

第二章部分习题答案

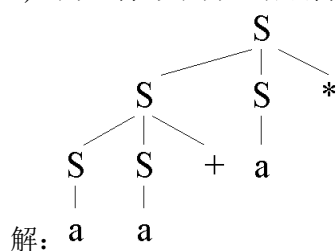
2.1 考虑文法

$S \rightarrow SS+ | SS* | a$

a) 证明文法可生成符号串 $aa+a^*$

解: $S \rightarrow \underline{S}S^* \rightarrow \underline{S}S+S^* \rightarrow a\underline{S}+S^* \rightarrow aa+\underline{S}^* \rightarrow aa+a^*$

b) 为此符号串构造语法树



c) 文法生成什么样的语言? 证明结论

解: 将 a 看作运算数, 文法生成语言 $L=\{\text{支持加法、乘法的表达式\text{的后缀表示形式}}\}$

证明类似 2.2 题 b)

=====

2.2 下列文法生成什么样的语言? 证明你的结论。是否有二义性?

a) $S \rightarrow 0S1 \mid 01$

解: 生成语言 $L=\{0^n1^n \mid n \geq 1\}$

证明: 1) 证文法推导出的符号串都在 L 中

i) 考虑最小语法树 $\begin{array}{c} s \\ / \quad \backslash \\ 0 \quad 1 \end{array}$, 推导出的符号串 01 显然 $\in L$

ii) 假定结点数 $< n$ 的语法树对应的符号串都 $\in L$,



子树 S_1 结点数 $< n$, 因此对应符号串 $t_1 \in L$,

S 对应符号串为 $t=0t_11$, 因此 $t \in L$

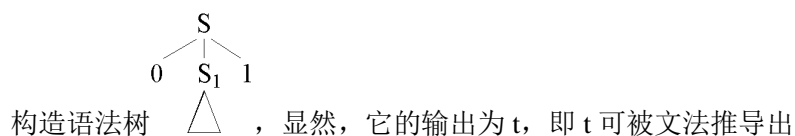
综合 i)、ii), 1)得证

2) 证 L 中符号串都可由文法推导出

i) L 中最短符号串 01 , 显然可由文法推导出

ii) 假定 L 中长度 $< 2n$ 的符号串都可由文法推导出,
考虑长度 $= 2n$ 的符号串 $t=0^n1^n$, 它可表示为 $0t_11$,

$t_1 \in L$ 且长度 $< 2n$, 因此它可被文法推导出, 对应语法树 \triangle_{S_1} ,



综合 i)、ii), 2)得证

综合 1)、2), 文法生成的语言即为 L

另外, 文法没有二义性

=====

b) $S \rightarrow +SS \mid -SS \mid a$

解: 生成语言 $L = \{\text{支持加法、减法的表达式的前缀表示形式}\}$

证明: 1) 证文法推导出的符号串都在 L 中

- i) 考虑最小语法树 \triangle_a , 推导出的符号串 a 显然 $\in L$
- ii) 假定结点数 $< n$ 的语法树对应的符号串都 $\in L$,



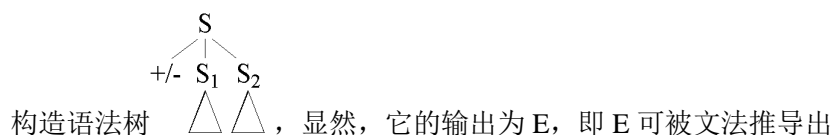
子树 S_1 、 S_2 结点数 $< n$, 因此对应符号串 E_1 、 $E_2 \in L$ (前缀表达式),
 S 对应符号串为 $E = +/- E_1 E_2$, E 也是前缀表达式。

综合 i)、ii), 1)得证

2) 证 L 中符号串都可由文法推导出

- i) L 中最短符号串 a , 显然可由文法推导出
- ii) 假定 L 中长度 $< n$ 的符号串都可由文法推导出,
 考虑长度 $= n$ 的符号串 E , 它的形式必为 $+/- E_1 E_2$,

E_1 、 $E_2 \in L$ 且长度 $< n$, 因此可被文法推导出, 对应语法树 \triangle_{S_1} 、 \triangle_{S_2} ,



综合 i)、ii), 2)得证

综合 1)、2), 文法生成的语言即为 L

另外, 文法没有二义性

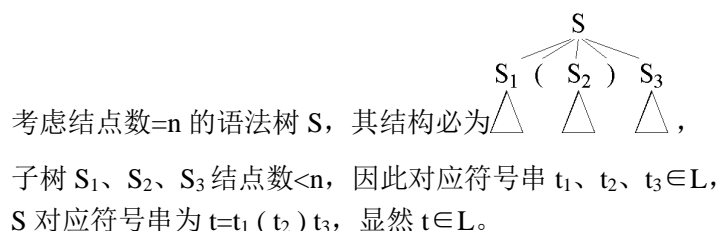
=====

c) $S \rightarrow S(S)S \mid \epsilon$

解：生成语言 $L=\{\text{括号匹配的符号串, 包括}\varepsilon\}$

证明：1) 证文法推导出的符号串都在 L 中

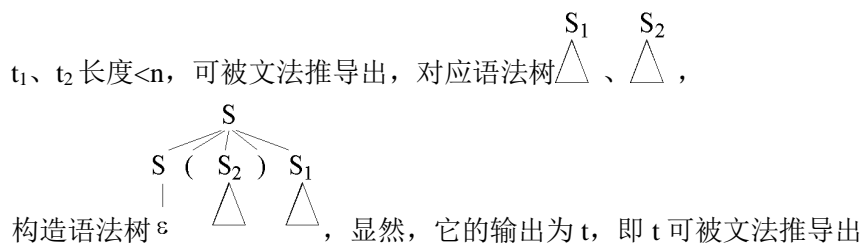
- i) 考虑最小语法树 ε ，推导出的符号串 ε 显然 $\in L$
- ii) 假定结点数 $<n$ 的语法树对应的符号串都 $\in L$ ，



综合 i)、ii)，1)得证

2) 证 L 中符号串都可由文法推导出

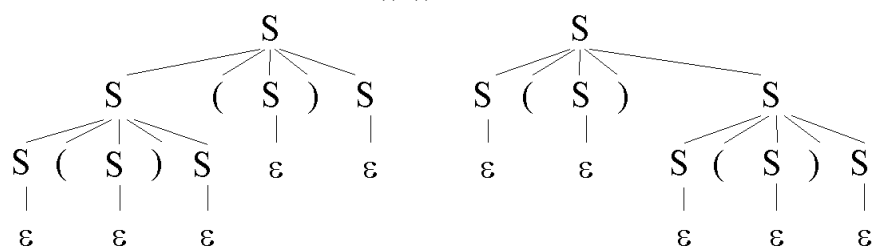
- i) L 中最短符号串 ε ，显然可由文法推导出
- ii) 假定 L 中长度 $<n$ 的符号串都可由文法推导出，
- 考虑长度 $=n$ 的符号串 $t=(\dots)$ ，寻找它的 $\in L$ 的最短前缀 t' ，则 $t=t't_1$ ， $t_1 \in L$ （可能为 ε ），
- t' 的形式必为 $()$ 或 $((\dots))$ （否则不是最短前缀），将它表示为 $t'=(t_2)$ ，无论哪种情况，必有 $t_2 \in L$ ，



综合 i)、ii)，2)得证

综合 1)、2)，文法生成的语言即为 L

另外，文法有二义性，符号串 $()()$ 可对应两个语法树



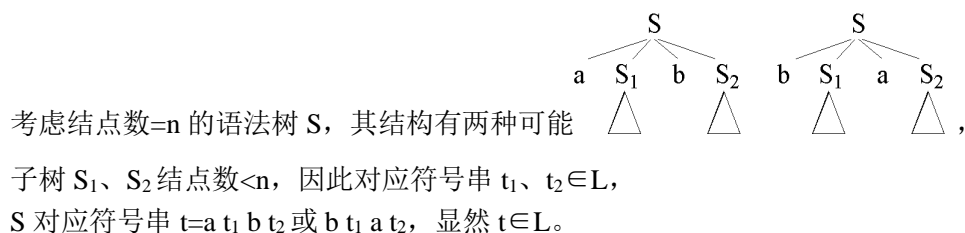
=====

d) $S \rightarrow a S b S \mid b S a S \mid \varepsilon$

解：生成语言 $L=\{\text{相同个数的 a、b 组成的所有符号串, 包括}\varepsilon\}$

证明：1) 证文法推导出的符号串都在 L 中

- i) 考虑最小语法树 ϵ ，推导出的符号串 ϵ 显然 $\in L$
- ii) 假定结点数 $< n$ 的语法树对应的符号串都 $\in L$ ，



综合 i)、ii)，1)得证

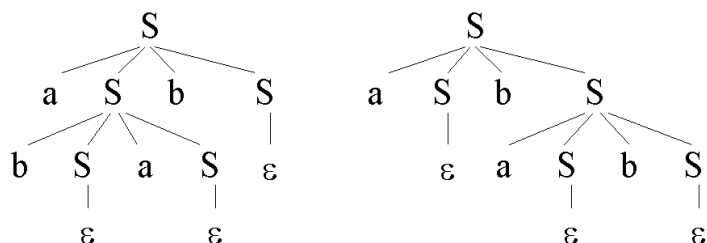
2) 证 L 中符号串都可由文法推导出

- i) L 中最短符号串 ϵ ，显然可由文法推导出
- ii) 假定 L 中长度 $< n$ 的符号串都可由文法推导出，
- 考虑长度 $= n$ 的符号串 t ，考虑以 a 开头的情况 $t = a \dots$ （以 b 开头的情况类似），
- 寻找它的 $\in L$ 的最短前缀 t' ， t' 必有形式 $a \dots b$ （否则不是最短前缀），
- 将它表示为 $t' = a t_1 b$ ，显然 $t_1 \in L$ （可能为 ϵ ），
- 而 $t = t' t_2$ ， $t_2 \in L$ ，则 $t = a t_1 b t_2$ 。
- 由 i)图，可为 t_1 、 t_2 构造语法树，则可为 t 构造语法树，即 t 可被文法推导出

综合 i)、ii)，2)得证

综合 1)、2)，文法生成的语言即为 L

另外，文法有二义性，符号串 $abab$ 可对应两个语法树



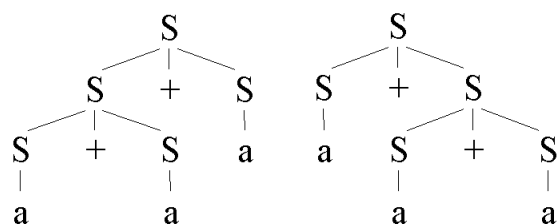
=====

e) $S \rightarrow a \mid S + S \mid S S \mid S^* \mid (S)$

解：生成语言 $L = \{\text{支持加法等四种运算的表达式}\}$

证明：类似 b)

文法是二义性文法，符号串 $a + a + a$ 可构造两个语法树



=====

2.4 为下列语言构造非二义性的上下文无关文法（任选三个）

a) 前缀表示的算术表达式

解: $E \rightarrow +EE \mid -EE \mid *EE \mid /EE \mid \text{num}$

证明参照 2.3

b) 以','间隔的左结合的标识符列表

解: $\text{list} \rightarrow \text{list}, \text{id} \mid \text{id}$

c) 以','间隔的右结合的标识符列表

解: $\text{list} \rightarrow \text{id}, \text{list} \mid \text{id}$

d) 包含整数和标识符, 支持+、-、*、/的算术表达式

解:

$\text{expr} \rightarrow \text{expr} + \text{term} \mid \text{expr} - \text{term} \mid \text{term}$

$\text{term} \rightarrow \text{term} * \text{factor} \mid \text{term} / \text{factor} \mid \text{factor}$

$\text{factor} \rightarrow \text{id} \mid \text{num} \mid (\text{expr})$

e) 在 d)的基础上, 添加一元+、-运算符

解:

$\text{expr} \rightarrow \text{expr} + \text{term} \mid \text{expr} - \text{term} \mid \text{term}$

$\text{term} \rightarrow \text{term} * \text{unary} \mid \text{term} / \text{unary} \mid \text{unary}$

$\text{unary} \rightarrow + \text{factor} \mid - \text{factor}$

$\text{factor} \rightarrow \text{id} \mid \text{num} \mid (\text{expr})$

=====

2.5

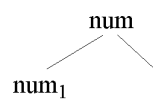
a) 证明文法

$\text{num} \rightarrow 11 \mid 1001 \mid \text{num} 0 \mid \text{num num}$

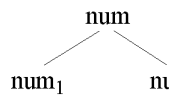
生成的二进制串所表示的数值均可被 3 整除

证明: 1) 考虑规模最小的语法树, 生成二进制串为 11、110、1001、1100、1111, 表示数 3、6、9、12、15, 均可被 3 整除

2) 假定结点数 $<n$ 的语法树生成的二进制串均可被 3 整除, 考虑结点数 $=n$ 的语法树其结构有两种可能

 , 子树 num_1 结点数 $<n$, 生成的二进制串 y 可被 3 整除, 而 num 生成

的二进制串 $x=y*2$ ，因此也可被 3 整除



，子树 num_1 、 num_2 结点数 $< n$ ，生成的二进制串 y 、 z 可被 3 整除，而 num 生成的二进制串 $x=y*2^n+z$ ，因此也可被 3 整除
综合 1)、2)，命题得证

b) 文法是否生成所有可被 3 整除的二进制串？

解：不是。二进制串 10101，数值为 21，可被 3 整除，但无法由文法推导出

=====

2.6 为罗马数字构造一个上下文无关文法

$RomanNumeral \rightarrow Thousand\ Hundreds\ Tens\ Ones$

$Ones \rightarrow LowOnes \mid IV \mid V\ LowOnes \mid IX$

$LowOnes \rightarrow \varepsilon \mid I \mid II \mid III$

$Tens \rightarrow LowTens \mid XL \mid L\ LowTens \mid XC$

$LowTens \rightarrow \varepsilon \mid X \mid XX \mid XXX$

$Hundreds \rightarrow LowHundreds \mid CD \mid D\ LowHundreds \mid CM$

$LowHundreds \rightarrow \varepsilon \mid C \mid CC \mid CCC$

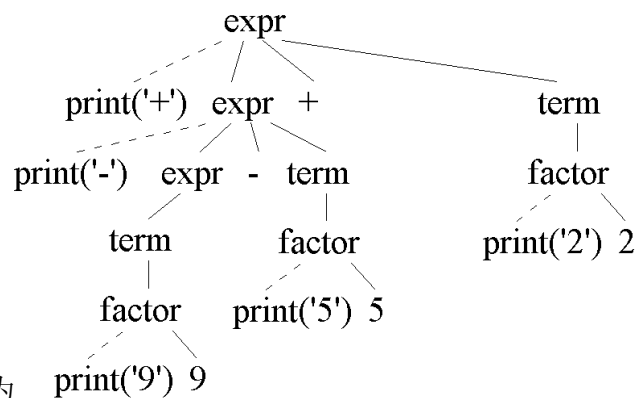
$Thousands \rightarrow M\ Thousands \mid \varepsilon$

=====

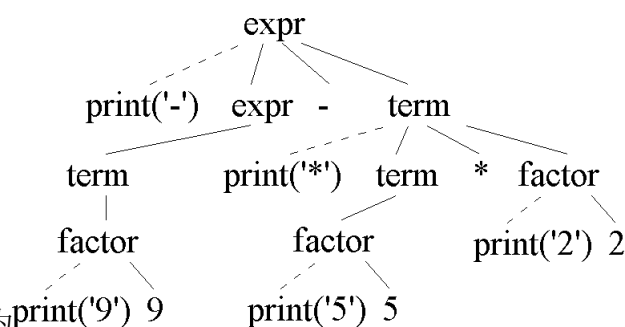
2.7 构造翻译模式，实现表达式中缀表示形式到前缀表示形式的转换。构造 $9-5+2$ 和 $9-5*2$ 的注释语法分析树

解：

```
expr → { print('+'); } expr + term
      | { print('-'); } expr - term
      | term
term  → { print('*'); } term * factor
      | { print('/'); } term / factor
      | factor
factor → { print(num.value); } num
        | ( expr )
```



9-5+2 的语法树为 9 5 2 , 输出+ - 9 5 2



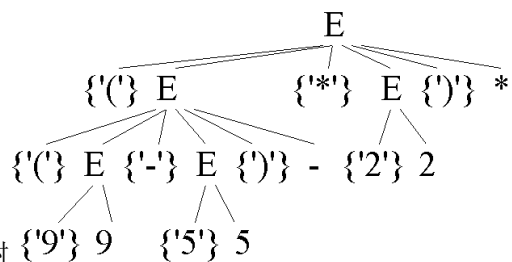
9-5*2 的语法树为 9 5 2 , 输出- 9 * 5 2

=====

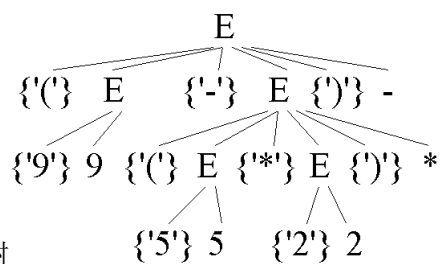
2.8 构造翻译模式，实现后缀表达式到中缀表达式的翻译，构造 95-2*和 952*-的注释语法树解：

```

E → { print('('); } E { print('+'); } E { print(')'}; } +
    | { print('('); } E { print('-'); } E { print(')'}; } -
    | { print('('); } E { print('*'); } E { print(')'}; } *
    | { print('('); } E { print('/'); } E { print(')'}; } /
    | id { print(id.lexeme); }
    | num { print(num.value); }
  
```



95-2*的语法树 9 5 2 , 输出为((9-5)*2)



952*-的语法树 9 5 2 , 输出为(9-(5*2))

显然，此解法括号有冗余情况，但结果是正确的

=====

2.9 构建一个语法制导翻译模式，将整数翻译成罗马数字

解：NT Ones、Tens、Hundreds 表示整数的个、十、百位，Thousands 表示千以上的所有位，Num 表示完整的整数，每个 NT 设定一个属性 *roman*，为字符串类型，代表罗马数字的表示形式。假定存在 *chrrep(char c, int n)* 函数，它将字符 *c* 重复 *n* 次，形成一个字符串，作为返回结果，若 *n=0*，则返回一个空串。

Num → *Thousands Hundreds Tens Ones*

{ *Num.roman* = *Thousands.roman* || *Hundreds.roman* || *Tens.roman* || *Ones.roman*; }

