+i)#	F ↑ (E	
21	0 F4 ↑ 10(6E11+8	i)#	F ↑ (E+	
22	0 F4 ↑ 10(6E11+8i5)#	F ↑ (E+i	i
23	0 F4 ↑ 10(6E11+8F4)#	F ↑ (E+F	P
24	0 F4 ↑ 10(6E11+8P3)#	F ↑ (E+P	P
25	0 F4 ↑ 10(6E11+8T2)#	F ↑ (E+T	E+T
26	0 F4 ↑ 10(6E11)#	F ↑ (E	
27	0 F4 ↑ 10(6E11)15	#	F ↑ (E)	(E)
28	0 F4 ↑ 10F4	#	F ↑ F	F
29	0 F4 ↑ 10P14	#	F ↑ P	F ↑
30	0 P3	#	P	P
31	0T2	#	T	T
32	0E1	#	E	
33	0E1#7		E#	E#
34	0S0	Accept		

3.解:

证明: 首先构造文法的 LR(0)

$V = \{S, P, B, H, T, C, b, d, s, e, ;, \# \}$

$I_0 = \text{closure}(\{S \rightarrow .P\# \}) = \{S \rightarrow .P\#, P \rightarrow .C, P \rightarrow .B, C \rightarrow .bT, B \rightarrow .H; T, h \rightarrow .bd, H \rightarrow .H; d\}$

$I_1 = \text{goto}(I_0, P) = \text{closure}(\{S \rightarrow P.\# \}) = \{S \rightarrow P.\#\}$

$I_2 = \text{goto}(I_0, C) = \{P \rightarrow C.\}$

$I_3 = \text{goto}(I_0, B) = \{P \rightarrow B.\}$

$I_4 = \text{goto}(I_0, b) = \{C \rightarrow b.T, H \rightarrow b.d, T \rightarrow .Se, T \rightarrow .S; T\}$

$I_5 = \text{goto}(I_0, H) = \{B \rightarrow H.; T, H \rightarrow H.; d\}$

$I_6 = \text{goto}(I_1, \#) = \{S \rightarrow p\#.\}$

$I_7 = \text{goto}(I_4, T) = \{C \rightarrow bT.\}$

$I_8 = \text{goto}(I_4, d) = \{H \rightarrow bd.\}$

$I_9 = \text{goto}(I_4, S) = \{T \rightarrow S.e, T \rightarrow S.; T\}$

$I_{10} = \text{goto}(I_5, ;) = \{B \rightarrow H.; T, H \rightarrow H.; d, T \rightarrow .Se, T \rightarrow .S; T\}$

$I_{11} = \text{goto}(I_9, e) = \{T \rightarrow Se.\}$

$I_{12} = \text{goto}(I_9, ;) = \{T \rightarrow S.; T, T \rightarrow .Se, T \rightarrow .S; T\}$

$I_{13} = \text{goto}(I_{10}, T) = \{H \rightarrow H; T.\}$

$I_{14} = \text{goto}(I_{10}, d) = \{H \rightarrow H; d.\}$

$I_{15} = \text{goto}(I_{12}, T) = \{T \rightarrow S; T.\}$

由其状态集可知任何一状态都无局部二义性是 LR(0)文法，当然包含于 SLR(1)文法中。

SLR(1)分析器如下：

状态	;	b	d	s	e	#	P	B	H	T	C
0		S4					1	3	5		2
1						S6					
2						r1					
3						r2					
4			S8	S9						7	
5	S10										
6						accept					
7						r8					
8	r4										
9	S12				S11						
10			S14	S9						13	
11						r6					
12				S9						15	
13						r3					
14	r5										
15						r7					

5. 证明下列文法不是 SLR (1) 文法：

$S \rightarrow E\# \quad E \rightarrow bEa \mid aEb \mid ba$

证明：构造项目集为：

C0: $\{S \rightarrow \cdot E\#, E \rightarrow \cdot bEa, E \rightarrow \cdot aEb, E \rightarrow \cdot ba\}$

C1: $\{S \rightarrow E \cdot \#\}$

C2: $\{S \rightarrow E\# \cdot\}$

C3: $\{E \rightarrow b \cdot Ea, E \rightarrow \cdot bEa, E \rightarrow \cdot aEb, E \rightarrow \cdot ba, E \rightarrow b \cdot a\}$

C4: $\{E \rightarrow a \cdot Eb, E \rightarrow \cdot bEa, E \rightarrow \cdot aEb, E \rightarrow \cdot ba\}$

C5: $\{E \rightarrow bE \cdot a\}$

C6: $\{E \rightarrow ba \cdot, E \rightarrow a \cdot Eb, E \rightarrow \cdot bEa, E \rightarrow \cdot aEb, E \rightarrow \cdot ba\}$

C7: $\{E \rightarrow aE \cdot b\}$

C8: $\{E \rightarrow aEb \cdot\}$

C9: $\{E \rightarrow bEa \cdot\}$

$\because \text{Follow}(E) = \{\#, a, b\}$ ，且在状态 C6 中，同时存在 $E \rightarrow ba \cdot$ 与 $E \rightarrow \cdot bEa$

\therefore 存在归约-移进冲突。

\therefore 此文法不是 SLR(1) 文法。

第五章

P138.

1. 解：由中缀表达式翻译成波兰前缀表达式的符号串翻译文法如下：

① $E \rightarrow @+E+T$ ② $E \rightarrow T$ ③ $T \rightarrow @*T*F$

④ $T \rightarrow F$ ⑤ $F \rightarrow (E)$ ⑥ $F \rightarrow a@a$

⑦ $F \rightarrow b@b$ ⑧ $F \rightarrow c@c$

2. 解：

a) 输入符号串的倒置：

① $S \rightarrow @0S0$ ② $S \rightarrow @1S1$ ③ $S \rightarrow 1@1$

④ $S \rightarrow 0@0$

c) 输入符号串本身：

① $S \rightarrow S0@0$ ② $S \rightarrow S1@1$ ③ $S \rightarrow 1@1$

④ $S \rightarrow 0@0$

3. 解:

该符号串翻译文法能将输入的符号序列“ENGLISH”翻译成动作序列“@C@HI@N@E@SE”，并执行该动作序列得到输出序列“CHINESE”。

4. 解:

根据活动序列将动作符号放到文法中适当位置得到该翻译文法为:

① $\langle s \rangle \rightarrow @q a \langle s \rangle @x \langle s \rangle @y$ ② $\langle s \rangle \rightarrow @x @y b @z$

5. 解:

由该符号串翻译文法得到其输入文法为:

① $\langle s \rangle \rightarrow \langle A \rangle c \langle B \rangle$ ② $\langle s \rangle \rightarrow dcb$
 ③ $\langle A \rangle \rightarrow \langle B \rangle a$ ④ $\langle A \rangle \rightarrow d$ ⑤ $\langle B \rangle \rightarrow b$

由推导: $s \Rightarrow \langle A \rangle c \langle B \rangle \Rightarrow \langle A \rangle cb \Rightarrow dcb$ 可得到对偶 $\langle dcb, xxy \rangle$

由推导: $s \Rightarrow dcb$ 可得到对偶 $\langle dcb, yxz \rangle$

由推导: $s \Rightarrow \langle A \rangle c \langle B \rangle \Rightarrow \langle A \rangle cb \Rightarrow \langle B \rangle acb \Rightarrow bacb$ 可得到对偶 $\langle dcb, xyxxy \rangle$

第六章 符号表管理技术

4、下面是一个非分程序结构语言的程序段。画出编译该程序段时将生成的有序符号表。

```
BLOCK
REAL  X, Y, Z1, Z2, Z3;
INTEGER  I, J, K, LASTI;
STRING  LIST-OF-NAMES;
LOGICAL  ENTRY-ON, EXIT-OFF;
ARRAY  REAL  VAL (20);
ARRAY  INTEGER  MIN-VAL-IND (20);
      .
      .
      .
END OF BLOCK
```

解:

变量名	类型	维数
ENTRY-ON	LOGICAL	0
EXIT-OFF	LOGICAL	0
I	INTEGER	0
J	INTEGER	0
K	INTEGER	0
LASTI	INTEGER	0
LIST-OF-NAMES	STRING	0
MIN-VAL-IND	INTEGER	1
VAL	REAL	1
X	REAL	0
Y	REAL	0
Z1	REAL	0
Z2	REAL	0
Z3	REAL	0

6、下面是一分程序结构的程序段。花处在程序段 3 和 4 的编译即将完成以前，栈式符号表的图形（包括有效部分和失效部分）。

解:

```
BBLOCK;
  REAL  Z;  INTEGER  Y;
  SUB-1  PBLOCK (INTEGER  J);
  BBLOCK;
```

```

        ARRAY  STRING  S (J+2);
        LOGICAL  FLAG;  INTEGER  Y;
    END;
    SUB-2  PBLOCK (REAL  W);
        REAL  J;  LOGICAL  TEST1, TEST2, TEST3;
    END SUB-2;
END SUB-1;
END;