Thousands → *digits* { *Thousands.roman* = *chrrep*(‘M’, *digits.val*); }

digits → *digits digit* { *digits.val* = *digits.val* * 10 + *digit.val*; }

| ε { *digits.val* = 0; }

digit → *small* { *digit.val* = *small.val*; }

| *big* { *digit.val* = *big.val*; }

| 4 { *digit.val* = 4; }

| 9 { *digit.val* = 9; }

small → 0 { *small.val* = 0; }

| 1 { *small.val* = 1; }

| 2 { *small.val* = 2; }

| 3 { *small.val* = 3; }

big → 5 { *small.val* = 5; }

| 6 { *small.val* = 6; }

| 7 { *small.val* = 7; }

| 8 { *small.val* = 8; }

Hundreds → *small* { *Hundreds.roman* = *chrrep*(‘C’, *small.val*); }

| *big* { *Hundreds.roman* = “D” || *chrrep*(‘C’, *small.val*); }

| 4 { *Hundreds.roman* = “CD”; }

| 9 { *Hundreds.roman* = “CM”; }

Tens → *small* { *Tens.roman* = *chrrep*(‘X’, *small.val*); }

| *big* { *Tens.roman* = “L” || *chrrep*(‘X’, *small.val*); }

| 4 { *Tens.roman* = “XL”; }

| 9 { *Tens.roman* = “XC”; }

Ones → *small* { *Ones.roman* = *chrrep*(‘I’, *small.val*); }

| *big* { *Ones.roman* = “V” || *chrrep*(‘I’, *small.val*); }

| 4 { *Ones.roman* = “IV”; }

| 9 { *Ones.roman* = “IX”; }

=====

2.10 构建一个语法制导翻译模式，将罗马数字翻译成整数

解：每个 NT 设定一个属性 *val*，表示对应整数值

RomanNumeral \rightarrow *Thousands Hundreds Tens Ones*

```

{ RomanNumeral.val = Thousands.val + Hundreds.val
  + Tens.val + Ones.val;
  printf("%d\n", RomanNumeral.val;) }

Ones  $\rightarrow$  LowOnes      { Ones.val = LowOnes.val; }
  | IV      { Ones.val = 4; }
  | V LowOnes { Ones.val = LowOnes.val + 5; }
  | IX      { Ones.val = 9; }

LowOnes  $\rightarrow$   $\epsilon$  { LowOnes.val = 0; }
  | I  { LowOnes.val = 1; }
  | II { LowOnes.val = 2; }
  | III { LowOnes.val = 3; }

Tens  $\rightarrow$  LowTens      { Tens.val = LowTens.val; }
  | XL      { Tens.val = 40; }
  | L LowTens { Tens.val = LowTens.val + 50; }
  | XC      { Tens.val = 90; }

LowTens  $\rightarrow$   $\epsilon$       { LowTens.val = 0; }
  | X      { LowTens.val = 10; }
  | XX     { LowTens.val = 20; }
  | XXX    { LowTens.val = 30; }

Hundreds  $\rightarrow$  LowHundreds { Hundreds.val = LowHundreds.val; }
  | CD      { Hundreds.val = 400; }
  | D LowHundreds { Hundreds.val = LowHundreds.val + 500; }
  | CM      { Hundreds.val = 900; }

LowHundreds  $\rightarrow$   $\epsilon$  { LowHundreds.val = 0; }
  | C      { LowHundreds.val = 100; }
  | CC     { LowHundreds.val = 200; }
  | CCC    { LowHundreds.val = 300; }

Thousands  $\rightarrow$  M Thousands1 { Thousands.val = Thousands1.val + 1000; }
  |  $\epsilon$  { Thousands.val = 0; }

```

=====

2.14

解：for 语句对应堆栈机代码段如下

expr ₁ 的代码段
label test
expr ₂ 的代码段
gofalse out
stmt 的代码段
expr ₃ 的代码段
goto test
label out

$stmt \rightarrow \mathbf{for} \ (\ expr_1 \ ; \ expr_2 \ ; \ expr_3 \) \ stmt_1 \{$	<pre>test = newlabel(); out = newlabel(); stmt.t = expr1.t 'label' test expr2.t 'gofalse' out stmt1.t expr3.t 'goto' test 'label' out; }</pre>
--	--

第三章习题解答

Aho:《编译原理》书中习题

陈:《程序设计语言编译原理》书中习题

(Aho)3.6 描述下列正规式定义的语言

a) $0(0|1)^*0$

解: 定义的语言 $L=\{\text{以 } 0 \text{ 开头、以 } 0 \text{ 结尾的 } 0、1 \text{ 串}\}$

b) $((\epsilon|0)1^*)^*$

解: 定义的语言 $L=\{\text{所有 } 0、1 \text{ 串}\}$

c) $(0|1)^*0(0|1)(0|1)$

解: 定义的语言 $L=\{\text{第 } 3 \text{ 位为 } 0 \text{ 的二进制串}\}$

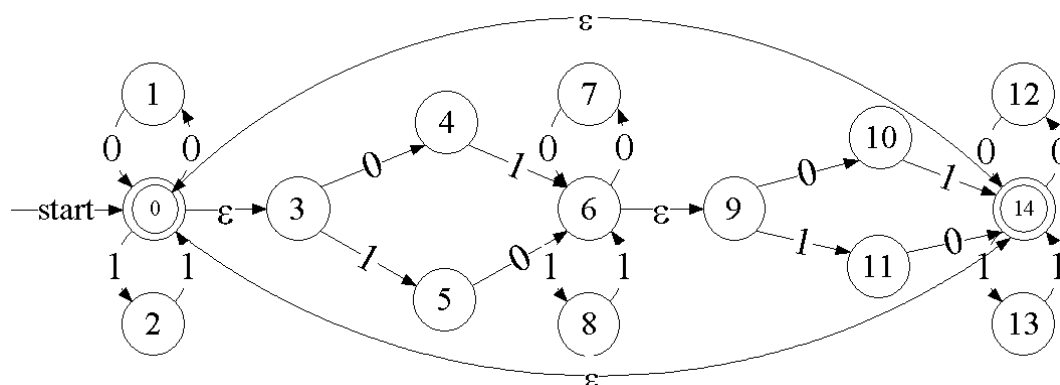
d) $0^*10^*10^*10^*$

解: 定义的语言 $L=\{\text{包含 } 3 \text{ 个 } 1 \text{ 的 } 0、1 \text{ 串}\}$

e) $(00 \mid 11)^*((01 \mid 10)(00 \mid 11)^*(01 \mid 10)(00 \mid 11)^*)^*$

解：定义的语言 $L = \{ \text{包含偶数个 0、偶数个 1 的 0、1 串} \}$

首先画出正规式对应的 NFA：



将它转换为 DFA

$\epsilon\text{-closure}(\{0\}) = \{0, 3, 14\} = A$

$\epsilon\text{-closure}(\delta(\{0, 3, 14\}, 0)) = \{1, 4, 12\} = B$ $\epsilon\text{-closure}(\delta(\{0, 3, 14\}, 1)) = \{2, 5, 13\} = C$

$\epsilon\text{-closure}(\delta(\{1, 4, 12\}, 0)) = A$

$\epsilon\text{-closure}(\delta(\{1, 4, 12\}, 1)) = \{6, 9\} = D$

$\epsilon\text{-closure}(\delta(\{2, 5, 13\}, 0)) = D$

$\epsilon\text{-closure}(\delta(\{2, 5, 13\}, 1)) = A$

$\epsilon\text{-closure}(\delta(\{6, 9\}, 0)) = \{7, 10\} = E$

$\epsilon\text{-closure}(\delta(\{6, 9\}, 1)) = \{8, 11\} = F$

$\epsilon\text{-closure}(\delta(\{7, 10\}, 0)) = D$

$\epsilon\text{-closure}(\delta(\{7, 10\}, 1)) = \{3, 14\} = G$

$\epsilon\text{-closure}(\delta(\{8, 11\}, 0)) = G$

$\epsilon\text{-closure}(\delta(\{8, 11\}, 1)) = D$

$\epsilon\text{-closure}(\delta(\{3, 14\}, 0)) = \{4, 12\} = H$

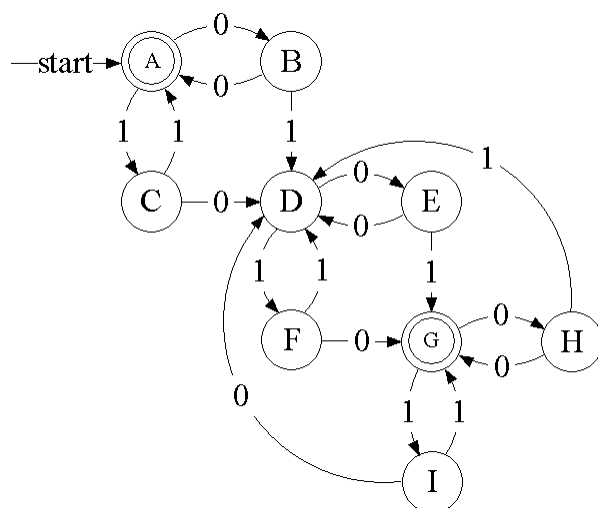
$\epsilon\text{-closure}(\delta(\{3, 14\}, 1)) = \{5, 13\} = I$

$\epsilon\text{-closure}(\delta(\{4, 12\}, 0)) = G$

$\epsilon\text{-closure}(\delta(\{4, 12\}, 1)) = D$

$\epsilon\text{-closure}(\delta(\{5, 13\}, 0)) = D$

$\epsilon\text{-closure}(\delta(\{5, 13\}, 1)) = G$

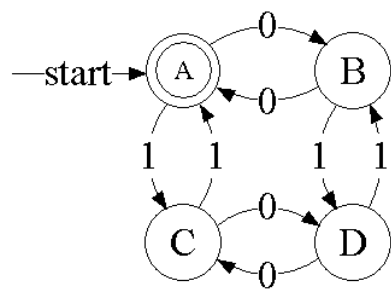


对此 DFA 进行化简：

1. $\{B, C, D, E, F, H, I\}, \{A, G\}$

2. $\{B, C, D, E, F, H, I\} \xrightarrow{0} \{B, F, H\}, \{C, D, E, I\} \quad \{A, G\}$

3. $\{C, D, E, I\} \xrightarrow{1} \{C, E, I\}, \{D\} \quad \{B, F, H\}, \{A, G\}$



A: 偶数个 0, 偶数个 1 的 0、1 串

B: 奇数个 0, 偶数个 1 的 0、1 串

C: 偶数个 0, 奇数个 1 的 0、1 串

D: 奇数个 0, 奇数个 1 的 0、1 串

(Aho)3.7 为下列语言写出正规定义（或正规式、正规文法、有限自动机）

a) 包含五个元音，且按顺序排列的所有字母串

解：con \rightarrow [b-df-hj-np-tv-z]

string \rightarrow (con)*a(con | a)*(con)*e(con | e)*(con)*i(con | i)*(con)*o(con | o)*(con)*u(con | u)*

b) 字母按字典升序排列的所有字母串

解：a*b*c*d*e*f*g*h*i*j*k*l*m*n*o*p*q*r*s*t*u*v*w*x*y*z*

c) 以/*开始，*/结束的注释，中间不能包含*/，除非包含在"和"中

解：/*([\wedge "])* | "." | *+[\wedge])****/

其中，/*和*/间出现内容，是三种不同情况的闭包（随意组合出的任意长度的串）

[\wedge "]*：除了*和"之外所有的符号任意长度的串

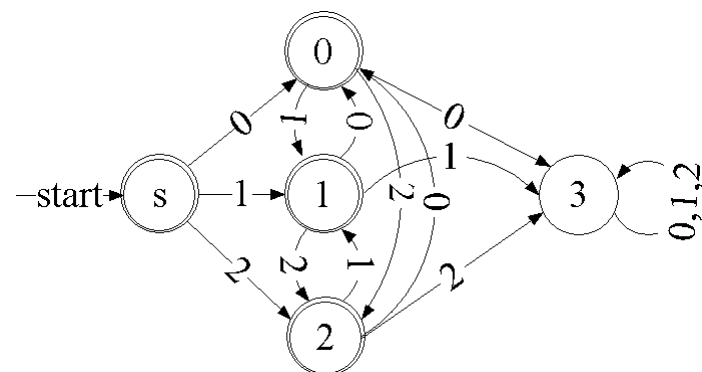
"."：两个引号括起来的串，其内允许任何符号

*+[\wedge]：出现*的情况，其后跟一个不是/的符号，这主要是避免*后接[\wedge "]*，产生*/的情况，至于后面跟随多个不是/的符号，第一个之后的内容可与[\wedge "]*匹配

但还遗漏了一种情况，即*后不跟随任何符号的情况，即立即接注释结尾的*/的情况，这只需在*/之前加一个**即可，有这种情况，与之匹配；没有—— ϵ ，也匹配

d) 不包含重复数字的数字串

解：0—9十个数字的情况比较复杂，考虑只有0—2三个数字的情况，对应 DFA 为



状态含义：

s: ϵ

0: 最后一个符号为0的数字串

1: 最后一个符号为1的数字串

2: 最后一个符号为2的数字串

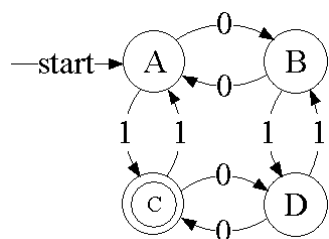
3: 死状态

e) 包含至多一个重复数字的数字串

解：同样可给出 DFA，按 d)类似方法设计状态即可

f) 包含偶数个 0 和奇数个 1 的 0、1 串

解：DFA 为



状态含义与上题 3.6 e) 相同。

正规式为 AB^* ，其中

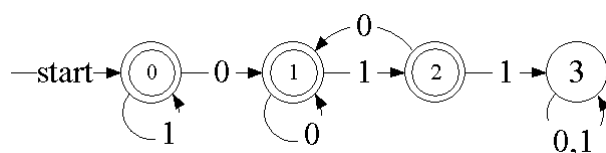
$A = 1 \mid 0(00 \mid 11)^*(10 \mid 01)$

$B = 00 \mid 11 \mid (01 \mid 10)(00 \mid 11)^*(01 \mid 10)$

DFA → 正规式的转换方法见参考资料 “NFA to RegEx Conversion.htm”

h) 不包含子串 011 的 0、1 串

解：DFA 为



状态 0：字符串未包含 0

状态 1：字符串包含子串 0

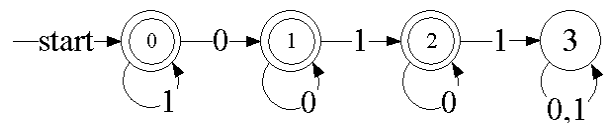
状态 2：字符串包含子串 01

状态 3：字符串包含子串 011

正规式为： $1^*0^* \mid 1^*0(0 \mid 10)^* \mid 1^*00^*1(00^*1)^* = 1^*(0 \mid 01)^*$

i) 不包含子序列 011 的 0、1 串

解：DFA 为



状态 0：字符串未包含 0

状态 1：字符串包含子序列 0

状态 2：字符串包含子序列 01

状态 3：字符串包含子序列 011

正规式为： $1^* \mid 1^*00^* \mid 1^*00^*10^*$

(Aho)3.14 用正规式表示 UNIX 的文件名表达式

解：只需将文件名表达式允许的所有操作符用正规式表示出来即可

's'——“s”

\c——\c

——.

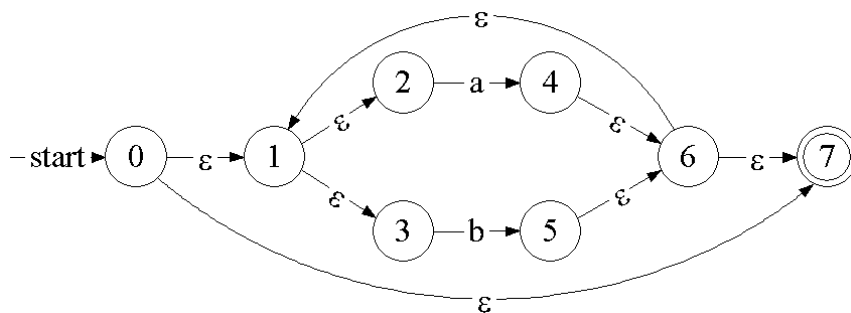
?——.

[s]——[s]

(Aho)3.16 使用 Thompson 构造法为下列正规式构造 NFA，写出每个 NFA 处理符号串 ababbab 过程中的状态转换序列

a) $(a | b)^*$

解：NFA 为

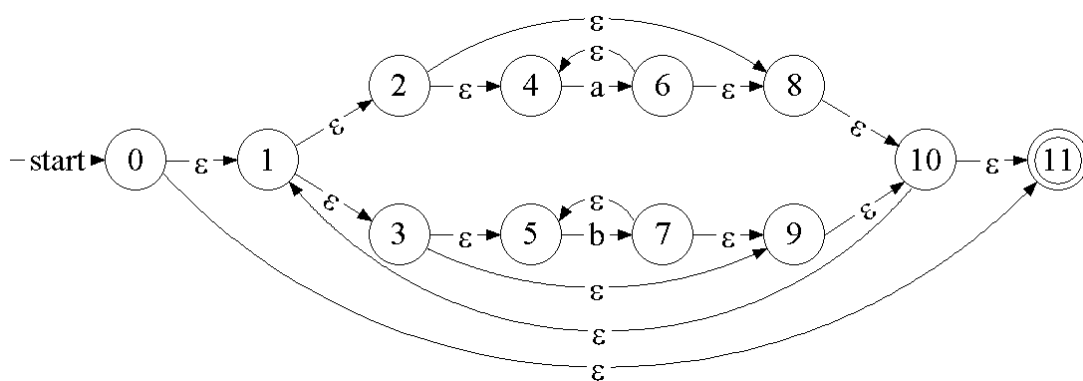


ababbab 状态转换序列：

0→1→2→4→6→1→3→5→6→1→2→4→6→1→3→5→6→1→3→5→6→1→2→4→6→1→3→5→6→7

b) $(a^* | b^*)^*$

解：NFA 为

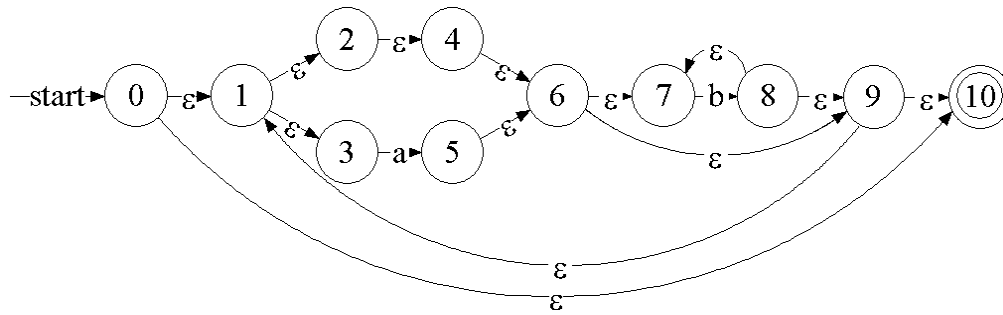


ababbab 状态转换序列：

0→1→2→4→6→8→10→1→3→5→7→9→10→1→2→4→6→8→10→1→3→5→7→5→7→9→10→11

c) $((\epsilon | a)b^*)^*$

解：NFA 为

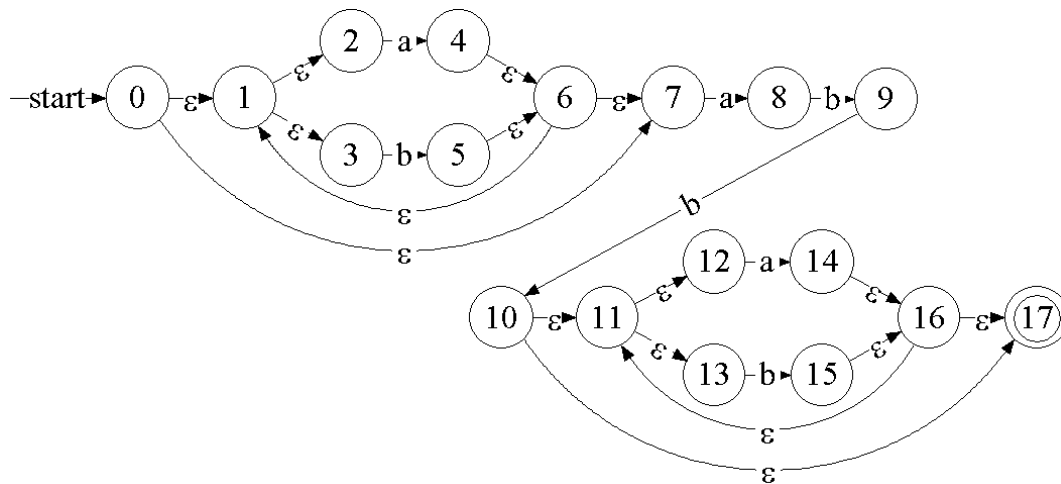


ababbab 状态转换序列:

0→1→3→5→6→7→8→9→1→3→5→6→7→8→7→8→9→1→3→5→6→7→8→9→10

d) $(a | b)^*abb(a | b)^*$

解：NFA 为



ababbab 状态转换序列:

0→1→2→4→6→1→3→5→6→7→8→9→10→11→12→14→16→11→13→15→16→17

(Aho)3.17 利用子集构造法将 3.16 题得到的 NFA 转换为 DFA，同样写出分析符号串 ababbab 过程中的状态转换。

a) 解：

$$\varepsilon\text{-closure}(\{0\}) = \{0, 1, 2, 3, 7\} = A$$

$$\varepsilon\text{-closure}(\delta(A, a)) = \{1, 2, 3, 4, 6, 7\} = B$$

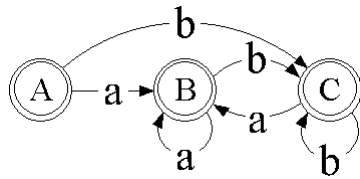
$$\varepsilon\text{-closure}(\delta(A, b)) = \{1, 2, 3, 5, 6, 7\} = C$$

$$\varepsilon\text{-closure}(\delta(B, a)) = B$$

$$\varepsilon\text{-closure}(\delta(B, b)) = C$$

$$\varepsilon\text{-closure}(\delta(C, a)) = B$$

$$\varepsilon\text{-closure}(\delta(C, b)) = C$$



ababbab 状态转换序列：

$A \rightarrow B \rightarrow C \rightarrow B \rightarrow C \rightarrow C \rightarrow B \rightarrow C$

b) 解：

$$\varepsilon\text{-closure}(\{0\}) = \{0, 1, 2, 3, 4, 5, 8, 9, 10, 11\} = A$$

$$\varepsilon\text{-closure}(\delta(A, a)) = \{1, 2, 3, 4, 5, 6, 8, 9, 10, 11\} = B$$

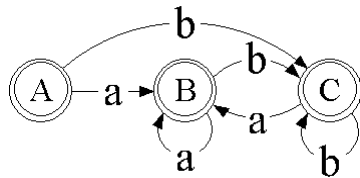
$$\varepsilon\text{-closure}(\delta(A, b)) = \{1, 2, 3, 4, 5, 7, 8, 9, 10, 11\} = C$$

$$\varepsilon\text{-closure}(\delta(B, a)) = B$$

$$\varepsilon\text{-closure}(\delta(B, b)) = C$$

$$\varepsilon\text{-closure}(\delta(C, a)) = B$$

$$\varepsilon\text{-closure}(\delta(C, b)) = C$$



ababbab 状态转换序列：

$A \rightarrow B \rightarrow C \rightarrow B \rightarrow C \rightarrow C \rightarrow B \rightarrow C$

c) 解：

$$\varepsilon\text{-closure}(\{0\}) = \{0, 1, 2, 3, 4, 6, 7, 9, 10\} = A$$

$$\varepsilon\text{-closure}(\delta(A, a)) = \{1, 2, 3, 4, 5, 6, 7, 9, 10\} = B$$

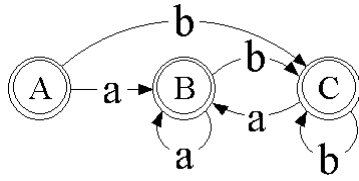
$$\varepsilon\text{-closure}(\delta(A, b)) = \{1, 2, 3, 4, 6, 7, 8, 9, 10\} = C$$

$$\varepsilon\text{-closure}(\delta(B, a)) = B$$

$$\varepsilon\text{-closure}(\delta(B, b)) = C$$

$$\varepsilon\text{-closure}(\delta(C, a)) = B$$

$$\varepsilon\text{-closure}(\delta(C, b)) = C$$



ababbab 状态转换序列:

$A \rightarrow B \rightarrow C \rightarrow B \rightarrow C \rightarrow C \rightarrow B \rightarrow C$

d) 解:

$\varepsilon\text{-closure}(\{0\}) = \{0, 1, 2, 3, 7\} = A$

$\varepsilon\text{-closure}(\delta(A, a)) = \{1, 2, 3, 4, 6, 7, 8\} = B$

$\varepsilon\text{-closure}(\delta(A, b)) = \{1, 2, 3, 5, 6, 7\} = C$

$\varepsilon\text{-closure}(\delta(B, a)) = B$

$\varepsilon\text{-closure}(\delta(B, b)) = \{1, 2, 3, 5, 6, 7, 9\} = D$

$\varepsilon\text{-closure}(\delta(C, a)) = B$

$\varepsilon\text{-closure}(\delta(C, b)) = C$

$\varepsilon\text{-closure}(\delta(D, a)) = B$

$\varepsilon\text{-closure}(\delta(D, b)) = \{1, 2, 3, 5, 6, 7, 10, 11, 12, 13, 17\} = E$

$\varepsilon\text{-closure}(\delta(E, a)) = \{1, 2, 3, 4, 6, 7, 8, 11, 12, 13, 14, 16, 17\} = F$

$\varepsilon\text{-closure}(\delta(E, b)) = \{1, 2, 3, 5, 6, 7, 11, 12, 13, 15, 16, 17\} = G$

$\varepsilon\text{-closure}(\delta(F, a)) = F$

$\varepsilon\text{-closure}(\delta(F, b)) = \{1, 2, 3, 5, 6, 7, 9, 11, 12, 13, 15, 16, 17\} = H$

$\varepsilon\text{-closure}(\delta(G, a)) = F$

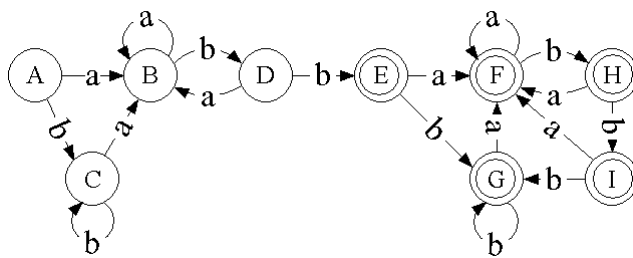
$\varepsilon\text{-closure}(\delta(G, b)) = G$

$\varepsilon\text{-closure}(\delta(H, a)) = F$

$\varepsilon\text{-closure}(\delta(H, b)) = \{1, 2, 3, 5, 6, 7, 10, 11, 12, 13, 15, 16, 17\} = I$

$\varepsilon\text{-closure}(\delta(I, a)) = F$

$\varepsilon\text{-closure}(\delta(I, b)) = G$

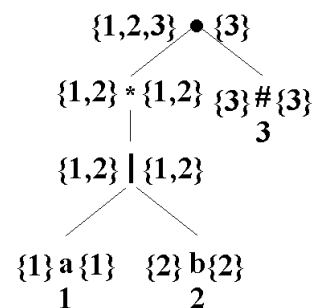


ababbab 状态转换序列:

$A \rightarrow B \rightarrow D \rightarrow B \rightarrow D \rightarrow E \rightarrow F \rightarrow H$

(Aho)3.18 为 3.16 的正规式直接构造 DFA，比较得到结果与 3.17 结果的状态数。

a) 解：

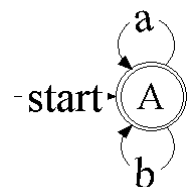


	followpos
1	{1, 2, 3}
2	{1, 2, 3}
3	

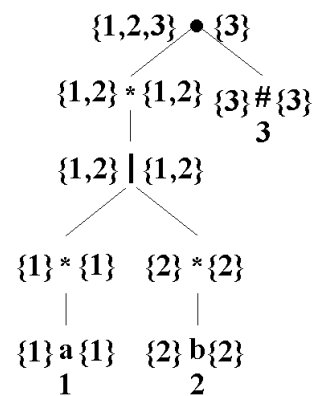
firstpos(root) = {1, 2, 3} = A

$\delta(A, a) = \text{followpos}(1) = \{1, 2, 3\} = A$

$\delta(A, b) = \text{followpos}(1) = \{1, 2, 3\} = A$



b) 解：

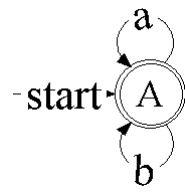


	followpos
1	{1, 2, 3}
2	{1, 2, 3}
3	

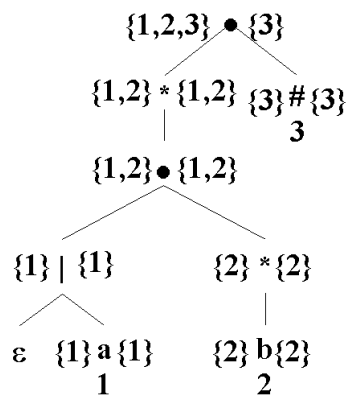
firstpos(root) = {1, 2, 3} = A

$\delta(A, a) = \text{followpos}(1) = \{1, 2, 3\} = A$

$\delta(A, b) = \text{followpos}(1) = \{1, 2, 3\} = A$



c) 解:

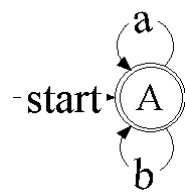


	followpos
1	{ 1, 2, 3 }
2	{ 1, 2, 3 }
3	

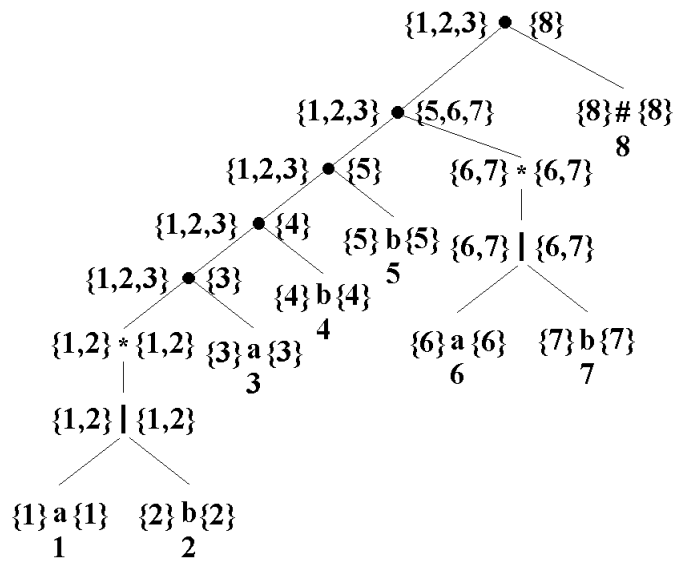
$\text{firstpos}(\text{root}) = \{ 1, 2, 3 \} = A$

$\delta(A, a) = \text{followpos}(1) = \{ 1, 2, 3 \} = A$

$\delta(A, b) = \text{followpos}(1) = \{ 1, 2, 3 \} = A$



d) 解:



	followpos
1	{1, 2, 3}
2	{1, 2, 3}
3	{4}
4	{5}
5	{6, 7, 8}
6	{6, 7, 8}
7	{6, 7, 8}
8	

$\text{firstpos}(\text{root}) = \{1, 2, 3\} = A$

$\delta(A, a) = \text{followpos}(1) \cup \text{followpos}(3) = \{1, 2, 3, 4\} = B$

$\delta(A, b) = \text{followpos}(2) = A$

$\delta(B, a) = \text{followpos}(1) \cup \text{followpos}(3) = B$

$\delta(B, b) = \text{followpos}(2) \cup \text{followpos}(4) = \{1, 2, 3, 5\} = C$

$\delta(C, a) = \text{followpos}(1) \cup \text{followpos}(3) = B$

$\delta(C, b) = \text{followpos}(2) \cup \text{followpos}(5) = \{1, 2, 3, 6, 7, 8\} = D$

$\delta(D, a) = \text{followpos}(1) \cup \text{followpos}(3) \cup \text{followpos}(6) = \{1, 2, 3, 4, 6, 7, 8\} = E$

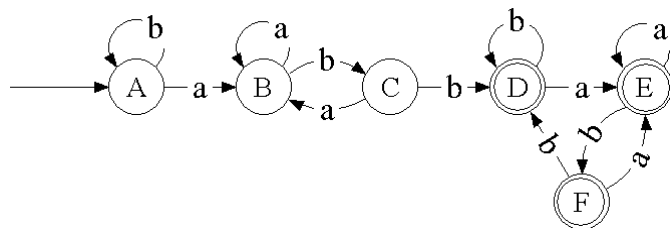
$\delta(D, b) = \text{followpos}(2) \cup \text{followpos}(7) = D$

$\delta(E, a) = \text{followpos}(1) \cup \text{followpos}(3) \cup \text{followpos}(6) = E$

$\delta(E, b) = \text{followpos}(2) \cup \text{followpos}(4) \cup \text{followpos}(7) = \{1, 2, 3, 5, 6, 7, 8\} = F$

$\delta(F, a) = \text{followpos}(1) \cup \text{followpos}(3) \cup \text{followpos}(6) = E$

$\delta(A, b) = \text{followpos}(2) \cup \text{followpos}(7) = D$



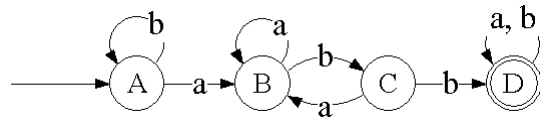
(Aho)3.21 最小化 3.18 得到的 DFA

d) 解:

初始划分: $\{A, B, C\}, \{D, E, F\}$

$\{A, B, C\} \xrightarrow{b} \{A, B\}, \{C\}$

$\{A, B\} \xrightarrow{b} \{A\}, \{B\}$



3.22 正规式的等价可以通过对应的最小化的 DFA 的结构等价性来证明, 利用这种方法, 证明下列正规式的等价性

a) $(a \mid b)^*$

b) $(a^* \mid b^*)^*$

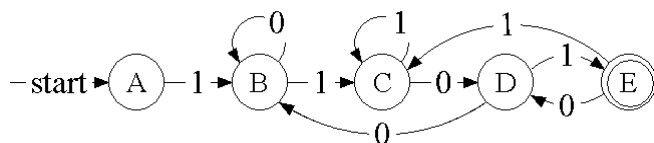
c) $((\epsilon \mid a)b^*)^*$

由 16、17、18、21 结果可知

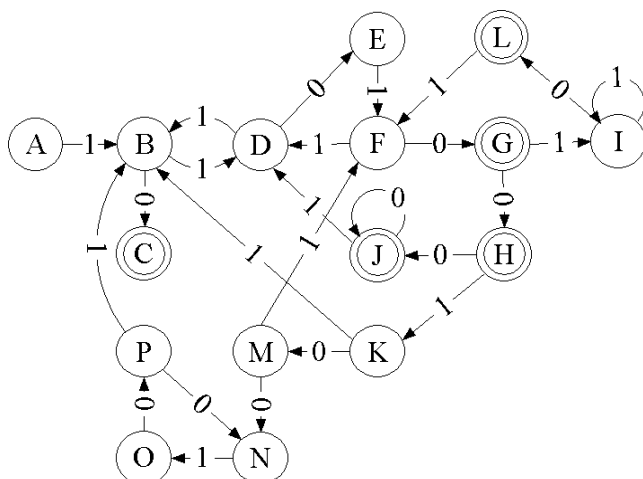
(陈) 7 构造下列正规式对应的 DFA

解法: 可正规式 \rightarrow NFA \rightarrow DFA, 或者正规式 \rightarrow DFA

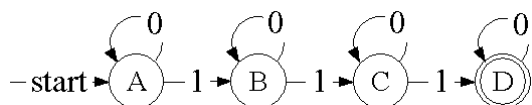
a) $1(0|1)^*101$



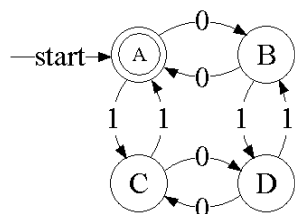
b) $1(1010^*|1(010)^*1)^*0$



c) $0^*10^*10^*10^*$



d) $(00|11)^*((01|10)(00|11)^*(01|10)(00|11)^*)^*$



(陈) 8 为下列语言写出正规定义 (或正规式、正规文法、有限自动机)

a) 以 01 结尾的 0、1 串

解: $(0|1)^*01$

b) 能被 5 整除的十进制整数

解: $[1-9][0-9]^*(0|5)|0|5$

c) 包含奇数个 1 和奇数个 0 的二进制串

解: $(00|11)^*(01|10)((01|10)^+(00|11)^*(01|10)^+)^*$

解法见 Aho3.7f

(陈) 10 一个人带一只狼、一只羊和一颗白菜过河，每次人只能将一样东西摆渡到对岸。而在无人的情况下，狼会吃掉羊，羊会吃掉白菜。目标，人将三样东西都摆渡到对岸，羊和白菜都不被吃掉。利用有限自动机得到渡河的方法

解：M——人，W——狼，S——羊，C——白菜，

状态——哪些东西在原岸，如：MWS，表示人、狼、羊在原岸（白菜在目的岸）

动作——人将一样东西摆渡到对岸（原→目的——表示为→，也可能是目的→原——表示为←）

不满足要求的状态为死状态，如：WS、SC、...

初态：MWSC，终态：ε，初态到终态的一条路径——渡河方案。

$\delta(\text{MWSC}, \rightarrow \text{MW}) = \text{SC}$ （红色表示死状态）

$\delta(\text{MWSC}, \rightarrow \text{MS}) = \text{WC}$

$\delta(\text{MWSC}, \rightarrow \text{MC}) = \text{WS}$

$\delta(\text{MWSC}, \rightarrow \text{M}) = \text{WSC}$

$\delta(\text{WC}, \leftarrow \text{MW}) = \text{MWSC}$ （蓝色表示出现过的状态）

$\delta(\text{WC}, \leftarrow \text{M}) = \text{MWC}$

$\delta(\text{WSC}, \leftarrow \text{M}) = \text{MWSC}$

$\delta(\text{MWC}, \rightarrow \text{MW}) = \text{C}$

$\delta(\text{MWC}, \rightarrow \text{MC}) = \text{W}$

$\delta(\text{MWC}, \rightarrow \text{M}) = \text{WC}$

$\delta(\text{C}, \leftarrow \text{MW}) = \text{MWC}$

$\delta(\text{C}, \leftarrow \text{MS}) = \text{MSC}$

$\delta(\text{C}, \leftarrow \text{M}) = \text{MC}$

$\delta(\text{W}, \leftarrow \text{MS}) = \text{MWS}$

$\delta(\text{W}, \leftarrow \text{MC}) = \text{MWC}$

$\delta(\text{W}, \leftarrow \text{M}) = \text{MW}$

$\delta(\text{MSC}, \rightarrow \text{MS}) = \text{C}$

$\delta(\text{MSC}, \rightarrow \text{MC}) = \text{S}$

$\delta(\text{MSC}, \rightarrow \text{M}) = \text{SC}$

$\delta(\text{MWS}, \rightarrow \text{MW}) = \text{S}$

$\delta(\text{MWS}, \rightarrow \text{MS}) = \text{W}$

$\delta(\text{MWS}, \rightarrow \text{M}) = \text{WS}$

$\delta(\text{S}, \leftarrow \text{MW}) = \text{MWS}$

$\delta(\text{S}, \leftarrow \text{MC}) = \text{MSC}$

$\delta(\text{S}, \leftarrow \text{M}) = \text{MS}$

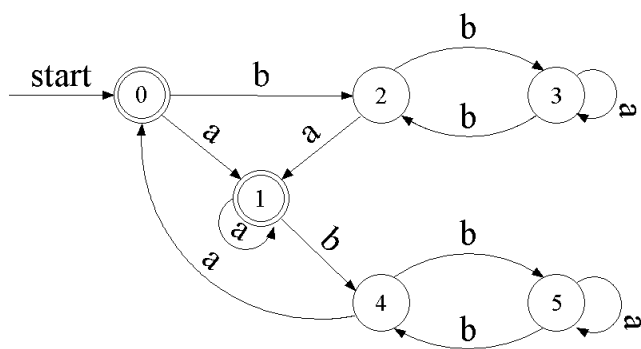
$\delta(\text{MS}, \rightarrow \text{MS}) = \varepsilon$ 终态！

$\delta(\text{MS}, \rightarrow \text{M}) = \text{S}$

状态转换序列：MWSC→WC→MWC→C→MSC→S→MS→ε

动作序列：→MS，←M，→MW，←MS，→MC，←M，→MS

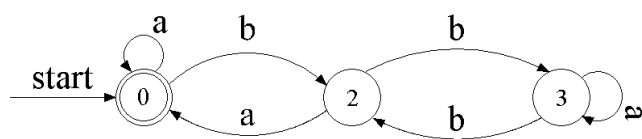
(陈) 12 将下面 DFA 最小化



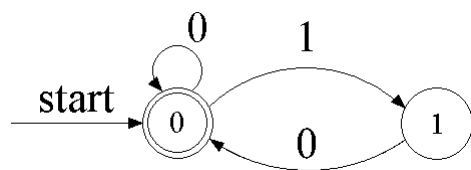
解:

初始划分: $\{0, 1\}, \{2, 3, 4, 5\}$

$\{2, 3, 4, 5\} \xrightarrow{a} \{2, 4\}, \{3, 5\}$



(陈) 14 构造 DFA, 它接受 $\Sigma=\{0, 1\}$ 上所有满足如下条件的符号串: 每个 1 都有 0 直接跟在右边。



(陈) 17 下面符号串集合是否为正规集? 若是, 写出正规式, 否则, 给出证明

a) $L_1 = \{a^n b^n \mid n \geq 0\}$

解: 不是正规集, 证明见讲义 4-1 第 51 页

b) $L_2 = \{x \mid x \text{ 中含有相同个数的 } a \text{ 和 } b\}$

解: 不是正规集, 证明类似 a)

c) $L_3 = \{a^p \mid p \text{ 为素数}\}$

解: 不是正规集

证明: 假设 L_3 是正规集, 对应 DFA D 的状态数为 k 。

令 p_0, p_1, \dots, p_k 为前 $k+1$ 个素数,

s_0, s_1, \dots, s_k 为 D 读入 $a^{p_0}, a^{p_1}, \dots, a^{p_k}$ 后到达的状态。

显然, 必有 $0 \leq i < j \leq k$, s_i, s_j 为同一状态, 且必然是终态, 用 s 表示它。

因此, $s \xrightarrow{a} s$ 存在一条长度为 $p_j - p_i$ 的路径, 每条边标记为 a 。

在读入 a^{p_i} 后, 重复此路径 $(p_j - p_i)$ 次, 得到的符号串 t 包含 a 的个数为

$$p_j + (p_j - p_i) * (p_i - 1) = p_j * p_i - p_i * (p_i - 1) = p_i * (p_j - p_i + 1)$$

而 $p_j - p_i + 1 > 1$, 因此 $p_i * (p_j - p_i + 1)$ 为合数。

D 接受 t , 但 t 不是 L_3 中符号串, 矛盾。

因此, L_3 不是正规集。

第四章部分习题解答

Aho: 《编译原理技术与工具》书中习题

(Aho) 4.1 考虑文法

$S \rightarrow (L) \mid a$

$L \rightarrow L, S \mid S$

a) 列出终结符、非终结符和开始符号

解:

终结符: (、)、a、,

非终结符: S、L

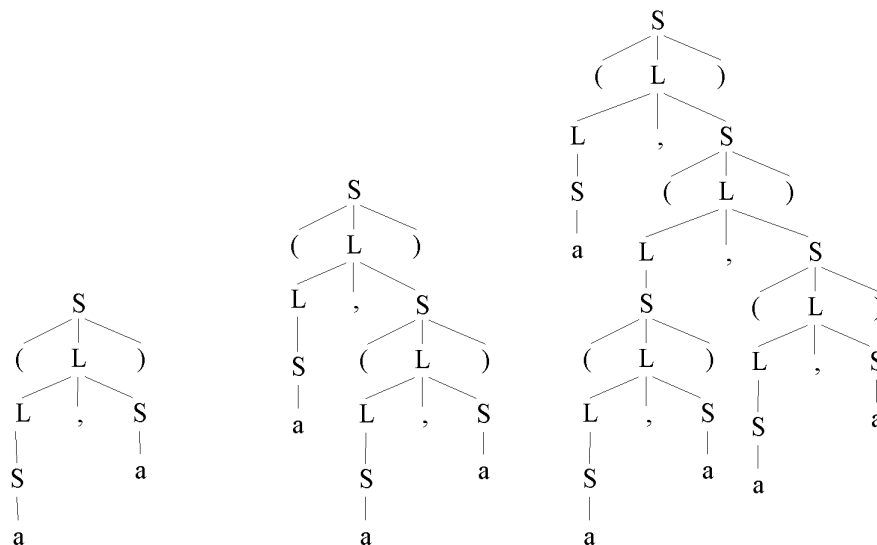
开始符号: S

b) 给出下列句子的语法树

i) (a, a)

ii) (a, (a, a))

iii) (a, ((a, a), (a, a)))



c) 构造 b)中句子的最左推导

i) $S \Rightarrow (L) \Rightarrow (L, S) \Rightarrow (S, S) \Rightarrow (a, S) \Rightarrow (a, a)$

ii) $S \Rightarrow (L) \Rightarrow (L, S) \Rightarrow (S, S) \Rightarrow (a, S) \Rightarrow (a, (L)) \Rightarrow (a, (L, S)) \Rightarrow (a, (S, S)) \Rightarrow (a, (a, S)) \Rightarrow (a, (a, a))$

iii) $S \Rightarrow (L) \Rightarrow (L, S) \Rightarrow (S, S) \Rightarrow (a, S) \Rightarrow (a, (L)) \Rightarrow (a, (L, S)) \Rightarrow (a, (S, S)) \Rightarrow (a, ((L, S))) \Rightarrow (a, ((L, S), S)) \Rightarrow (a, ((S, S), S)) \Rightarrow (a, ((a, S), S)) \Rightarrow (a, ((a, a), S)) \Rightarrow (a, ((a, a), (L))) \Rightarrow (a, ((a, a), (L, S))) \Rightarrow (a, ((a, a), (S, S))) \Rightarrow (a, ((a, a), (a, S))) \Rightarrow (a, ((a, a), (a, a)))$

d) 构造 b) 中句子的最右推导

- i) $S \Rightarrow (L) \Rightarrow (L, S) \Rightarrow (L, a) \Rightarrow (S, a) \Rightarrow (a, a)$
- ii) $S \Rightarrow (L) \Rightarrow (L, S) \Rightarrow (L, (L)) \Rightarrow (L, (L, S)) \Rightarrow (L, (L, a)) \Rightarrow (L, (S, a)) \Rightarrow (L, (a, a)) \Rightarrow (S, (a, a)) \Rightarrow (a, (a, a))$
- iii) $S \Rightarrow (L) \Rightarrow (L, S) \Rightarrow (L, (L)) \Rightarrow (L, (L, S)) \Rightarrow (L, (L, (L))) \Rightarrow (L, (L, (L, S))) \Rightarrow (L, (L, (L, (L, S)))) \Rightarrow (L, (L, (L, (S, a)))) \Rightarrow (L, (L, (L, (a, a)))) \Rightarrow (L, (L, (S, (a, a)))) \Rightarrow (L, ((L), (a, a))) \Rightarrow (L, ((L, S), (a, a))) \Rightarrow (L, ((L, a), (a, a))) \Rightarrow (L, ((S, a), (a, a))) \Rightarrow (L, ((a, a), (a, a))) \Rightarrow (S, ((a, a), (a, a))) \Rightarrow (a, ((a, a), (a, a)))$

e) 该文法产生的语言是什么

解：设该文法产生语言（符号串集合）L，则

$$L = \{ (A_1, A_2, \dots, A_n) \mid n \text{ 是任意正整数, } A_i = a, \text{ 或 } A_i \in L, i \text{ 是 } 1 \sim n \text{ 之间的整数} \}$$

(Aho) 4.2 考虑文法

$$S \rightarrow aSbS \mid bSaS \mid \varepsilon$$

a) 为句子构造两个不同的最左推导，以证明它是二义性的

$$S \Rightarrow aSbS \Rightarrow abS \Rightarrow abaSbS \Rightarrow ababS \Rightarrow abab$$

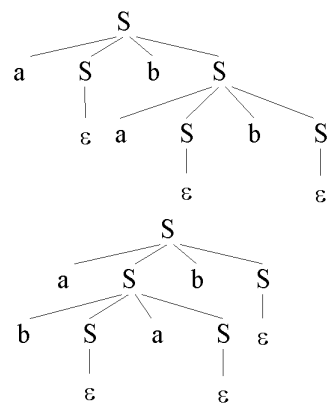
$$S \Rightarrow aSbS \Rightarrow abSaSbS \Rightarrow abaSbS \Rightarrow ababS \Rightarrow abab$$

b) 构造 abab 对应的最右推导

$$S \Rightarrow aSbS \Rightarrow aSbaSbS \Rightarrow aSbaSb \Rightarrow aSbab \Rightarrow abab$$

$$S \Rightarrow aSbS \Rightarrow aSb \Rightarrow abSaSb \Rightarrow abSab \Rightarrow abab$$

c) 构造 abab 对应语法树



d) 该文法产生什么样的语言？

解：生成的语言：a、b 个数相等的 a、b 串的集合

(Aho) 4.3 考虑文法

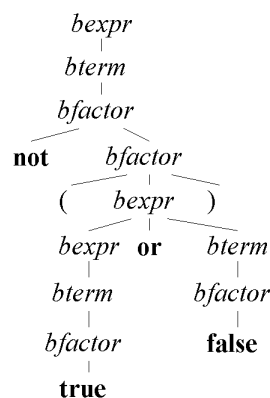
$$bexpr \rightarrow bexpr \text{ or } bterm \mid bterm$$

$$bterm \rightarrow bterm \text{ and } bfactor \mid bfactor$$

$$bfactor \rightarrow \text{not } bfactor \mid (bexpr) \mid \text{true} \mid \text{false}$$

a) 试为句子 not (true or false) 构造分析树

解:



b) 试证明该文法产生所有布尔表达式

证明:

一、首先证明文法产生的所有符号串都是布尔表达式

变换命题形式——以 *bexpr*、*bterm*、*bfactor* 开始的推导得到的所有符号串都是布尔表达式

最短的推导过程得到 **true**、**false**，显然成立

假定对步数小于 *n* 的推导命题都成立

考虑步数等于 *n* 的推导，其开始推导步骤必为以下情况之一

$bexpr \Rightarrow bexpr \text{ or } bterm$

$bexpr \Rightarrow bterm$

$bterm \Rightarrow bterm \text{ and } bfactor$

$bexpr \Rightarrow bfactor$

$bfactor \Rightarrow \text{not } bfactor$

$bfactor \Rightarrow (bexpr)$

而后继推导的步数显然 $< n$ ，因此由归纳假设，第二步句型中的 NT 推导出的串均为布尔表达式，这些布尔表达式经过 *or*、*and*、*not* 运算或加括号，得到的仍是布尔表达式

因此命题一得证。

二、证明所有布尔表达式均可由文法生成

变换命题——所有析取式均可由 *bexpr* 推导出来，所有合取式均可由 *bterm* (*bexpr*) 推导出来，所有对子布尔表达式施加 *not* 运算或加括号或简单 **true**、**false** 都可由 *bfactor* (*bexpr*、*bterm*) 推导出来

最简单的布尔表达式 **true** 和 **false** 显然成立

假定对长度小于 *n* 的布尔表达式，均可由文法推导出来

考虑长度等于 *n* 的布尔表达式 *B*，显然，*B* 只能是以下形式之一

$B = B_1 \text{ or } B_2$

$B = B_1 \text{ and } B_2$

$B = \text{not } B_1$

$B = (B_1)$

以上几种情况，*B*₁、*B*₂ 的长度均小于 *n*

对于情况 1: *B* 为析取式，*B*₁ 可为析取式也可合取式，*B*₂ 为合取式，根据假设可由 *bexpr* 合 *bterm* 推导出来，显然可构造推导过程，由 *bexpr* 推导出 *B*

其他情况类似，命题二得证

综合一、二，可知文法产生的语言就是布尔表达式

c)

(Aho) 4.4 考虑文法

$$R \rightarrow R \mid R \mid RR \mid R^* \mid (R) \mid a \mid b$$

c) 构造等价的非二义性文法

$$R \rightarrow R \mid A \mid A$$
$$A \rightarrow AB \mid B$$
$$B \rightarrow B^* \mid C$$
$$C \rightarrow (R) \mid a \mid b$$

(Aho) 4.5 下面 if-then-else 文法试图消除空悬 else 的二义性，证明它仍是二义性的

$$stmt \rightarrow \text{if } expr \text{ then } stmt$$
$$\quad \mid matched_stmt$$
$$matched_stmt \rightarrow \text{if } expr \text{ then } matched_stmt \text{ else } stmt$$
$$\quad \mid \text{other}$$

解：

用 S、M、E 表示 *stmt*、*matched_stmt* 和 *expr*，用 i、t、e、o 表示 **if**、**then**、**else** 和 **other** 则句子 *iEt iEtoeiEtoeo* 对应两个最左推导：

$$\begin{aligned} S &\Rightarrow iEt \underline{S} \Rightarrow iEt \underline{M} \Rightarrow iEt iEt \underline{M} eS \Rightarrow iEt iEtoe \underline{S} \Rightarrow iEt iEtoe \underline{M} \\ &\Rightarrow iEt iEtoeiEt \underline{M} eS \Rightarrow iEt iEtoeiEtoe \underline{S} \Rightarrow iEt iEtoeiEtoe \underline{M} \\ &\Rightarrow iEt iEtoeiEtoeo \end{aligned}$$
$$\begin{aligned} S &\Rightarrow \underline{M} \Rightarrow iEt \underline{M} eS \Rightarrow iEt iEt \underline{M} eSeS \Rightarrow iEt iEtoe \underline{S} eS \Rightarrow iEt iEtoeiEt \underline{S} eS \\ &\Rightarrow iEt iEtoeiEt \underline{M} eS \Rightarrow iEt iEtoeiEtoe \underline{S} \Rightarrow iEt iEtoeiEtoe \underline{M} \\ &\Rightarrow iEt iEtoeiEtoeo \end{aligned}$$

因此文法是二义性文法

直接构造比较困难，可从 SLR 分析表的构造角度考虑，LR(0)项目集规范族中，项目

$I_8 = \{M \rightarrow iEtM \cdot eS, S \rightarrow \cdot M\}$ ，有移进/归约冲突，这就是二义性所在。

显然，存在句型...*iEtMeS*...和...*iEtSeS*...，当 *M* 位于栈顶时，产生移进/归约冲突。

按此思路，构造形如...*iEtSeS*...的句型即可

(Aho) 4.6 为下列语言设计上下文无关文法。哪些语言是正规式？

a) 满足这样条件的二进制串：每个 0 之后都紧跟着至少一个 1

$$S \rightarrow 0A \mid 1S \mid \varepsilon$$
$$A \rightarrow 1S$$

正规式： $(1 \mid 01)^*$

b) 0 和 1 个数相等的二进制串

$$S \rightarrow 0S1S \mid 1S0S \mid \varepsilon$$

d) 不含 011 子串的二进制串

$$S \rightarrow 0A \mid 1S \mid \varepsilon$$
$$A \rightarrow 0A \mid 1B \mid \varepsilon$$

$B \rightarrow 0A \mid \varepsilon$

正规式: $1^*(0 \mid 01)^*$

e) 具有形式 xy 的二进制串, $x \neq y$

$S \rightarrow A \mid B \mid AB \mid BA$

$A \rightarrow DAD \mid 0$

$B \rightarrow DBD \mid 1$

$D \rightarrow 0 \mid 1$

A、B 分别表示中心符号为 0、1 的长度为奇数的二进制串

将 AB 串接, 长度为偶数, 将它从中间分为长度相等的两部分, x 、 y

虽然 A、B 长度可能不一样, 但容易得到, A 的中心 0 在 x 中的位置, 与 B 的中心 1 在 y 中的位置是相同的, 因此 $x \neq y$

BA 的情况类似

f) 形如 xx 的 0、1 串

解: 此语言无法用上下文无关文法描述

(Aho) 4.11 对习题 4.1 中文法

a) 消除左递归

$S \rightarrow (L) \mid a$

$L \rightarrow S L'$

$L' \rightarrow , S L' \mid \varepsilon$

b) 构造预测分析表, 对 4.1(b)中句子, 给出预测分析器的运行过程

$FIRST(S) = \{ (, a \}$

$FIRST(L) = \{ (, a \}$

$FIRST(L') = \{ ', , \varepsilon \}$

$FOLLOW(S) = \{ ', ,), \$ \}$

$FOLLOW(L) = \{) \}$

$FOLLOW(L') = \{) \}$

预测分析表:

	a	()	,	\$
S	$S \rightarrow a$	$S \rightarrow (L)$			
L	$L \rightarrow S L'$	$L \rightarrow S L'$			
L'			$L' \rightarrow \varepsilon$	$L' \rightarrow , S L'$	

(a, a)的分析过程

栈	输入	输出
\$S	(a, a)\$	
\$) L ((a, a)\$	$S \rightarrow (L)$
\$) L	a, a)\$	$L \rightarrow S L'$
\$) L' S	a, a)\$	$S \rightarrow a$
\$) L' a	a, a)\$	
\$) L'	, a)\$	$L' \rightarrow , S L'$

\$) L' S ,	, a)\$	
\$) L' S	a)\$	$S \rightarrow a$
\$) L' a	a)\$	
\$) L')\$	$L' \rightarrow \varepsilon$
\$))\$	
\$	\$	

其他两个句子的分析过程类似

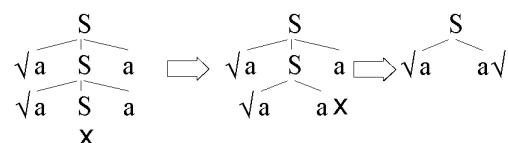
(Aho) 4.13 下面文法产生除 ε 外所有长度为偶数的 a 的串

$$S \rightarrow aSa | aa$$

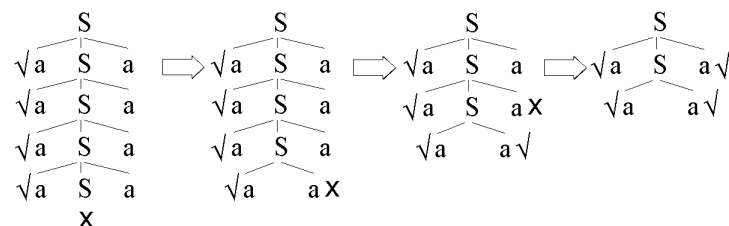
a) 试为该文法构造一个带回溯的递归下降语法分析器，对 S 的两个候选式首先考虑 aSa。

证明：S 所对应的过程可以成功分析 2、4、8 个 a 的串，但 6 个 a 的串不行。

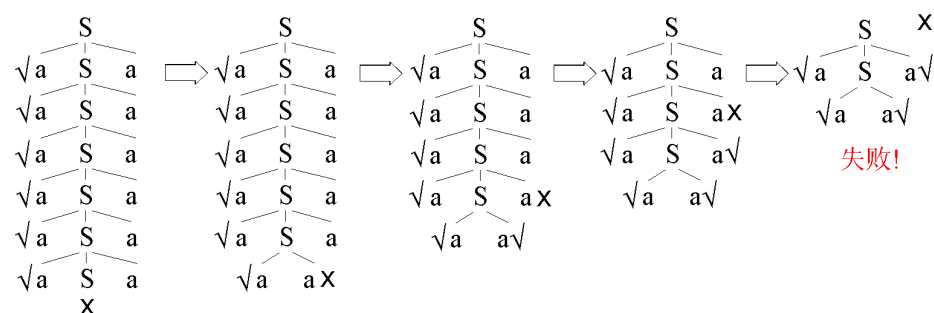
解：aa 的分析过程，其中 \checkmark 表示匹配成功， \times 表示匹配失败，匹配失败则尝试下个候选式



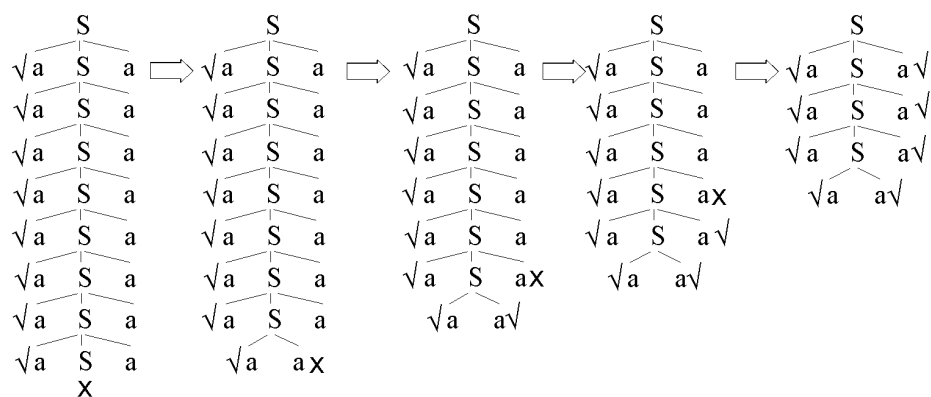
aaaa 分析过程：



aaaaaa 分析过程：



aaaaaaaa 分析过程：



b) 此语法分析器能识别什么样的语言?