```

程序段 3 编译即将完成以前:

	Y	INTEGETR		
	FLAG	LOGICAL		
5	S	STRING	←	
4	J	INTEGETR	←	
3	SUB-1	PBLOCK		5
2	Y	INTEGETR		4
1	Z	REAL	←	1

程序段 4 编译即将完成以前:

10	TEST3	LOGICAL		
9	TEST2	LOGICAL		
8	TEST1	LOGICAL		
7	J	REAL		
6	W	REAL	←	
5	SUB-2	PBLOCK		
4	J	INTEGER	←	
3	SUB-1	PBLOCK		6
2	Y	INTEGER		4
1	Z	REAL	←	1

第七章 运行时的存贮组织及管理

1、考虑下面的类 ALGOL 程序:

```

BEGIN
    INTEGER  I;
    REAL  X, Y;
    READ (I, X, Y);
    .
    .
    .
    PROCEDURE  EXAMPLEPROC (INTEGER J, REAL Z, Y)
    BEGIN
        STRING S(8);
        .
    
```

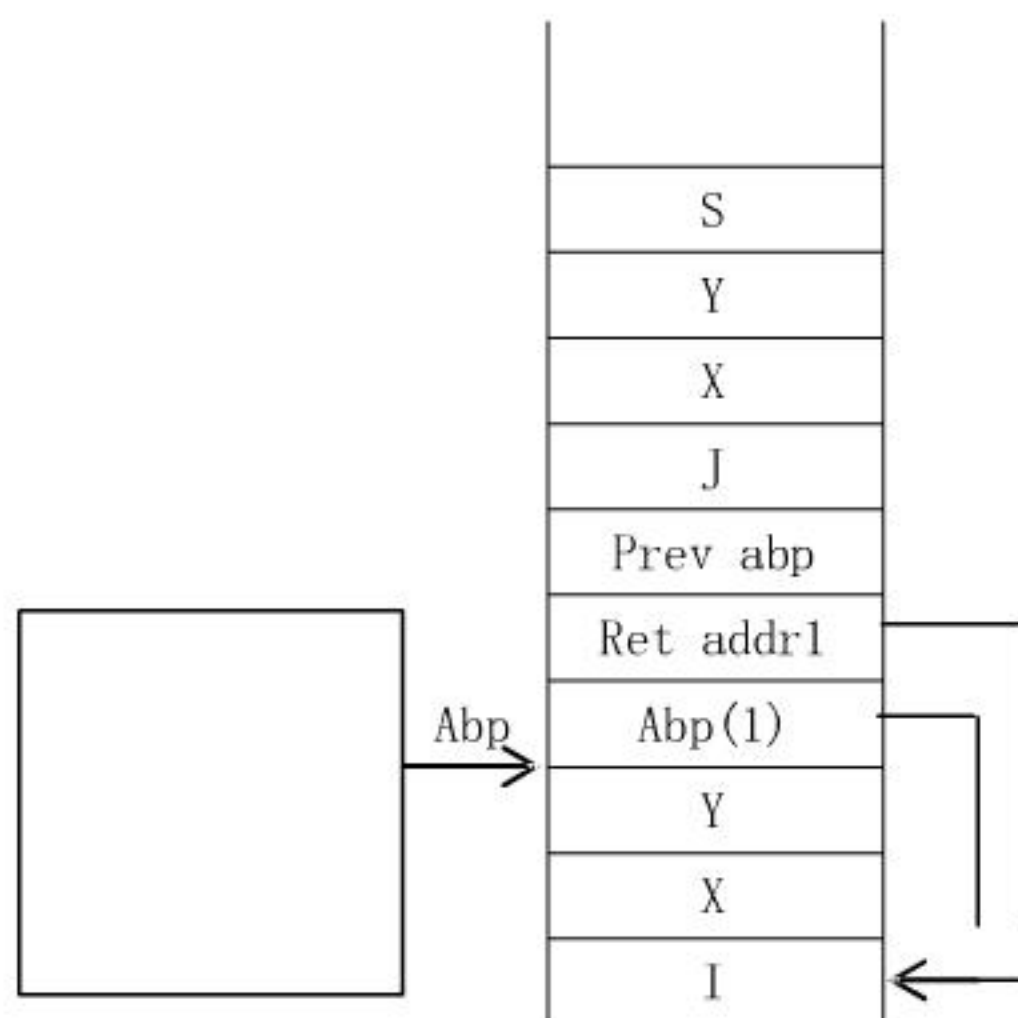
```

      BEGIN
        REAL (J, J)
        .
        .
        .
      END
    END
  END

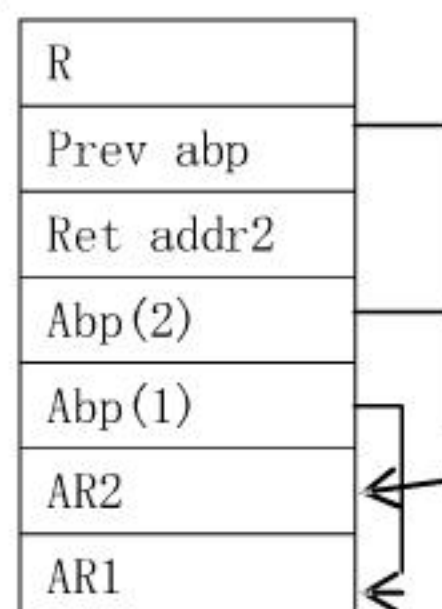
```