解: 由 a) 的解可以看出, $2N$ 个 a 的串分析过程中, 步骤如下

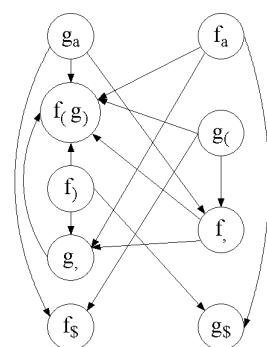
- 1) 产生 $2N+1$ 个 S 的语法树, 对第 $2N+1$ 个 S 进行扩展时输入缓冲已空, 失败
- 2) 对第 $2N$ 个 S 尝试候选式 aa , 第二个 a 匹配失败
- 3) 对第 $2N-1$ 个 S 尝试候选式 aa , 左边 $N-1$ 个 a 匹配, 右边最后一个 a 匹配, 倒数第二个 a 失败
- 4) 对第 $2N-2$ 个 S 尝试候选式 aa , 左边 $N-2$ 个 a 匹配, 右边最后一个 a 和倒数第二个 a 匹配, 倒数第三个 a 失败
- 5) 对第 $2N-4$ 个 S 尝试候选式 aa , 左边 $N-4$ 个 a 匹配, 右边最后一个 a ~ 倒数第四个 a 匹配, 倒数第五个 a 失败
- 6) 对第 $2N-8$ 个 S 尝试候选式 aa , ...

最后正确识别的情况必然是: 对第 N 个 S 尝试 aa , 左边 N 个 a 和右边 N 个 a 恰与输入匹配
显然, 可以正确识别的符号串的 N 满足

$$2N - 1 - 1 - 2 - 4 - \dots = N \rightarrow N = 2^i$$

(Aho) 4.25 试给出图 4-60 中的优先关系表对应的优先级函数

解: 有向图如下



优先级函数为

	a	()	,	\$
f	2	0	2	2	0
g	3	3	0	1	0

(Aho) 4.26 对习题 4.1 中文法，利用讲义中给出的算法计算终结符之间的优先关系

解： $S \rightarrow (L) | a$
 $L \rightarrow L, S | S$

由于 $S \rightarrow (L)$ ，因此 (\doteq)

$S \rightarrow (L)$ ，而 $L \Rightarrow L, S$ ， $L \Rightarrow S \Rightarrow (L)$ ， $L \Rightarrow S \Rightarrow a$

因此 $(< , , (< (, (< a ; , >) ,) >) , a >)$

由于 $L \rightarrow L, S$ ，而 $L \Rightarrow L, S$ ， $L \Rightarrow S \Rightarrow (L)$ ， $L \Rightarrow S \Rightarrow a$ ，因此 $, > , ,) > , , a > ,$

而 $S \Rightarrow (L)$ ， $S \Rightarrow a$ ，因此 $, < (, , < a$

非终结符与\$优先关系的计算方法：

如果存在 $S \Rightarrow a \dots$ ，或 $S \Rightarrow Qa \dots$ ，则 $\$ < a$ ，

若存在 $S \Rightarrow \dots a$ ，或 $S \Rightarrow \dots aQ$ ，则 $a > \$$

因此， $\$ < (, \$ < a ,) > \$, a > \$$

算符优先关系表为：

	a	()	,	\$
a			>	>	>
(<	<	\doteq	<	
)			>	>	>
,	<	<	>	>	
\$	<	<			

(Aho) 4.27 试给下列文法构造算符优先关系

a) 练习 4.2 中文法

解：

$S \rightarrow aSbS \mid bSaS \mid \varepsilon$

由 $S \rightarrow aSbS$ 可得 $a \doteq b$

由 $S \rightarrow bSaS$ 可得 $b \doteq a$

由 $S \rightarrow aSbS$ ，和 $S \Rightarrow bSaS$ 可得 $a < b$ 、 $b < b$ 、 $a > b$ ，和 $S \Rightarrow aSbS$ 可得 $a < a$ 、 $b < a$ 、 $b > b$

由 $S \rightarrow bSaS$, 和 $S \Rightarrow bSaS$ 可得 $b < b$ 、 $a < b$ 、 $a > a$, 和 $S \Rightarrow aSbS$ 可得 $b < a$ 、 $a < a$ 、 $a > a$

文法不是算符优先文法, 二义性文法, 很自然

b) 练习 4.3 中文法

$bexpr \rightarrow bexpr \text{ or } bterm \mid bterm$

$bterm \rightarrow bterm \text{ and } bfactor \mid bfactor$

$bfactor \rightarrow \text{not } bfactor \mid (bexpr) \mid \text{true} \mid \text{false}$

解: 无论分析语法含义, 还是利用 4.26 算法计算, 均可得到

	true	false	not	and	or	()	\$
true				>	>		>	>
false				>	>		>	>
not	<	<	<	>	>	<	>	>
and	<	<	<	>	>	<	>	>
or	<	<	<	<	>	<	>	>
(<	<	<	<	<	<	\doteq	
)				>	>		>	>
\$	<	<	<	<	<	<		

(Aho) 4.30 一个文法称为 Greibach 范式 (GNF) 文法, 如果它无 ϵ 产生式, 且每个产生式 ($S \rightarrow \epsilon$ 除外) 均形如 $A \rightarrow a\alpha$, 其中, a 是终结符, α 是非终结符串, 也可能为空。

a) 试编写一个算法, 将给定文法转换为 Greibach 范式

解: 算法步骤如下

1. 先将文法消除左递归、消除 ϵ 产生式、删除无用符号, 然后对每个非终结符 A 的每个产生式, 执行 2
2. 若产生式右部以终结符开始, 则略过, 考虑其他产生式, 否则产生式必为 $A \rightarrow A_1\alpha$ 的形式, A_1 为 $\neq A$ 的 NT, α 为语法符号串, 对它执行以下操作
 - i. 将 A_1 的所有产生式的右部替换 A_1 , 产生新的关于 A 的产生式
 - ii. 对于这些产生式, 若右部以 T 开始, 略过, 不予处理, 考虑那些以 NT 开始的产生式, 反复执行 i、ii
 - iii. 由于文法的 NT 个数是有限的 (设为 k), 且已消除左递归, 则最多 k 个步骤后, 处理完毕, 此时, A 的产生式右部应该均以 T 开始。否则, 若某个产生式右部以 NT 开始, 表明 A 无论经过怎样的推导过程, 均不可能得到一个以终结符开始的串, 当然也就不可能得到一个终结符串, 这显然是一个错误的文法, 矛盾。这样, A 的某个产生式处理完毕, 其右部均以 T 开始。转向 2, 继续考虑其他产生式, 所有产生式处理完毕, 则转向 3

3. 此时，每个产生式均为 $A \rightarrow a\alpha$ 的形成， a 为 NT， α 为语法符号串，若 α 中包含 T ，则进行如下处理

i. 假定 α 中包含 k 个 T ，则产生式形为 $A \rightarrow a\alpha_0 a_1 \alpha_1 \dots a_k \alpha_k$ ，其中 a_i 为 T ， α_i 为 NT 串或 ϵ

ii. 引入 k 个新的 NT A_1, A_2, \dots, A_k ，和 k 个新的产生式

$$A_1 \rightarrow a_1 \alpha_1 A_2$$

$$A_2 \rightarrow a_2 \alpha_2 A_3$$

...

$$A_{k-1} \rightarrow a_{k-1} \alpha_{k-1} A_k$$

$$A_k \rightarrow a_k \alpha_k$$

而将原产生式改为 $A \rightarrow a\alpha_0 A_1$

4. 经过 2、3 处理，所有产生式必然满足 Greibach 范式的格式

b) 将你的算法应用到表达式文法 4-10 上

解：文法 4-10 消除左递归、消除 ϵ 产生式后得到

$$E \rightarrow T E' \mid T$$

$$E' \rightarrow + T E' \mid + T$$

$$T \rightarrow F T' \mid F$$

$$T' \rightarrow * F T' \mid * F$$

$$F \rightarrow (E) \mid \text{id}$$

将每个产生式转换为以 T 开始的形式，得到

$$E \rightarrow (E) T' E' \mid \text{id} T' E' \mid (E) E' \mid \text{id} E' \mid (E) T' \mid \text{id} T' \mid (E) \mid \text{id}$$

$$E' \rightarrow + T E' \mid + T$$

$$T \rightarrow (E) T' \mid \text{id} T' \mid (E) \mid \text{id}$$

$$T' \rightarrow * F T' \mid * F$$

$$F \rightarrow (E) \mid \text{id}$$

将每个产生式右部转换为 $T N T^*$ 的形式，最后结果为：

$$E \rightarrow (E E'' \mid \text{id} T' E' \mid (E E''' \mid \text{id} E' \mid (E \mid \text{id} T' \mid (E E'''' \mid \text{id}$$

$$E'' \rightarrow) T' E'$$

$$E''' \rightarrow) E'$$

$$E'''' \rightarrow) T'$$

$$E'''' \rightarrow)$$

$$E' \rightarrow + T E' \mid + T$$

$$T \rightarrow (E E'''' \mid \text{id} T' \mid (E E'''' \mid \text{id}$$

$$T' \rightarrow * F T' \mid * F$$

$$F \rightarrow (E E'''' \mid \text{id}$$

(Aho) 4.33 考虑文法

$$S \rightarrow AS \mid b$$

$$A \rightarrow SA \mid a$$

a) 构造此文法的 LR(0) 项目集规范族

解：

$$I_0 = \{ S' \rightarrow \cdot S, S \rightarrow \cdot AS, S \rightarrow \cdot b, A \rightarrow \cdot SA, A \rightarrow \cdot a \}$$

$$\text{goto}(I_0, S) = \{ S' \rightarrow S \cdot, A \rightarrow S \cdot A, A \rightarrow \cdot SA, A \rightarrow \cdot a, S \rightarrow \cdot AS, S \rightarrow \cdot b \} = I_1$$

$\text{goto}(I_0, A) = \{ S \rightarrow A \cdot S, S \rightarrow \cdot AS, S \rightarrow \cdot b, A \rightarrow \cdot SA, A \rightarrow \cdot a \} = I_2$
 $\text{goto}(I_0, a) = \{ A \rightarrow a \cdot \} = I_3$
 $\text{goto}(I_0, b) = \{ S \rightarrow b \cdot \} = I_4$
 $\text{goto}(I_1, S) = \{ A \rightarrow S \cdot A, A \rightarrow \cdot SA, A \rightarrow \cdot a, S \rightarrow \cdot AS, S \rightarrow \cdot b \} = I_5$
 $\text{goto}(I_1, A) = \{ A \rightarrow SA \cdot, S \rightarrow A \cdot S, S \rightarrow \cdot AS, S \rightarrow \cdot b, A \rightarrow \cdot SA, A \rightarrow \cdot a \} = I_6$
 $\text{goto}(I_1, a) = I_3, \quad \text{goto}(I_1, b) = I_4$
 $\text{goto}(I_2, S) = \{ S \rightarrow AS \cdot, A \rightarrow S \cdot A, A \rightarrow \cdot SA, A \rightarrow \cdot a, S \rightarrow \cdot AS, S \rightarrow \cdot b \} = I_7$
 $\text{goto}(I_2, A) = I_2, \quad \text{goto}(I_2, a) = I_3, \quad \text{goto}(I_2, b) = I_4$
 $\text{goto}(I_5, S) = I_5, \quad \text{goto}(I_5, A) = I_6, \quad \text{goto}(I_5, a) = I_3, \quad \text{goto}(I_5, b) = I_4$
 $\text{goto}(I_6, S) = I_7, \quad \text{goto}(I_6, A) = I_2, \quad \text{goto}(I_6, a) = I_3, \quad \text{goto}(I_6, b) = I_4$
 $\text{goto}(I_7, S) = I_5, \quad \text{goto}(I_7, A) = I_6, \quad \text{goto}(I_7, a) = I_3, \quad \text{goto}(I_7, b) = I_4$

c) 构造 SLR 分析表

解:

$\text{FIRST}(S) = \text{FIRST}(A) = \{a, b\}$

$\text{FOLLOW}(S) = \{\$, a, b\} \quad \text{FOLLOW}(A) = \{a, b\}$

SLR 分析表为:

	action			goto	
	a	b	\$	S	A
0	s3	s4		1	2
1	s3	s4	acc	5	6
2	s3	s4		7	2
3	r4	r4			
4	r2	r2	r2		
5	s3	s4		5	6
6	s3/r3	s4/r3		7	2
7	s3/r1	s4/r1	r1	5	6

SLR 分析表冲突，分析过程有多种可能路径，选择其中一种导致正确结果的即可。

d) 对输入串 abab，给出 SLR 分析器运行过程

栈	输入	动作
0	abab\$	移进
0a3	bab\$	归约 $A \rightarrow a$
0A2	bab\$	移进
0A2b4	ab\$	归约 $S \rightarrow b$
0A2S7	ab\$	归约 $S \rightarrow AS$
0S1	ab\$	移进
0S1a3	b\$	归约 $A \rightarrow a$
0S1A6	b\$	归约 $A \rightarrow SA$
0A2	b\$	移进
0A2b4	\$	归约 $S \rightarrow b$
0A2S7	\$	归约 $S \rightarrow AS$
0S1	\$	接受

e) 构造规范 LR 分析表

解:

$I_0 = \{ [S' \rightarrow \cdot S, \$], [S \rightarrow \cdot AS, \$/a/b], [S \rightarrow \cdot b, \$/a/b], [A \rightarrow \cdot SA, a/b], [A \rightarrow \cdot a, a/b] \}$
 $goto(I_0, S) = \{ [S' \rightarrow S \cdot, \$], [A \rightarrow S \cdot A, a/b], [A \rightarrow \cdot SA, a/b], [A \rightarrow \cdot a, a/b], [S \rightarrow \cdot AS, a/b], [S \rightarrow \cdot b, a/b] \}$
 $= I_1$
 $goto(I_0, A) = \{ [S \rightarrow A \cdot S, \$/a/b], [S \rightarrow \cdot AS, \$/a/b], [S \rightarrow \cdot b, \$/a/b], [A \rightarrow \cdot SA, a/b], [A \rightarrow \cdot a, a/b] \} = I_2$
 $goto(I_0, a) = \{ [A \rightarrow a \cdot, a/b] \} = I_3, \quad goto(I_0, b) = \{ [S \rightarrow b \cdot, \$/a/b] \} = I_4$
 $goto(I_1, S) = \{ [A \rightarrow S \cdot A, a/b], [A \rightarrow \cdot SA, a/b], [A \rightarrow \cdot a, a/b], [S \rightarrow \cdot AS, a/b], [S \rightarrow \cdot b, a/b] \} = I_5$
 $goto(I_1, A) = \{ [A \rightarrow SA \cdot, a/b], [S \rightarrow A \cdot S, a/b], [S \rightarrow \cdot AS, a/b], [S \rightarrow \cdot b, a/b], [A \rightarrow \cdot SA, a/b], [A \rightarrow \cdot a, a/b] \} = I_6$
 $goto(I_1, a) = I_3, \quad goto(I_1, b) = \{ [S \rightarrow b \cdot, a/b] \} = I_7$
 $goto(I_2, S) = \{ [S \rightarrow AS \cdot, \$/a/b], [A \rightarrow S \cdot A, a/b], [A \rightarrow \cdot SA, a/b], [A \rightarrow \cdot a, a/b], [S \rightarrow \cdot AS, a/b], [S \rightarrow \cdot b, a/b] \} = I_8$
 $goto(I_2, A) = I_2, \quad goto(I_2, a) = I_3, \quad goto(I_2, b) = I_4$
 $goto(I_5, S) = I_5, \quad goto(I_5, A) = I_6, \quad goto(I_5, a) = I_3, \quad goto(I_5, b) = I_7$
 $goto(I_6, S) = \{ [S \rightarrow AS \cdot, a/b], [A \rightarrow S \cdot A, a/b], [A \rightarrow \cdot SA, a/b], [A \rightarrow \cdot a, a/b], [S \rightarrow \cdot AS, a/b], [S \rightarrow \cdot b, a/b] \} = I_9$
 $goto(I_6, A) = \{ [S \rightarrow A \cdot S, a/b], [S \rightarrow \cdot AS, a/b], [S \rightarrow \cdot b, a/b], [A \rightarrow \cdot SA, a/b], [A \rightarrow \cdot a, a/b] \} = I_{10}$
 $goto(I_6, a) = I_3, \quad goto(I_6, b) = I_7$
 $goto(I_8, S) = I_5, \quad goto(I_8, A) = I_6, \quad goto(I_8, a) = I_3, \quad goto(I_8, b) = I_7$
 $goto(I_9, S) = I_5, \quad goto(I_9, A) = I_6, \quad goto(I_9, a) = I_3, \quad goto(I_9, b) = I_7$
 $goto(I_{10}, S) = I_9, \quad goto(I_{10}, A) = I_{10}, \quad goto(I_{10}, a) = I_3, \quad goto(I_{10}, b) = I_7$

规范 LR 分析表为:

	action			goto	
	a	b	\$	S	A
0	s3	s4		1	2
1	s3	s7	acc	5	6
2	s3	s4		8	2
3	r4	r4			
4	r2	r2	r2		
5	s3	s7		5	6
6	s3/r3	s7/r3		9	10
7	r2	r2			
8	s3/r1	s7/r1	r1	5	6
9	s3/r1	s7/r1		5	6
10	s3	s7		9	10

f) 利用 LR(1)项目集合并的方法构造 LALR 分析表

解:

同心集合并:

$I_{210} = \{ [S \rightarrow A \cdot S, \$/a/b], [S \rightarrow \cdot AS, \$/a/b], [S \rightarrow \cdot b, \$/a/b], [A \rightarrow \cdot SA, a/b], [A \rightarrow \cdot a, a/b] \}$
 $I_{47} = \{ [S \rightarrow b \cdot, \$/a/b] \}$
 $I_{89} = \{ [S \rightarrow AS \cdot, \$/a/b], [A \rightarrow S \cdot A, a/b], [A \rightarrow \cdot SA, a/b], [A \rightarrow \cdot a, a/b], [S \rightarrow \cdot AS, a/b], [S \rightarrow \cdot b, a/b] \}$

LALR 分析表为:

	action			goto	
	a	b	\$	S	A
0	s3	s47		1	2 10
1	s3	s47	acc	5	6
2 10	s3	s47		89	2 10
3	r4	r4			
47	r2	r2	r2		
5	s3	s47		5	6
6	s3/r3	s47/r3		89	2 10
89	s3/r1	s47/r1	r1	5	6

g) 利用高效构造方法构造 LALR 分析表

解: LR(0)项目集的核:

$I_0 = \{ S' \rightarrow \cdot S \}$

$I_1 = \{ S' \rightarrow S \cdot, A \rightarrow S \cdot A \}$

$I_2 = \{ S \rightarrow A \cdot S \}$

$I_3 = \{ A \rightarrow a \cdot \}$

$I_4 = \{ S \rightarrow b \cdot \}$

$I_5 = \{ A \rightarrow S \cdot A \}$

$I_6 = \{ A \rightarrow SA \cdot, S \rightarrow A \cdot S \}$

$I_7 = \{ S \rightarrow AS \cdot, A \rightarrow S \cdot A \}$

$\text{closure}([S' \rightarrow \cdot S, \#]) = \{ [S' \rightarrow \cdot S, \#], [S \rightarrow \cdot AS, \#/a/b], [S \rightarrow \cdot b, \#/a/b], [A \rightarrow \cdot SA, a/b], [A \rightarrow \cdot a, a/b] \}$

$I_0: S' \rightarrow \cdot S$ 传播到 $I_1: S' \rightarrow S \cdot$, $I_2: S \rightarrow A \cdot S$, $I_4: S \rightarrow b \cdot$

自生搜索符 a/b: $I_1: A \rightarrow S \cdot A$, $I_2: S \rightarrow A \cdot S$, $I_3: A \rightarrow a \cdot$, $I_4: S \rightarrow b \cdot$

$\text{closure}([A \rightarrow \cdot S \cdot A, \#]) = \{ [A \rightarrow \cdot S \cdot A, \#], [A \rightarrow \cdot SA, \#/a/b], [A \rightarrow \cdot a, \#/a/b], [S \rightarrow \cdot AS, a/b], [S \rightarrow \cdot b, a/b] \}$

$I_1: A \rightarrow \cdot S \cdot A$ 传播到 $I_6: A \rightarrow SA \cdot$, $I_5: A \rightarrow S \cdot A$, $I_3: A \rightarrow a \cdot$

自生搜索符 a/b: $I_5: A \rightarrow S \cdot A$, $I_6: S \rightarrow A \cdot S$, $I_3: A \rightarrow a \cdot$, $I_4: S \rightarrow b \cdot$

$\text{closure}([S \rightarrow \cdot A \cdot S, \#]) = \{ [S \rightarrow \cdot A \cdot S, \#], [S \rightarrow \cdot AS, \#/a/b], [S \rightarrow \cdot b, \#/a/b], [A \rightarrow \cdot SA, a/b], [A \rightarrow \cdot a, a/b] \}$

$I_2: S \rightarrow \cdot A \cdot S$ 传播到 $I_7: S \rightarrow AS \cdot$, $I_4: S \rightarrow b \cdot$

自生搜索符 a/b: $I_2: S \rightarrow A \cdot S$, $I_7: A \rightarrow S \cdot A$, $I_3: A \rightarrow a \cdot$, $I_4: S \rightarrow b \cdot$

$I_5: A \rightarrow \cdot S \cdot A$ 传播到 $I_6: A \rightarrow SA \cdot$, $I_3: A \rightarrow a \cdot$

自生搜索符 a/b: $I_5: A \rightarrow S \cdot A$, $I_6: S \rightarrow A \cdot S$, $I_3: A \rightarrow a \cdot$, $I_4: S \rightarrow b \cdot$

$I_6: S \rightarrow A \cdot S$ 传播到 $I_7: S \rightarrow AS \cdot$, $I_2: S \rightarrow A \cdot S$, $I_4: S \rightarrow b \cdot$

自生搜索符 a/b: $I_2: S \rightarrow A \cdot S$, $I_7: A \rightarrow S \cdot A$, $I_3: A \rightarrow a \cdot$, $I_4: S \rightarrow b \cdot$

$\text{closure}([A \rightarrow \cdot S \cdot A, \#]) = \{ [A \rightarrow \cdot S \cdot A, \#], [A \rightarrow \cdot SA, \#/a/b], [A \rightarrow \cdot a, \#/a/b], [S \rightarrow \cdot AS, a/b], [S \rightarrow \cdot b, a/b] \}$

$I_7: A \rightarrow \cdot S \cdot A$ 传播到 $I_6: A \rightarrow SA \cdot$, $I_5: A \rightarrow S \cdot A$, $I_3: A \rightarrow a \cdot$

自生搜索符 a/b: I₅: A→S·A, I₆: S→A·S, I₃: A→a·, I₄: S→b·

计算搜索符:

集合	项目	搜索符		
		初始	第一遍	第二遍
I ₀	S'→·S	\$	\$	\$
I ₁	S'→S·		\$	\$
I ₁	A→S·A	a/b	a/b	a/b
I ₂	S→A·S	a/b	\$/a/b	\$/a/b
I ₃	A→a·	a/b	a/b	a/b
I ₄	S→b·	a/b	\$/a/b	\$/a/b
I ₅	A→S·A	a/b	a/b	a/b
I ₆	A→SA·		a/b	a/b
I ₆	S→A·S	a/b	a/b	a/b
I ₇	S→AS·		a/b	\$/a/b
I ₇	A→S·A	a/b	a/b	a/b

LALR 分析表为:

	action			goto	
	a	b	\$	S	A
0	s3	s4		1	2
1	s3	s4	acc	5	6
2	s3	s4		7	2
3	r4	r4			
4	r2	r2	r2		
5	s3	s4		5	6
6	s3/r3	s4/r3		7	2
7	s3/r1	s4/r1	r1	5	6

(Aho) 4.35 考虑下面文法

$E \rightarrow E + T \mid T$

$T \rightarrow T F \mid F$

$F \rightarrow F * \mid a \mid b$

a) 构造 SLR 分析表

解: 拓广文法

$E' \rightarrow E$

$I_0 = \{ E' \rightarrow \cdot E, E \rightarrow \cdot E + T, E \rightarrow \cdot T, T \rightarrow \cdot T F, T \rightarrow \cdot F, F \rightarrow \cdot F *, F \rightarrow \cdot a, F \rightarrow \cdot b \}$

$\text{goto}(I_0, E) = \{ E' \rightarrow E \cdot, E \rightarrow E \cdot + T \} = I_1$

$\text{goto}(I_0, T) = \{ E \rightarrow T \cdot, T \rightarrow T \cdot F, F \rightarrow \cdot F *, F \rightarrow \cdot a, F \rightarrow \cdot b \} = I_2$

$\text{goto}(I_0, F) = \{ T \rightarrow F \cdot, F \rightarrow F \cdot * \} = I_3$

$\text{goto}(I_0, a) = \{ F \rightarrow a \cdot \} = I_4$

$\text{goto}(I_0, b) = \{ F \rightarrow b \cdot \} = I_5$

$\text{goto}(I_1, +) = \{ E \rightarrow E + \cdot T, T \rightarrow \cdot T F, T \rightarrow \cdot F, F \rightarrow \cdot F *, F \rightarrow \cdot a, F \rightarrow \cdot b \} = I_6$

$\text{goto}(I_2, F) = \{ T \rightarrow T F \cdot, F \rightarrow F \cdot * \} = I_7$

$\text{goto}(I_2, a) = I_4$

$\text{goto}(I_2, b) = I_5$

$\text{goto}(I_3, *) = \{ F \rightarrow F * \cdot \} = I_8$

$\text{goto}(I_6, T) = \{ E \rightarrow E + T \cdot, T \rightarrow T \cdot F, F \rightarrow \cdot F *, F \rightarrow \cdot a, F \rightarrow \cdot b \} = I_9$

$\text{goto}(I_6, F) = I_3$

$\text{goto}(I_6, a) = I_4$

$\text{goto}(I_6, b) = I_5$

$\text{goto}(I_7, *) = I_8$

$\text{goto}(I_9, F) = I_7$

$\text{goto}(I_9, a) = I_4$

$\text{goto}(I_9, b) = I_5$

$\text{follow}(E) = \{ +, \$ \}$

$\text{follow}(T) = \{ a, b, +, \$ \}$

$\text{follow}(F) = \{ a, b, *, +, \$ \}$

SLR 分析表:

	action					goto		
	a	b	*	+	\$	E	T	F
0	s4	s5				1	2	3
1				s6	acc			
2	s4	s5		r2	r2			7
3	r4	r4	s8	r4	r4			
4	r6	r6	r6	r6	r6			
5	r7	r7	r7	r7	r7			
6	s4	s5					9	3
7	r3	r3	s8	r3	r3			
8	r5	r5	r5	r5	r5			
9	s4	s5		r1	r1			7

b) 构造 LALR 分析表

解: LR(0)项目集规范族的核

$I_0 = \{ E' \rightarrow \cdot E \}$

$I_1 = \{ E' \rightarrow E \cdot, E \rightarrow E \cdot + T \}$

$I_2 = \{ E \rightarrow T \cdot, T \rightarrow T \cdot F \}$

$I_3 = \{ T \rightarrow F \cdot, F \rightarrow F \cdot * \}$

$I_4 = \{ F \rightarrow a \cdot \}$

$I_5 = \{ F \rightarrow b \cdot \}$

$I_6 = \{ E \rightarrow E + \cdot T \}$

$I_7 = \{ T \rightarrow T F \cdot, F \rightarrow F \cdot * \}$

$I_8 = \{ F \rightarrow F * \cdot \}$

$I_9 = \{ E \rightarrow E + T \cdot, T \rightarrow T \cdot F \}$

$\text{closure}(\{ [E' \rightarrow \cdot E, \#] \}) = \{ [E' \rightarrow \cdot E, \#], [E \rightarrow \cdot E + T, \#], [E \rightarrow \cdot T, +], [T \rightarrow \cdot T F, +/a/b],$

$[T \rightarrow \cdot F, +/a/b], [F \rightarrow \cdot F^*, +/a/b/*], [F \rightarrow \cdot a, +/a/b/*], [F \rightarrow \cdot b, +/a/b/*] \}$

传播: $I_0: E' \rightarrow \cdot E \rightarrow I_1: E' \rightarrow E \cdot, I_1: E \rightarrow E \cdot + T$

自生: $I_2: E \rightarrow T \cdot, I_2: T \rightarrow T \cdot F, I_3: T \rightarrow F \cdot, I_3: F \rightarrow F \cdot *, I_4: F \rightarrow a \cdot, I_5: F \rightarrow b \cdot$

传播: $I_1: E \rightarrow E \cdot + T \rightarrow I_6: E \rightarrow E + \cdot T$

$\text{closure}(\{[T \rightarrow T \cdot F, \#]\}) = \{[T \rightarrow T \cdot F, \#], [F \rightarrow \cdot F^*, \#/*], [F \rightarrow \cdot a, \#/*], [F \rightarrow \cdot b, \#/*] \}$

传播: $I_2: T \rightarrow T \cdot F \rightarrow I_7: T \rightarrow T F \cdot, I_7: F \rightarrow F \cdot *, I_4: F \rightarrow a \cdot, I_5: F \rightarrow b \cdot$

自生: $I_7: F \rightarrow F \cdot *, I_4: F \rightarrow a \cdot, I_5: F \rightarrow b \cdot$

传播: $I_3: F \rightarrow F \cdot * \rightarrow I_8: F \rightarrow F * \cdot$

$\text{closure}(\{[E \rightarrow E + \cdot T, \#]\}) = \{[E \rightarrow E + \cdot T, \#], [T \rightarrow \cdot T F, \#/a/b], [T \rightarrow \cdot F, \#/a/b], [F \rightarrow \cdot F^*, \#/a/b/*], [F \rightarrow \cdot a, \#/a/b/*], [F \rightarrow \cdot b, \#/a/b/*] \}$

传播: $I_6: E \rightarrow E + \cdot T \rightarrow I_9: E \rightarrow E + T \cdot, I_9: T \rightarrow T \cdot F, I_3: T \rightarrow F \cdot, I_3: F \rightarrow F \cdot *, I_4: F \rightarrow a \cdot, I_5: F \rightarrow b \cdot$

自生: $I_9: T \rightarrow T \cdot F, I_3: T \rightarrow F \cdot, I_3: F \rightarrow F \cdot *, I_4: F \rightarrow a \cdot, I_5: F \rightarrow b \cdot$

传播: $I_7: F \rightarrow F \cdot * \rightarrow I_8: F \rightarrow F * \cdot$

$\text{closure}(\{[T \rightarrow T \cdot F, \#]\}) = \{[T \rightarrow T \cdot F, \#], [F \rightarrow \cdot F^*, \#/*], [F \rightarrow \cdot a, \#/*], [F \rightarrow \cdot b, \#/*] \}$

传播: $I_9: T \rightarrow T \cdot F \rightarrow I_7: T \rightarrow T F \cdot, I_7: F \rightarrow F \cdot *, I_4: F \rightarrow a \cdot, I_5: F \rightarrow b \cdot$

自生: $I_3: F \rightarrow F \cdot *, I_4: F \rightarrow a \cdot, I_5: F \rightarrow b \cdot$

计算搜索符:

集合	项目	搜索符		
		初始	第一遍	第二遍
I_0	$E' \rightarrow \cdot E$	\$	\$	\$
I_1	$E' \rightarrow E \cdot$		\$	\$
I_1	$E \rightarrow E \cdot + T$		\$	\$
I_2	$E \rightarrow T \cdot$	+	+	+
I_2	$T \rightarrow T \cdot F$	$+/a/b$	$+/a/b$	$+/a/b$
I_3	$T \rightarrow F \cdot$	$+/a/b$	$+/a/b/\$$	$+/a/b/\$$
I_3	$F \rightarrow F \cdot *$	$+/a/b/*$	$+/a/b/*/\$$	$+/a/b/*/\$$
I_4	$F \rightarrow a \cdot$	$+/a/b/*$	$+/a/b/*/\$$	$+/a/b/*/\$$
I_5	$F \rightarrow b \cdot$	$+/a/b/*$	$+/a/b/*/\$$	$+/a/b/*/\$$
I_6	$E \rightarrow E + \cdot T$		\$	\$
I_7	$T \rightarrow T F \cdot$		$+/a/b/\$$	$+/a/b/\$$
I_7	$F \rightarrow F \cdot *$	*	$*/+/a/b/\$$	$*/+/a/b/\$$
I_8	$F \rightarrow F * \cdot$		$+/a/b/*$	$+/a/b/*/\$$
I_9	$E \rightarrow E + T \cdot$		\$	\$
I_9	$T \rightarrow T \cdot F$	a/b	$a/b/\$$	$a/b/\$$

LALR 分析表:

	action					goto		
	a	b	*	+	\$	E	T	F
0	s4	s5				1	2	3
1				s6	acc			
2	s4	s5		r2				7
3	r4	r4	s8	r4	r4			
4	r6	r6	r6	r6	r6			
5	r7	r7	r7	r7	r7			
6	s4	s5					9	3
7	r3	r3	s8	r3	r3			
8	r5	r5	r5	r5	r5			
9	s4	s5			r1			7

(Aho) 4.37 对文法

$S \rightarrow AaAb \mid BbBa$

$A \rightarrow \epsilon$

$B \rightarrow \epsilon$

a) 证明它是 LL(1)文法但不是 SLR(1)文法

解:

构造预测分析表:

$FIRST(S) = \{a, b\}$, $FIRST(A) = FIRST(B) = \{\epsilon\}$

$FOLLOW(S) = \{\$, \}$, $FOLLOW(A) = FOLLOW(B) = \{a, b\}$

	a	b	\$
S	$S \rightarrow AaAb$	$S \rightarrow BbBa$	
A	$A \rightarrow \epsilon$	$A \rightarrow \epsilon$	
B	$B \rightarrow \epsilon$	$B \rightarrow \epsilon$	

所以, 文法是 LL(1)文法

构造 SLR 分析表:

LR(0)项目集 $I_0 = \{S' \rightarrow \cdot S, S \rightarrow \cdot AaAb, S \rightarrow \cdot BbBa, A \rightarrow \cdot, B \rightarrow \cdot\}$

而 $FOLLOW(A) = FOLLOW(B) = \{a, b\}$, 因此 $action[0, a] = action[0, b] = r3/r4$, 冲突!

所有, 文法不是 SLR(1)文法

(Aho) 4.39 证明文法

$S \rightarrow Aa \mid bAc \mid dc \mid bda$

$A \rightarrow d$

是 LALR(1)文法但不是 SLR(1)文法

解:

构造 SLR 分析表:

项目集 $I_4 = \{S \rightarrow d \cdot c, A \rightarrow d \cdot\}$, 而 $FOLLOW(A) = \{a, c\}$, 因此 $action[4, c] = s8/r5$, 冲突!

$I_7 = \{S \rightarrow bd \cdot a, A \rightarrow d \cdot\}$, 因此 $action[7, a] = s10/r5$, 冲突!

这是 SLR 分析表仅有的两个冲突的地方

因此文法不是 SLR(1)文法

高效构造 LALR 分析表:

通过计算搜索符, 所有项目都具有搜索符\$, $I_4: A \rightarrow d \cdot$ 具有搜索符 a, $I_7: A \rightarrow d \cdot$ 具有搜索符 c, 因此 $action[4, c]=s8$, $action[7, a]=s10$, LALR 分析表不冲突
因此文法是 LALR(1)文法

(Aho) 4.40 证明文法

$S \rightarrow Aa \mid bAc \mid Bc \mid bBa$

$A \rightarrow d$

$B \rightarrow d$

是 LR(1)文法但不是 LALR(1)文法

解:

构造规范 LR 分析表:

$I_0 = \{ [S' \rightarrow \cdot S, \$], [S \rightarrow \cdot Aa, \$], [S \rightarrow \cdot bAc, \$], [S \rightarrow \cdot Bc, \$], [S \rightarrow \cdot bBa, \$], [A \rightarrow \cdot d, a], [B \rightarrow \cdot d, c] \}$

$goto(I_0, S) = \{ [S' \rightarrow S \cdot, \$] \} = I_1$

$goto(I_0, A) = \{ [S \rightarrow A \cdot a, \$] \} = I_2$

$goto(I_0, B) = \{ [S \rightarrow B \cdot c, \$] \} = I_3$

$goto(I_0, b) = \{ [S \rightarrow b \cdot Ac, \$], [S \rightarrow b \cdot Ba, \$], [A \rightarrow \cdot d, c], [B \rightarrow \cdot d, a] \} = I_4$

$goto(I_0, d) = \{ [A \rightarrow d \cdot, a], [B \rightarrow d \cdot, c] \} = I_5$

$goto(I_2, a) = \{ [S \rightarrow Aa \cdot, \$] \} = I_6$

$goto(I_3, c) = \{ [S \rightarrow Bc \cdot, \$] \} = I_7$

$goto(I_4, A) = \{ [S \rightarrow bA \cdot c, \$] \} = I_8$

$goto(I_4, B) = \{ [S \rightarrow bB \cdot a, \$] \} = I_9$

$goto(I_4, d) = \{ [A \rightarrow d \cdot, c], [B \rightarrow d \cdot, a] \} = I_{10}$

$goto(I_8, c) = \{ [S \rightarrow bAc \cdot, \$] \} = I_{11}$

$goto(I_9, a) = \{ [S \rightarrow bBa \cdot, \$] \} = I_{12}$

规范 LR 分析表为:

	action					goto		
	a	b	c	d	\$	S	A	B
0		s4		s5		1	2	3
1					acc			
2	s6							
3			s7					
4				10			8	9
5	r5		r6					
6					r1			
7					r3			
8			s11					
9	s12							
10	r6		r5					
11					r2			
12					r4			

构造 LALR 分析表, 同心集 I_5 和 I_{10} 合并, 显然会造成归约/归约冲突。

因此，文法是 LR(1)文法，但不是 LALR(1)文法

(Aho) 4.42 试编写一个算法，为文法中每个 NT A 计算集合 $\{B \mid A \xRightarrow{*} B\alpha, \text{ 其中 } B \text{ 是 NT, } \alpha \text{ 文法符号串}\}$

解：算法描述如下

1. 设置一个栈，保存 NT，初始栈中只有 A，将结果集合设置为空
2. 若栈空，算法结束，否则执行以下步骤
3. 将栈顶 NT B 弹出，将 B 加入结果集合
4. 对所有形如 $B \rightarrow C\alpha$ 的产生式，将 C 压栈
5. 返回 2