画出当程序执行到点 和点 时,运行栈内容的图象。

解：运行到处运行栈的内容为：



运行到 处运行栈的内容为:



编译原理科目分析和解题方法（转贴）

初步了解编译原理

考研的朋友很多都是计算机专业高年级（或者同等水平）的学生，对于编程已经不再陌生，甚至我们很多人都要把它作为一门安身立命的技术。这里先给大家出一个很小的题目，它不是考研试题，而是一个公司在一次招聘时出的考试题目，很有意思，题目是这样的：

以 DOS 为操作系统，使用 C 语言，编写一个函数，功能是取得当时的 BP 寄存器的值，只能使用标准的 C 语句（也就是不能使用扩充的宏等其他手段）。

如果大家在面试的时候遇到这道题目，头脑中能够马上反应出这到底考的是什么吗？这里实际上考的是编译程序的存储组织。

下面我们简单分析一下该题。我们知道 C 语言是栈式存储分配，局部变量的分配和参数传递都是通过栈来完成的，在调用函数的时候，首先把被调函数的参数从右向左依次压入栈，再把返回地址压入栈，然后转入被调函数内部，进入被调函数以后，首先执行固定的两条指令：**PUSH BP** **MOVE BP, SP**

这样，通过 BP 寄存器，加上或者减去一定的偏移量，就可以对传递来的参数和函数内新分配的局部变量进行寻址了。到这里，我们就找到答案了，这个题目所要求的函数内不需要做任何实际的工作，只需要声明一个整数类型的变量，那么显然这个变量就在栈顶，而它的下面的两个字节保存的就是在调用这个函数之前 BP 寄存器的值（也就是上面说的指令 PUSH BP 所保存的内容），将这个值返回就可以了，也就是把这个整型变量的地址加 2 后的内容返回即可。

这里举这个小例子就是为了说明编译原理对于编程的重要性，它可以帮助我们更深刻地理解程序运行时内部的机制。

编译原理不好学

这里我们回顾一下我们学习编程的历程：最初通常是学习一门高级语言，比如 C 或者 Pascal，这样就掌握了编写一些简单程序的方法；然后又学习了数据结构，这是一门非常重要的课程，学习了数据结构，我们的头脑中建立了“算法”的概念，对编程有了更深的理解，遇到问题的时候，我们知道应该寻找相应的数据结构模型，设计适当的算法来解决问题；此外，我们通常还要学习汇编语言，尽管大多数人并不会在实际中用汇编语言来进行开发，但实际上，这门课程是我们真正深入了解计算机内部的第一门课程。通过学习汇编语言，我们知道了每一条指令是如何在计算机内部运行的。而此时，我们已经可以编写很多程序了，但是我们还不知道高级语言最终是如何成为汇编语言，甚至机器语言的，这时，编译原理课程就可以帮助我们了解高级语言变成汇编语言的过程。更进一步，要想了解高级语言是如何变为机器语言，以及如何在 CPU 内部运行的，就要通过计算机组成原理的课程的学习了。这样，我们就可以清楚地认识到编译原理在计算机学科中所处的重要位置了。同时，我们也可以看出，它的复杂性和学习的难度都是很大的。

1 编译原理不好学

编译原理课程的一个明显特点是它的理论性很强，在学习的时候需要很强的逻辑思维能力。这门课程初看起来内容很多，而且又非常抽象，如果和“数据结构”比起来，就会发现，编译程序所涉及的算法要复杂得多，而且要深入地理解这些算法也很困难。比如在学习数据结构的时候，我们也会接触很多算法，经常会跟踪算法的执行过程，比如遍历二叉树等。而跟踪编译程序中的算法则要麻烦得多，从而要想深刻理解算法的思想就很困难。编译程序的构造方法确实非常精妙，就像一部走时精确的时钟，很多齿轮、部件协调地运转，以驱动指针准确地旋转；编译程序也是如此，一边扫描源程序，一边经过各个部件的运算，准确地输出为目标语言。

2 编译原理也不难学

上面说了编译原理不好学的原因，但如果我们弄清了这门课程的特点和规律，以及它作为一门考试科目的特点，就会发现攻克编译原理其实也不难：

1. 编译原理课程各个部分之间的独立性很强，实际上就包括了词法分析、语法分析、存储的组织与分配、中间语言、语法制导翻译、代码生成与优化这几大部分。而且其中前两个部分是其中的重点，也是难点，需要掌握比较复杂的算法逻辑；其他部分相对来说知识性更强一些。另外，编译原理每个部分之间的方法也互相独立，在复习的时候，便于逐个击破。

2. 实际上编译原理考试考查的内容相对来说是很稳定的，而且绝大多数题目的解法都非常机械，如果熟练掌握，考试的时候沉着冷静，不要粗心，做对是非常容易的。

具体举例说明一下，比如词法分析这一部分，如果仔细研究之后，就会发现，最重要的就是要掌握图 0.1 中的 6 个转换关系。比如给出一个正规式，然后要依次求出相应的 NFA、DFA、最小化的 DFA 等，而这 6 种转换关系都有确定的方法，只要熟练掌握了相应的方法，无论怎么出题都应该能够套用。

再比如语法分析部分，这部分是整个编译原理考试中比重最大、难度也最大的一个部分。但仔细分析之后，我们会发现，语法分析只有两大类：自顶向下和自底向上，每类中又有几种不同的方法，每种方法都对应于一类文法。因此在复习的时候，首先弄清楚每种方法适用的具体条件，具体解法套用相应的步骤就可以了。

下面以自底向上得 LR 分析法为例加以说明。LR 分析法通常包括了 4 种具体的分析方法：LR(0)、SLR(1)、LR(1)、LALR(1)分析法，这 4 种 LR 分析法的原理如图 0.2 所示。

这 4 种 LR 分析法的总控程序是相同的，只是分析表不同。因此复习的关键是掌握构造分析表的方法，以及使用分析表进行分析的方法。LR(0)是 LR 分析法的基础，首先要深刻地理解 LR(0)分析法的思想，熟练掌握 LR(0)文法的判断方法和 LR(0)分析表的构造方法，这里再次强调，其方法是非常机械的，只要多加练习，就可以掌握得非常牢固，只有这样，在考场上才能游刃有余。

在掌握了 LR(0)分析法后，就可以看出 LR(0)分析法要求构造出的识别可归前缀的有穷自动机的各个状态不能有冲突项目，而 SLR(1)分析可以部分地解决这个问题，其方法是向前多看一个输入符号，这样又需要掌握 SLR(1)分析法，同样要注意其适用的文法限制和分析表的构造方法。要注意 SLR(1)只是部分解决了 LR(0)的缺陷，当我们需要进一步加强分析法的能力时，就到达了 LR(1)分析法和 LALR(1)分析法，大体的复习思路都是相同的。

这 4 种分析法都掌握后，最好仔细总结一下其内在的关系，使用的文法限制等。总之，这部分要掌握每种分析法适用的文法，并能构造出相应分析表，以及应该做到当给出一个该文法的句子，能够跟踪使用分析表进行分析。

其他各个章节也都有类似的特点，大家在复习的时候应该认真地进行总结，提纲挈领地抓出线索，熟练掌握具体的方法。一定要多做习题，千万不能觉得自己知道方法了，就不用亲手做了。尤其对于编译这样的学科，题目的规模很大，步骤繁多，而且前面的步骤一旦出错，后面都错。如果不熟练掌握的话，考试的时候必定慌乱，所以切忌眼高手低，一定要多加练习，才能到考场上沉着应对。

客观来说，如果这些大的方法（如 DFA 相关的转换、LR 分析法）都掌握了，即使对编译原理本身的理解并不十分深刻，也可以应对考试。因为在考试中绝大部分都是常规题目，抓住这些常规题是效率最高的复习方法。