(Aho) 4.44 为练习 4.4 文法构造 SLR 分析表，解决冲突

解：拓广文法

$R' \rightarrow R$

$R \rightarrow R \mid 'R \mid RR \mid R^* \mid (R) \mid a \mid b$

$I_0 = \{ R' \rightarrow \cdot R, R \rightarrow \cdot R \mid 'R, R \rightarrow \cdot RR, R \rightarrow \cdot R^*, R \rightarrow \cdot (R), R \rightarrow \cdot a, R \rightarrow \cdot b \}$

$\text{goto}(I_0, R) = \{ R' \rightarrow R \cdot, R \rightarrow R \cdot \mid 'R, R \rightarrow R \cdot R, R \rightarrow R \cdot ^*, R \rightarrow R \cdot (R), R \rightarrow \cdot a, R \rightarrow \cdot b \} = I_1$

$\text{goto}(I_0, () = \{ R \rightarrow (\cdot R), R \rightarrow R \cdot \mid 'R, R \rightarrow \cdot RR, R \rightarrow \cdot R^*, R \rightarrow \cdot (R), R \rightarrow \cdot a, R \rightarrow \cdot b \} = I_2$

$\text{goto}(I_0, a) = \{ R \rightarrow a \cdot \} = I_3$

$\text{goto}(I_0, b) = \{ R \rightarrow b \cdot \} = I_4$

$\text{goto}(I_1, R) = \{ R \rightarrow RR \cdot, R \rightarrow R \cdot \mid 'R, R \rightarrow R \cdot R, R \rightarrow R \cdot ^*, R \rightarrow R \cdot (R), R \rightarrow \cdot a, R \rightarrow \cdot b \} = I_5$

$\text{goto}(I_1, () = \{ R \rightarrow R (\cdot R), R \rightarrow R R \cdot \mid 'R, R \rightarrow \cdot RR, R \rightarrow \cdot R^*, R \rightarrow \cdot (R), R \rightarrow \cdot a, R \rightarrow \cdot b \} = I_6$

$\text{goto}(I_1, *) = \{ R \rightarrow R^* \cdot \} = I_7$

$\text{goto}(I_1, () = I_2$

$\text{goto}(I_1, a) = I_3$

$\text{goto}(I_1, b) = I_4$

$\text{goto}(I_2, R) = \{ R \rightarrow (R \cdot), R \rightarrow R \cdot \mid 'R, R \rightarrow R \cdot R, R \rightarrow R \cdot ^*, R \rightarrow R \cdot (R), R \rightarrow \cdot a, R \rightarrow \cdot b \} = I_8$

$\text{goto}(I_2, () = I_2$

$\text{goto}(I_2, a) = I_3$

$\text{goto}(I_2, b) = I_4$

$\text{goto}(I_5, R) = I_5$

$\text{goto}(I_5, () = I_6$

$\text{goto}(I_5, *) = I_7$

$\text{goto}(I_5, () = I_2$

$\text{goto}(I_5, a) = I_3$

$\text{goto}(I_5, b) = I_4$

$\text{goto}(I_6, R) = \{ R \rightarrow R \mid 'R, R \rightarrow R \cdot \mid 'R, R \rightarrow R \cdot R, R \rightarrow R \cdot ^*, R \rightarrow R \cdot (R), R \rightarrow \cdot a, R \rightarrow \cdot b \} = I_9$

$\text{goto}(I_6, () = I_2$

$\text{goto}(I_6, a) = I_3$

goto(I₆, b) = I₄
 goto(I₈, R) = I₅
 goto(I₈, |) = I₆
 goto(I₈, *) = I₇
 goto(I₈, () = I₂
 goto(I₈,)) = { R→(R) · } = I₁₀
 goto(I₈, a) = I₃
 goto(I₈, b) = I₄
 goto(I₉, R) = I₅
 goto(I₉, |) = I₆
 goto(I₉, *) = I₇
 goto(I₉, () = I₂
 goto(I₉, a) = I₃
 goto(I₉, b) = I₄

FOLLOW(R)={ |, *, (,), a, b, \$}

SLR 分析表为：

	action							goto
		*	()	a	b	\$	R
0			s2		s3	s4		1
1	s6	s7	s2		s3	s4	acc	5
2			s2		s3	s4		8
3	r5	r5	r5	r5	r5	r5	r5	
4	r6	r6	r6	r6	r6	r6	r5	
5	s6/r2	s7/r2	s2/r2	r2	s3/r2	s4/r2	r2	5
6			s2		s3	s4		9
7	r3	r3	r3	r3	r3	r3	r3	
8	s6	s7	s2	s10	s3	s4		5
9	s6/r1	s7/r1	s2/r1	r1	s3/r1	s4/r1	r1	5
10	r4	r4	r4	r4	r4	r4	r4	

对于状态 5，几种冲突分别表示以下含义（用·表示连接操作）：

...R·R|..., ...R·R*..., ...R·R· (... , ...R·R· a..., ...R·R· a...

显然，除了第二种情况，均应选择归约操作

对于状态 9，冲突含义为：

...R|R|..., ...R|R*..., ...R|R· (... , ...R|R· a..., ...R|R· a...

除了第一种情况，均应选择移进操作

（Aho）4.46 a) 为下面文法构造 SLR 分析表，解决冲突，使得与图 4-52 同样方式分析

E→E sub R | E sup E | { E } | c

R→E sup E | E

解： 拓广

E'→E

构造 LR(0)项目集规范族

$I_0 = \{ E' \rightarrow \cdot E, E \rightarrow \cdot E \text{ sub } R, E \rightarrow \cdot E \text{ sup } E, E \rightarrow \cdot \{ E \}, E \rightarrow \cdot c \}$
 $\text{goto}(I_0, E) = \{ E' \rightarrow E \cdot, E \rightarrow E \cdot \text{sub } R, E \rightarrow E \cdot \text{sup } E \} = I_1$
 $\text{goto}(I_0, \{ \}) = \{ E \rightarrow \{ \cdot E \}, E \rightarrow \cdot E \text{ sub } R, E \rightarrow \cdot E \text{ sup } E, E \rightarrow \cdot \{ E \}, E \rightarrow \cdot c \} = I_2$
 $\text{goto}(I_0, c) = \{ E \rightarrow c \cdot \} = I_3$
 $\text{goto}(I_1, \text{sub}) = \{ E \rightarrow E \text{ sub } \cdot R, R \rightarrow \cdot E \text{ sup } E, R \rightarrow \cdot E, E \rightarrow \cdot E \text{ sub } R, E \rightarrow \cdot E \text{ sup } E, E \rightarrow \cdot \{ E \}, E \rightarrow \cdot c \} = I_4$
 $\text{goto}(I_1, \text{sup}) = \{ E \rightarrow E \text{ sup } \cdot E, E \rightarrow \cdot E \text{ sub } R, E \rightarrow \cdot E \text{ sup } E, E \rightarrow \cdot \{ E \}, E \rightarrow \cdot c \} = I_5$
 $\text{goto}(I_2, E) = \{ E \rightarrow \{ E \cdot \}, E \rightarrow E \cdot \text{sub } R, E \rightarrow E \cdot \text{sup } E \} = I_6$
 $\text{goto}(I_2, \{ \}) = I_2$
 $\text{goto}(I_2, c) = I_3$
 $\text{goto}(I_4, E) = \{ R \rightarrow E \cdot \text{sup } E, R \rightarrow E \cdot, E \rightarrow E \cdot \text{sub } R, E \rightarrow E \cdot \text{sup } E \} = I_7$
 $\text{goto}(I_4, R) = \{ E \rightarrow E \text{ sub } R \cdot \} = I_8$
 $\text{goto}(I_4, \{ \}) = I_2$
 $\text{goto}(I_4, c) = \{ E \rightarrow c \cdot \} = I_3$
 $\text{goto}(I_5, E) = \{ E \rightarrow E \text{ sup } E \cdot, E \rightarrow E \cdot \text{sub } R, E \rightarrow E \cdot \text{sup } E \} = I_9$
 $\text{goto}(I_5, \{ \}) = I_2$
 $\text{goto}(I_5, c) = I_3$
 $\text{goto}(I_6, \{ \}) = \{ E \rightarrow \{ E \cdot \} \} = I_{10}$
 $\text{goto}(I_6, \text{sub}) = I_4$
 $\text{goto}(I_6, \text{sup}) = I_5$
 $\text{goto}(I_7, \text{sub}) = I_4$
 $\text{goto}(I_7, \text{sup}) = \{ R \rightarrow E \text{ sup } \cdot E, E \rightarrow E \text{ sup } \cdot E, E \rightarrow \cdot E \text{ sub } R, E \rightarrow \cdot E \text{ sup } E, E \rightarrow \cdot \{ E \}, E \rightarrow \cdot c \} = I_{11}$
 $\text{goto}(I_9, \text{sub}) = I_4$
 $\text{goto}(I_9, \text{sup}) = I_5$
 $\text{goto}(I_{11}, E) = \{ R \rightarrow E \text{ sup } E \cdot, E \rightarrow E \text{ sup } E \cdot, E \rightarrow E \cdot \text{sub } R, E \rightarrow E \cdot \text{sup } E \} = I_{12}$
 $\text{goto}(I_{11}, \{ \}) = I_2$
 $\text{goto}(I_{11}, c) = I_3$
 $\text{goto}(I_{12}, \text{sub}) = I_4$
 $\text{goto}(I_{13}, \text{sup}) = I_5$

$\text{FOLLOW}(E) = \text{FOLLOW}(R) = \{ \text{sub}, \text{sup}, \{ \}, \$ \}$

SLR 分析表为:

	action						goto	
	sub	sup	{	}	c	\$	E	R
0			s2		s3		1	
1	s4	s5				acc		
2			s2		s3		6	
3	r4	r4		r4		r4		
4			s2		s3		7	8
5			s2		s3		9	
6	s4	s5		s10				
7	s4/r6	s11/r6		r6		r6		
8	r1	r1		r1		r1		
9	s4/r2	s5/r2		r2		r2		

10	r3	r3		r3		r3		
11			s2		s3		12	
12	s4/r5/r2	s5/r5/r2		r5/r2		r5/r2		

(Aho) 4.48 考虑下面 n 个二元中缀操作符的二义性文法:

$E \rightarrow E \theta_1 E \mid E \theta_2 E \mid \dots \mid E \theta_n E \mid (E) \mid id$

假设所有操作符都是左结合的, 而且若 $i > j$, 则 θ_i 的优先级高于 θ_j 。

a) 试为该文法构造 SLR 项目集。共有多少个项目集? 是 n 的函数吗?

解:

$I_0 = \{ E' \rightarrow \cdot E, E \rightarrow \cdot E \theta_i E, E \rightarrow \cdot (E), E \rightarrow \cdot id \} \quad (1 \leq i \leq n)$
 $goto(I_0, E) = \{ E' \rightarrow E \cdot, E \rightarrow E \cdot \theta_i E \} = I_1 \quad (1 \leq i \leq n)$
 $goto(I_0, () = \{ E \rightarrow (\cdot E), E \rightarrow \cdot E \theta_i E, E \rightarrow \cdot (E), E \rightarrow \cdot id \} = I_2 \quad (1 \leq i \leq n)$
 $goto(I_0, id) = \{ E \rightarrow id \cdot \} = I_3$
 $goto(I_1, \theta_i) = \{ E \rightarrow E \theta_i \cdot E, E \rightarrow \cdot E \theta_j E, E \rightarrow \cdot (E), E \rightarrow \cdot id \} = I_{3+i} \quad (1 \leq i, j \leq n)$
 $goto(I_2, E) = \{ E \rightarrow (E \cdot), E \rightarrow E \cdot \theta_i E \} = I_{n+4} \quad (1 \leq i \leq n)$
 $goto(I_2, () = I_2$
 $goto(I_2, id) = I_3$
 $goto(I_{3+i}, E) = \{ E \rightarrow E \theta_i E \cdot, E \rightarrow E \cdot \theta_j E \} = I_{n+4+i} \quad (1 \leq i, j \leq n)$
 $goto(I_{3+i}, () = I_2$
 $goto(I_{3+i}, id) = I_3$
 $goto(I_{n+4}, \theta_i) = I_{3+i} \quad (1 \leq i \leq n)$
 $goto(I_{n+4},)) = \{ E \rightarrow (E) \cdot \} = I_{2n+5}$
 $goto(I_{n+4+i}, \theta_j) = I_{3+j} \quad (1 \leq j \leq n)$
 共 $2n+5$ 个项目集

b) 试为该文法构造 SLR 语法分析表, 并用 4.7 节方法进行压缩。压缩后总长度是多少? 是 n 的函数吗?

解:

$FOLLOW(E) = \{ \theta_i,), \$ \}$

SLR 分析表为

$action(0, () = s2, \quad action(0, id) = s3, \quad goto(0, E) = 1$
 $action(1, \theta_i) = si+3, \quad action(1, \$) = acc \quad (1 \leq i \leq n)$
 $action(2, () = s2, \quad action(2, id) = s3, \quad goto(2, E) = n+4$
 $action(3, \theta_i) = action(3,)) = action(3, \$) = rn+2$
 $action(i+3, () = s2, \quad action(i+3, id) = s3, \quad goto(i+3, E) = n+i+4 \quad (1 \leq i \leq n)$
 $action(n+4, \theta_i) = si+3, \quad action(n+4,)) = s2n+5 \quad (1 \leq i \leq n)$
 $action(n+i+4, \theta_j) = sj+3 \quad (1 \leq i < j \leq n), \quad action(n+i+4, \theta_j) = ri \quad (1 \leq j \leq i \leq n)$
 $action(n+i+4,)) = action(n+i+4, \$) = ri$
 $action(2n+5, \theta_i) = action(2n+5,)) = action(2n+5, \$) = rn+1 \quad (1 \leq i \leq n)$

压缩 aciton 表:

状态 0、2、4— $n+3$ 的行完全相同, 用三个二元组表示:

$((, s2), (id, s3), (any, error))$

状态 1:

$(\theta_1, s_4), (\theta_2, s_5), \dots, (\theta_n, s_{n+3}), (\$, acc), (any, error)$

状态 3, 用归约动作替换 error:

$(any, rn+2)$

状态 $n+4$:

$(\theta_1, s_4), (\theta_2, s_5), \dots, (\theta_n, s_{n+3}), (), s_{2n+5}), (any, error)$

状态 $n+5$:

$(\theta_2, s_5), \dots, (\theta_n, s_{n+3}), (any, r_1)$

状态 $n+6$:

$(\theta_3, s_6), \dots, (\theta_n, s_{n+3}), (any, r_2)$

...

状态 $2n+4$:

(any, rn)

状态 $2n+5$:

$(any, rn+1)$

压缩 goto 表:

$(2, n+4), (4, n+5), (5, n+6), \dots, (n+3, 2n+4), (any, 1)$

长度与 n 有关

c) 分析 $id \theta_i id \theta_j id$ 需要多少步?

解: 若 $i \geq j$, 分析过程如下

id 移进, id 归约为 E , θ_i 移进, id 移进, id 归约为 E , $E \theta_i E$ 归约为 E , θ_j 移进, id 移进, id 归约为 E , $E \theta_j E$ 归约为 E , 共 10 步

若 $i < j$, 分析过程为:

id 移进, id 归约为 E , θ_i 移进, id 移进, id 归约为 E , θ_j 移进, id 移进, id 归约为 E , $E \theta_j E$ 归约为 E , $E \theta_i E$ 归约为 E , 也是 10 步

(Aho) 4.49 对下面的无二义性文法重做练习 4.48

$E_1 \rightarrow E_1 \theta_1 E_2 \mid E_2$

$E_2 \rightarrow E_2 \theta_2 E_3 \mid E_3$

...

$E_n \rightarrow E_n \theta_n E_{n+1} \mid E_{n+1}$

$E_{n+1} \rightarrow (E_1) \mid id$

解: 构造 LR(0)项目集规范族, 拓广文法, $E_0 \rightarrow E_1$

$I_0 = \{ E_0 \rightarrow \cdot E_1, E_i \rightarrow \cdot E_i \theta_i E_{i+1}, E_i \rightarrow \cdot E_{i+1}, E_{n+1} \rightarrow \cdot (E_1), E_{n+1} \rightarrow \cdot id \} \quad (1 \leq i \leq n)$

$goto(I_0, E_i) = \{ E_{i-1} \rightarrow E_i \cdot, E_i \rightarrow E_i \cdot \theta_i E_{i+1} \} = I_i \quad (1 \leq i \leq n)$

$goto(I_0, E_{n+1}) = \{ E_n \rightarrow E_{n+1} \cdot \} = I_{n+1}$

$goto(I_0, ()) = \{ E_{n+1} \rightarrow (\cdot E_1), E_i \rightarrow \cdot E_i \theta_i E_{i+1}, E_i \rightarrow \cdot E_{i+1}, E_{n+1} \rightarrow \cdot (E_1), E_{n+1} \rightarrow \cdot id \} = I_{n+2} \quad (1 \leq i \leq n)$

$goto(I_0, id) = \{ E_{n+1} \rightarrow id \cdot \} = I_{n+3}$

$goto(I_i, \theta_i) = \{ E_i \rightarrow E_i \theta_i \cdot E_{i+1}, E_j \rightarrow \cdot E_j \theta_j E_{j+1}, E_j \rightarrow \cdot E_{j+1}, E_{n+1} \rightarrow \cdot (E_1), E_{n+1} \rightarrow \cdot id \} = I_{n+i+3} \quad (1 \leq i \leq n, i < j \leq n)$

$\text{goto}(I_{n+2}, E_1) = \{ E_{n+1} \rightarrow (E_1 \cdot), E_1 \rightarrow E_1 \cdot \theta_1 E_2 \} = I_{2n+4}$
 $\text{goto}(I_{n+2}, E_i) = I_i \quad (2 \leq i \leq n+1)$
 $\text{goto}(I_{n+2}, ()) = I_{n+2}$
 $\text{goto}(I_{n+2}, \text{id}) = I_{n+3}$
 $\text{goto}(I_{n+i+3}, E_{i+1}) = \{ E_i \rightarrow E_i \theta_i E_{i+1} \cdot, E_{i+1} \rightarrow E_{i+1} \cdot \theta_{i+1} E_{i+2} \} = I_{2n+i+4} \quad (1 \leq i < n)$
 $\text{goto}(I_{2n+3}, E_{n+1}) = \{ E_n \rightarrow E_n \theta_n E_{n+1} \cdot \} = I_{3n+4}$
 $\text{goto}(I_{n+i+3}, E_j) = I_j \quad (2 \leq i+1 < j \leq n+1)$
 $\text{goto}(I_{n+i+3}, ()) = I_{n+2} \quad (1 \leq i \leq n)$
 $\text{goto}(I_{n+i+3}, \text{id}) = I_{n+3} \quad (1 \leq i \leq n)$
 $\text{goto}(I_{2n+4}, ()) = \{ E_{n+1} \rightarrow (E_1) \cdot \} = I_{3n+5}$
 $\text{goto}(I_{2n+i+4}, \theta_{i+1}) = I_{n+i+4} \quad (0 \leq i < n)$

$\text{FOLLOW}(E_i) = \{ \cdot, \$, \theta_j \} \quad (1 \leq i \leq n+1, j \leq \min(i, n))$

构造 SLR 分析表

$\text{action}(0, ()) = \text{sn}+2, \quad \text{action}(0, \text{id}) = \text{sn}+3, \quad \text{goto}(0, E_i) = \text{si} \quad (1 \leq i \leq n+1)$
 $\text{action}(1, \$) = \text{acc}$
 $\text{action}(i+1, ()) = \text{action}(i+1, \$) = \text{action}(i+1, \theta_j) = r2*i \quad (1 \leq i \leq n, j \leq i)$
 $\text{action}(i, \theta_i) = \text{sn}+i+3 \quad (1 \leq i \leq n)$
 $\text{action}(n+2, ()) = \text{sn}+2, \quad \text{action}(n+2, \text{id}) = \text{sn}+3, \quad \text{goto}(n+2, E_1) = 2n+4, \quad \text{goto}(n+2, E_i) = i \quad (2 \leq i \leq n+1)$
 $\text{action}(n+3, ()) = \text{action}(n+3, \$) = \text{action}(n+3, \theta_j) = r2*(n+1) \quad (1 \leq j \leq n)$
 $\text{action}(n+i+3, ()) = \text{sn}+2, \quad \text{action}(n+i+3, \text{id}) = \text{sn}+3, \quad \text{goto}(I_{n+i+3}, E_{i+1}) = 2n+i+4, \quad (1 \leq i \leq n)$
 $\text{goto}(n+i+3, E_j) = j \quad (2 \leq i+1 < j \leq n)$
 $\text{action}(2n+4, ()) = s3n+5$
 $\text{action}(2n+i+4, \theta_{i+1}) = \text{sn}+i+4 \quad (0 \leq i < n)$
 $\text{action}(2n+i+4, ()) = \text{action}(2n+i+4, \$) = \text{action}(2n+i+4, \theta_j) = r2i-1 \quad (1 \leq i \leq n, j \leq i)$
 $\text{action}(3n+5, ()) = \text{action}(3n+5, \$) = \text{action}(3n+5, \theta_j) = r2n+1 \quad (1 \leq j \leq n)$

分析 $\text{id } \theta_i \text{id } \theta_j \text{id}$ 的过程中，显然 id 需按 $E_{n+1} \rightarrow E_n \rightarrow \dots \rightarrow E_i$ 的顺序进行归约

很明显，无论是形成的语法分析器的效率还是构造语法分析器的效率，都是二义性文法要更好。

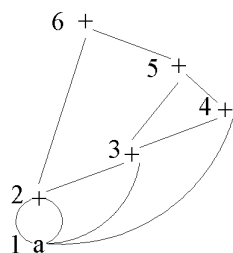
第五章部分习题解答

Aho:《编译原理技术与工具》书中习题

(Aho) 5.3 为下面表达式构造有向无环图, 标出结点(子表达式)编号, +是左结合的

$a + a + (a + a + a + (a + a + a + a))$

解:



(Aho) 5.5 设计语法制导定义, 实现多项式(包含+和*, 如 $x*(3*x+x*x)$)的求导, 结果无需化简, 如 $3*x$ 直接翻译为 $3*1+0*x$ 即可。(思路: $(xy)'=x'y+y'x$, $(x+y)'=x'+y'$)

解:

综合属性 diff 为求导后的表示形式, val 为原多项式表示形式

```
E → E1 + T      { E.val = E1.val || “+” || E.val;      E.diff = E1.diff || “+” || E.diff; }
    | T             { E.val = T.val;      E.diff = T.diff; }
T → T1 * F      { T.val = T1.val || “*” || F.val;
                  T.diff = “(” || T1.diff || “*” || F.val || “+” || T1.val || “*” || F.diff || “)” ; }
    | F             { T.val = F.val;      T.diff = F.diff; }
F → ( E )         { F.val = “(” || E.val || “)” ;      F.diff = “(” || E.diff || “)” ; }
    | num          { F.val = tostr(num.val);  F.diff = “0” ; }
    | x            { F.val = “x” ;      F.diff = “1” ; }
```

此题和后面两题均可编写 Lex&Yacc 程序进行验证。

(Aho) 5.4 设计翻译模式, 实现将中缀表达式翻译为无多余括号形式

解: 所谓“多余括号”, 可以理解为:

本该是表达式 E, 写成了(E)的形式,

本该是项 T, 写成了(T)的形式,

本该是因式 F, 写成了(F)的形式, 因此, 可写出文法和翻译模式:

```
E → ( E1 ) { E.s = E1.s; }
    | E1 + T { E.s = E1.s || “+” || T.s; }
    | T      { E.s = T.s; }
```

```
T → ( T1 ) { T.s = T1.s; }
    | T1 + F { T.s = T1.s || “+” || F.s; }
    | F      { T.s = F.s; }
```

```

F → ( F1 ) { F.s = F1.s; }
    | ( E ) { F.s = '(' || E.s || ')'; }
    | num { F.s = num.lexeme; }
    | id { F.s = id.lexeme; }

```

显然，这是一个二义性文法，存在移进—归约冲突和归约—归约冲突，但通过分析语法含义，可解决冲突。

我们将 $E \rightarrow (E_1)$ 、 $T \rightarrow (T_1)$ 、 $F \rightarrow (F_1)$ 称为“多余括号产生式”。显然，“多余括号表达式”存在多个语法树，可通过“多余括号产生式”推导而得，也可仅通过“正常产生式”推导而来，这也是二义性的根源。而“无多余括号表达式”是不可能通过“多余括号产生式”推导而来的。因此，对于冲突的情况，我们可以优先选择“多余括号产生式”，确定移进/归约。在 Parser Generator 2 工具中，移进/归约冲突总是选择移进，归约/归约冲突选择顺序靠前的产生式。如上的产生式顺序即可保证每个冲突情况都选择“多余括号产生式”。

(Aho) 5.8 令综合属性 val 给出下列文法 S 产生的二进制数的值，例如，输入 101.101 时，
S.val=5.625

```

S → L , L | L
L → L B | B
B → 0 | 1

```

设计语法制导定义，实现对 S.val 的求值，B 的综合属性只允许有一个：c，表示 B 产生的位对最终值的贡献。

```

S → L1 , L2 { S.val = L1.val + L2.val / L2.f; }
    | L { S.val = L.val; }
L → L1 B { L.val = L1.val * 2 + B.c; L.f = L1.f * 2; }
    | B { L.val = B.c; L.f = 2; }
B → 0 { B.c = 0; }
    | 1 { B.c = 1; }

```

(Aho) 5.9 重写例 5.3 中语法制导定义的基础文法，使得类型信息只需用综合属性来传播解：改写后文法和语法制导定义为

```

L → T id      { L.type = T.type; addtype(id.entry, L.type); }
L → L1, id    { L.type = L1.type; addtype(id.entry, L.type); }
T → int       { T.type = integer; }
T → real      { T.type = real; }

```

(Aho) 5.10 当下列文法产生的语句翻译为抽象机器代码时，break 语句翻译为一个跳转指令，它跳转到最近的 while 循环。终结符 expr 表示表达式，other 表示其他类型语句，这两个终结符都有综合属性 code，表示翻译后的代码。

```

S → while expr do begin S end
    | S ; S
    | break

```

| other

试给出一个语法制导定义，将语句翻译为抽象堆栈机代码，保证 break 的正确翻译

解：语法制导定义如下，其中 begin 和 next 为继承属性，分别表示语句的开始和结束（下一条语句开始）

```
S → while expr do begin S1 end { S.begin = newlabel();  
                                   S1.next = S.begin;  
                                   S.code = 'label' S.begin || expr.code || 'gotofalse' S.next ||  
                                   S1.code || 'goto' S.begin || 'label' S.next; }  
  | S1 ; S2 { S1.begin = S.begin; S1.next = S2.begin = newlabel();  
                                   S2.next = S.next; S.code = S1.code || S2.code; }  
  | break { S.code = 'goto' S.next; }  
  | other { S.code = other.code; }
```

(Aho) 5.12 下列文法产生包含赋值操作的表达式

S → E

E → E := E | E + E | (E) | id

a) 试构造一个语法制导定义，检查赋值表达式的左部是否为左值。用 E 的继承属性 side 来指明 E 应该出现在赋值号左部还是右部

解：语法制导定义如下

```
S → E { E.side = right; }  
E → E1 := E2 { if (E.side == left) error;  
                  else { E1.side = left; E2.side = right; } }  
E → E1 + E2 { if (E.side == left) error;  
                  else { E1.side = E2.side = right; } }  
E → ( E1 ) { if (E.side == left) error;  
               else { E1.side = right; } }
```

E → id

b) 扩充 a) 中语法制导定义，使得在检查同时将表达式翻译为抽象堆栈机代码

解：语法制导定义如下，其中，valcode 表示取表达式右值的代码

```
S → E { E.side = right; S.code = E.code; }  
E → E1 := E2 { if (E.side == left) error;  
                  else { E1.side = left; E2.side = right; }  
                  E.code = E1.code || E2.code || ':' || E1.valcode; }  
E → E1 + E2 { if (E.side == left) error;  
                  else { E1.side = E2.side = right; }  
                  E.code = E1.code || E2.code || '+'; }  
E → ( E1 ) { if (E.side == left) error;  
               else { E1.side = right; }  
               E.code = E1.code; }  
E → id { E.code = 'lvalue' id.lexeme; E.valcode = 'rvalue' id.lexeme; }
```

(Aho) 5.16 下面文法是书中图 5-22 文法的无二义性版本，其中{}只是用来为盒子分组，在翻译过程中无实际意义。

$S \rightarrow L$
 $L \rightarrow L B \mid B$
 $B \rightarrow B \text{ sub } F \mid F$
 $F \rightarrow \{ L \} \mid \text{text}$

a) 试改写图 5-22 的语法制导定义，以适用于上面改写后的文法

解：语法制导定义如下，其中，ps 为继承属性，ht 为综合属性

$S \rightarrow L \quad \{ L.ps = 10; S.ht = L.ht; \}$
 $L \rightarrow L_1 B \quad \{ L_1.ps = L.ps; B.ps = L.ps; L.ht = \max(L_1.ht, B.ht); \}$
 $L \rightarrow B \quad \{ B.ps = L.ps; L.ht = B.ht; \}$
 $B \rightarrow B_1 \text{ sub } F \quad \{ B_1.ps = B.ps; F.ps = \text{shrink}(B.ps); B.ht = \text{disp}(B_1.ht, F.ht); \}$
 $B \rightarrow F \quad \{ F.ps = B.ps; B.ht = F.ht; \}$
 $F \rightarrow \{ L \} \quad \{ L.ps = F.ps; F.ht = L.ht; \}$
 $F \rightarrow \text{text} \quad \{ F.ht = \text{text.h} * F.ps; \}$

b) 将 a) 的结果转换为翻译模式

解：翻译模式如下

$S \rightarrow \quad \{ L.ps = 10; \}$
 $\quad L \quad \{ S.ht = L.ht; \}$
 $L \rightarrow \quad \{ L_1.ps = L.ps; \}$
 $\quad L_1 \quad \{ B.ps = L.ps; \}$
 $\quad B \quad \{ L.ht = \max(L_1.ht, B.ht); \}$
 $L \rightarrow \quad \{ B.ps = L.ps; \}$
 $\quad B \quad \{ L.ht = \max(L_1.ht, B.ht); \}$
 $B \rightarrow \quad \{ B_1.ps = B.ps; \}$
 $\quad B_1 \text{ sub} \quad \{ F.ps = \text{shrink}(B.ps); \}$
 $\quad F \quad \{ B.ht = \text{disp}(B_1.ht, F.ht); \}$
 $B \rightarrow \quad \{ F.ps = B.ps; \}$
 $\quad F \quad \{ B.ht = F.ht; \}$
 $F \rightarrow \quad \{ L.ps = F.ps; \}$
 $\quad \{ L \} \quad \{ F.ht = L.ht; \}$
 $F \rightarrow \text{text} \quad \{ F.ht = \text{text.h} * F.ps; \}$

(Aho) 5.17 扩充 5.5 节中消除左递归的变换方法，使之允许(5-2)中的非终结符 A 具有：

a) 由复制规则定义的继承属性

解：具有复制规则定义的继承属性的左递归文法的翻译模式为

$A \rightarrow \{ A_1.in = A.in; \} A_1 Y \{ A.a = g(A_1.a, Y.y, A.in); \}$
 $A \rightarrow X \{ A.a = f(X.x, A.in); \}$

消除左递归后变为，其中 R.r 为继承属性，R.s 为综合属性

$A \rightarrow X \{ R.r = f(X.x, A.in); R.in = A.in; \} R \{ A.a = R.s; \}$
 $R \rightarrow Y \{ R_1.r = g(R.r, Y.y, R.in); R_1.in = R.in; \} R_1 \{ R.s = R_1.s; \}$

$R \rightarrow \varepsilon \{ R.s = R.r; \}$

b) 继承属性

解：具继承属性的左递归文法的翻译模式为

$A \rightarrow \{ A_1.in = h(A.in); \} A_1 Y \{ A.a = g(A_1.a, Y.y, A.in); \}$

$A \rightarrow X \{ A.a = f(X.x, A.in); \}$

消除左递归后变为，其中 $R.r$ 为继承属性， $R.s$ 为综合属性

$A \rightarrow X \{ R.r = f(X.x, A.in); R.in = h(A.in); \} R \{ A.a = R.s; \}$

$R \rightarrow Y \{ R_1.r = g(R.r, Y.y, R.in); R_1.in = h(R.in); \} R_1 \{ R.s = R_1.s; \}$

$R \rightarrow \varepsilon \{ R.s = R.r; \}$

(Aho) 5.18 将 5.16 (b) 的结果消除左递归

解：消除左递归后如下

$S \rightarrow \{ L.ps = 10; \}$
 $L \{ S.ht = L.ht; \}$
 $L \rightarrow \{ B.ps = L.ps; \}$
 $B \{ L'.in = B.ht; L'.ps = L.ps; \}$
 $L' \{ L.ht = L'.ht; \}$
 $L' \rightarrow \{ B.ps = L.ps; \}$
 $B \{ L'_1.in = \max(L.in, B.ht); L'_1.ps = L'.ps; \}$
 $L'_1 \{ L'.ht = L'_1.ht; \}$
 $L' \rightarrow \varepsilon \{ L'.ht = L'.in; \}$
 $B \rightarrow \{ F.ps = B.ps; \}$
 $F \{ B'.in = F.ht; B'.ps = B.ps; \}$
 $B' \{ B.ht = B'.ht; \}$
 $B' \rightarrow \{ F.ps = \text{shrink}(B.ps); \}$
 $\text{sub } F \{ B'_1.in = \text{disp}(B.in, F.ht); B'_1.ps = F.ps; \}$
 $B'_1 \{ B'.ht = B'_1.ht; \}$
 $B' \rightarrow \varepsilon \{ B'.ht = B'.in; \}$
 $F \rightarrow \{ L.ps = F.ps; \}$
 $\{ L \} \{ F.ht = L.ht; \}$
 $F \rightarrow \text{text} \{ F.ht = \text{text.h} * F.ps; \}$

第六、七章作业

Aho:《编译原理技术与工具》书中习题

(Aho) 6.1 请写出下列类型的类型表达式

a) 指向实数的指针的数组, 数组下标从 1 到 100

解: $\text{array}(1..100, \text{pointer}(\text{real}))$

b) 二维整型数组 (数组的数组), 行下标从 0 到 9, 列下标从 -10 到 10

解: $\text{array}(0..9, \text{array}(-10..10, \text{integer}))$

c) 函数, 其定义域是从整数到整型指针的函数, 值域是整数和字符组成的记录

解: $(\text{integer} \rightarrow \text{pointer}(\text{integer})) \rightarrow \text{record}(\text{integer} \times \text{char})$

(Aho) 7.6 对下面这段程序

```
program main(input, output);
```

```
  procedure p(x, y, z);
```

```
    begin
```

```
      y := y + 1;
```

```
      z := z + x;
```

```
    end;
```

```
  begin
```

```
    a := 2;
```

```
    b := 3;
```

```
    p(a + b, a, a);
```

```
    print a
```

```
  end.
```

采取下列参数传递机制, 最终显示的结果分别是什么?

a) 传值方式

解: 由于采取传值方式, 无论函数 p 如何运行, 形参的变化都不会影响实参 a, 因此最终 a 的值仍未 2, 程序输出为 2

b) 传地址方式

解: 程序运行过程为

调用 p 之前, 实参 $a+b=5$ 、 $a=2$ 、 $a=2$, 由于是传地址方式, 因此 y、z 都与 a 占用相同的内存空间, $a+b$ 没有左值, 因此为它分配一个临时变量空间, x 与它占用同一空间。

$y := y + 1$, $y=a=z=3$

$z := z + x$, $z=a=8$

因此, 最终 a 的值为 8

c) 复制—恢复方式

解: 前一阶段与传值方式相同

调用 p 之前, 实参 $a+b=5$ 、 $a=2$ 、 $a=2$, x、y、z 的值分别为 5、2、2, 与实参占用不同

内存空间。

$y := y + 1, y = 3$

$z := z + x, z = 7$

然后进行形参→实参的复制，最终结果依赖于复制的顺序，假定赋值由左至右，则 a 的最终值与 z 相同，为 7

d) 传名方式

解：可将传名调用看作宏，相当于将调用 p 的代码转换为

$a := a + 1;$

$a := a + a + b;$

因此，最终 a 的值为 9

第八、九章习题解答参考

Aho:《编译原理技术与工具》书中习题

(Aho) 8.3、9.6 将下面 C 程序, 翻译为三地址码, 画出流图, 翻译为汇编代码

```
main()
{
    int i;
    int a[10];
    i = 1;
    while (i <= 10) {
        a[i] = 0; i = i + 1;
    }
}
```

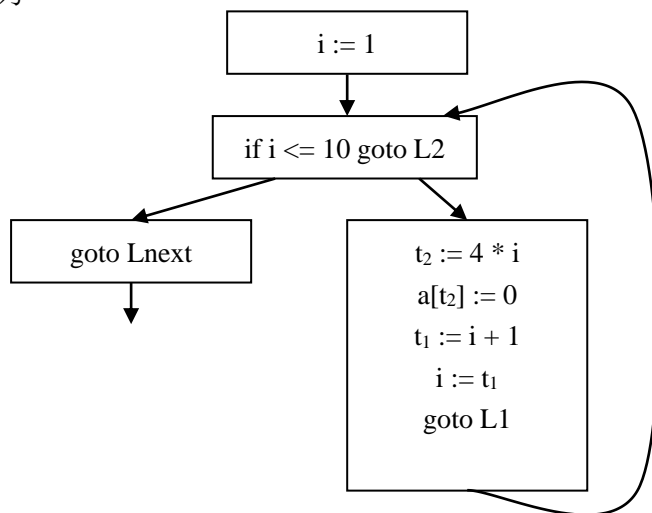
解:

三地址码为

```
i := 1
L1:  if i <= 10 goto L2
    goto Lnext
L2:  t1 := c
    t2 := 4 * i
    t1[t2] := 0
    t1 := i + 1
    i := t1
    goto L1
Lnext:
```

Lnext:

流图为

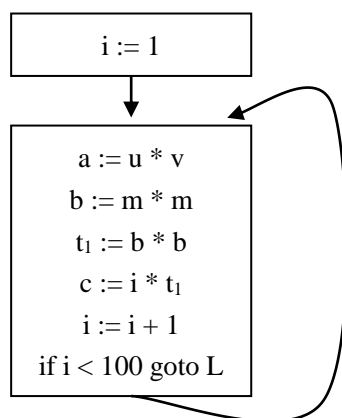


第十章作业解答参考

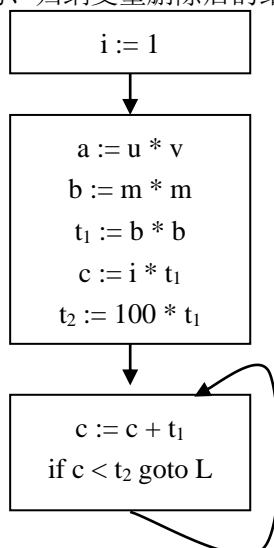
将下面三地址码程序段划分基本块，画出流图，然后进行代码外提、强度削弱、归纳变量删除等优化

```
      i := 1
L:    a := u * v
      b := m * m
      t1 := b * b
      c := i * t1
      i := i + 1;
      if i < 100 goto L
```

解：流图如下



进行代码外提、强度削弱、归纳变量删除后的结果为



汇编码为采取了 9.7 节寄存器优化算法并已进行窥孔优化：

```
      MOV 1, R0
L1:    CMP R0, 10
      J<= L2
      GOTO Lnext
L2:    MOV 4, R1
      MUL R0, R1
      MOV 0, a(R1)
      MOV i, R1
      ADD 1, R1
      MOV R1, R0
      GOTO L1
Lnext:
```