

国家一级假勤奋大学生整理

32. 表达式  $-a+b*(-c+d)$  的逆波兰式是 (D)

- A.  $ab+-cd+-*$     B.  $a-b+c-d+*$     C.  $a-b+cd-+*$     D.  $a-bc-d+*+$

33. 逆波兰式  $ab+cd+*$  所代表的中缀形式的表达式是 (B)。

- A.  $a+b+c*d$   
 B.  $(a+b)*(c+d)$   
 C.  $(a+b)*c+d$   
 D.  $a+b*c+d$

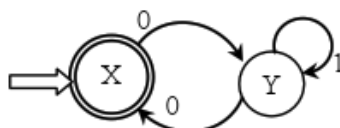
55. 给定语言  $L(G)=\{a^n b b^n \mid n \geq 1\}$ , 则文法 (D) 可产生语言  $L(G)$ 。

- A.  $Z \rightarrow aZb \mid aAb \mid b, A \rightarrow aAb \mid b$   
 B.  $A \rightarrow aAb \mid b$   
 C.  $Z \rightarrow AbB, A \rightarrow aA \mid a, B \rightarrow bB \mid b$   
 D.  $Z \rightarrow aAb, A \rightarrow aAb \mid b$

57. 设文法  $G=(\{S\}, \{0,1\}, P, S)$ , 其中  $P=\{S \rightarrow SS \mid 0S1 \mid 1S0 \mid \epsilon\}$ , 该文法所描述的语言为 (B)。

- A.  $\{0^n 1^n \mid n \geq 0\}$   
 B.  $\{w \mid w \in \{0,1\}^* \text{ 且 } w \text{ 中 } 0 \text{ 和 } 1 \text{ 的个数相等}\}$   
 C.  $\{0^m 1^k \mid m, k \geq 0\} \cup \{1^m 0^k \mid m, k \geq 0\}$   
 D.  $\{0^n 1^n \mid n \geq 0\} \cup \{1^n 0^n \mid n \geq 0\}$

59. 下图所示状态转换图接受的字集为 (D)。



- A. 以 0 开头的二进制数组成的集合    B. 以 0 结尾的二进制数组成的集合  
 C. 含奇数个 0 的二进制数组成的集合    D. 含偶数个 0 的二进制数组成的集合

5. 设文法  $G(S)$  的 BNF 描述为:

$$S \rightarrow S, E \mid E$$

$$E \rightarrow E+T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow a \mid (E) \mid a[S]$$

① 给出  $G(S)$  的元语言符号集、文法符号集、终结符集和非终结符集。

② 判断符号串  $S1:a,a+a[a[S]]$  和  $S2:a*a,a+a[a]$  是否为文法  $G(S)$  的句子, 对是  $L(G(S))$  的句子给出相应的分析树。

解: (1) 元语言符号集:  $\{\rightarrow, \mid\}$

$$V_N = \{S, E, T, F\}$$

$$V_T = \{,, +, *, a, (, ), [, ]\}$$

$$V = V_N \cup V_T = \{S, E, T, F, ,, +, *, a, (, ), [, ]\}$$

(3) 由于  $S1:a,a+a[a[S]]$  串中含有非终结符  $S$ , 所以该符号串不是文法  $G(S)$  的句子。

$S2:a*a,a+a[a]$  是文法  $G(S)$  的句子, 其对应的语法分析树如下 (略)

6. 若有  $\Sigma = \{0, 1\}$ , 求  $\Sigma^* - \Sigma^+ = ?$ ,  $\Sigma^{2*} = ?$

解:  $\Sigma^* - \Sigma^+ = \epsilon$   $\Sigma^{2*} = \{00, 01, 10, 11\}$

### 3、已知文法 $G[S]$ 为:

$$S \rightarrow dAB$$

$$A \rightarrow aA \mid a$$

$$B \rightarrow Bb \mid \epsilon$$

①  $G[S]$  产生的语言是什么?

②  $G[S]$  能否改写为等价的正规文法?

答: ①  $G[S]$  产生的语言是  $L(G[S]) = \{da^n b^m \mid n \geq 1, m \geq 0\}$ 。

②  $G[S]$  能改写为等价的正规文法, 其改写后的等价的

正规文法  $G[S']$  为:

$$S' \rightarrow dA$$

$$A \rightarrow aA \mid aB \mid a$$

$$B \rightarrow bB \mid b$$

一、构造一个文法，使其描述的语言L(G)是 $\{0^n1^n | n \geq 1\}$ 。

G[S]: G:  $S \rightarrow 0S1, S \rightarrow 01$

二、已知文法G[A]:

$A \rightarrow aABl \mid a$

$B \rightarrow Bb \mid d$

给出与G[A]等价的LL(1)文法并构造该文法的预测分析表。

答：把文法变为G'[A]:

$A \rightarrow aA'$

$A' \rightarrow ABl \mid \varepsilon$

$B \rightarrow dB'$

$B' \rightarrow bB' \mid \varepsilon$

此时

$FIRST(A) = \{a\} \quad FOLLOW(A) = \{\#, d\}$

$FIRST(A') = \{a, \varepsilon\} \quad FOLLOW(A') = \{\#, d\}$

$FIRST(B) = \{d\} \quad FOLLOW(B) = \{l\}$

$FIRST(B') = \{b, \varepsilon\} \quad FOLLOW(B') = \{l\}$

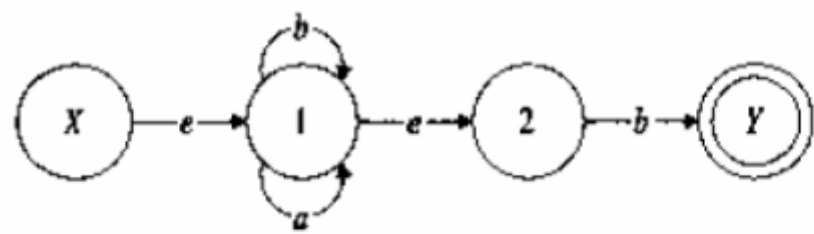
由于

$SELECT(A' \rightarrow ABl) \cap SELECT(A' \rightarrow \varepsilon) = \{a\} \cap \{\#, d\} = \emptyset$

$SELECT(B' \rightarrow bB') \cap SELECT(B' \rightarrow \varepsilon) = \{b\} \cap \{l\} = \emptyset$

占位符而已	a	b	l	d	#
$A$	$A \rightarrow aA'$				
$A'$	$A' \rightarrow AB l$			$A' \rightarrow \varepsilon$	$A' \rightarrow \varepsilon$
$B$				$B \rightarrow dB'$	
$B'$		$B' \rightarrow bB'$	$B' \rightarrow \varepsilon$		

二．试构造与正规式 $R = (a|b)^*b$ 等价的状态最少的DFA。



	a	b
X12	12	12Y
12	12	12Y
12Y	12	12Y

	a	b
1	2	3
2	2	3

三. 已知文法G[S]:  $S \rightarrow a|(T)$

$T \rightarrow T,S|S$

对文法消除左递归, 然后判断是不是LL(1)文法? 若是, 构造该文法预测分析表。

答: 1) 消除文法左递归, 得到文法  $G' [S]$ :

$S \rightarrow a|\Lambda|(T) \quad T \rightarrow ST' \quad T' \rightarrow ,ST'|\epsilon$

2)  $SELECT(S \rightarrow a) \cap SELECT(S \rightarrow \Lambda) = \Phi$

$SELECT(S \rightarrow a) \cap SELECT(S \rightarrow (T)) = \Phi$

$SELECT(S \rightarrow \Lambda) \cap SELECT(S \rightarrow (T)) = \Phi$

$SELECT(T' \rightarrow ,ST') \cap SELECT(T' \rightarrow \epsilon)$

$= FIRST(,ST') \cap (FIRST(\epsilon) \cup FOLLOW(T'))$

$= \{, \} \cap \{\epsilon, )\} = \Phi$

所以文法  $G' [S]$  是 LL(1)文法。[https://www.csdn.net/m0\\_37482190](https://www.csdn.net/m0_37482190)

3) 对规则  $S \rightarrow a$ , 因为  $FIRST(a) = \{a\}$ , 所以  $M[S, a] = S \rightarrow a$

对规则  $S \rightarrow \Lambda$ , 因为  $FIRST(\Lambda) = \{\Lambda\}$ , 所以  $M[S, \Lambda] = S \rightarrow \Lambda$

对规则  $S \rightarrow (T)$ , 因为  $FIRST((T)) = \{($ , 所以  $M[S, (] = S \rightarrow (T)$

对规则  $T \rightarrow ST'$ , 因为  $FIRST(ST') = \{a, \Lambda, ($ , 所以  $M[T, a] = T \rightarrow ST'$

$M[T, \Lambda] = T \rightarrow ST'$

$M[T, (] = T \rightarrow ST'$

对规则  $T' \rightarrow , ST'$ , 因为  $FIRST(, ST') = \{ , \}$ , 所以  $M[T' , ,] = T' \rightarrow , ST'$

对规则  $T' \rightarrow \epsilon$ , 因为  $FOLLOW(T') = \{ ) \}$ , 所以  $M[T' , )] = T' \rightarrow \epsilon$

文法预测分析表:

	a	$\Lambda$	(	)	,	#
S	$S \rightarrow a$	$S \rightarrow \Lambda$	$S \rightarrow (T)$			
T	$T \rightarrow ST'$	$T \rightarrow ST'$	$T \rightarrow ST'$			
T'				$T' \rightarrow \epsilon$	$T' \rightarrow , ST'$	

#### 4.把下面文法改为无二义的: $S \rightarrow SS | (S) | ()$

答:

$S \rightarrow TS | T$

$T \rightarrow (S) | ()$

#### 1.已知右线性正则文法G (M) :

$M \rightarrow aM | bN | a | b$

$N \rightarrow aN | bN | \epsilon$

①写出等价的正则表达式

②构造识别其所定义语言的NFA

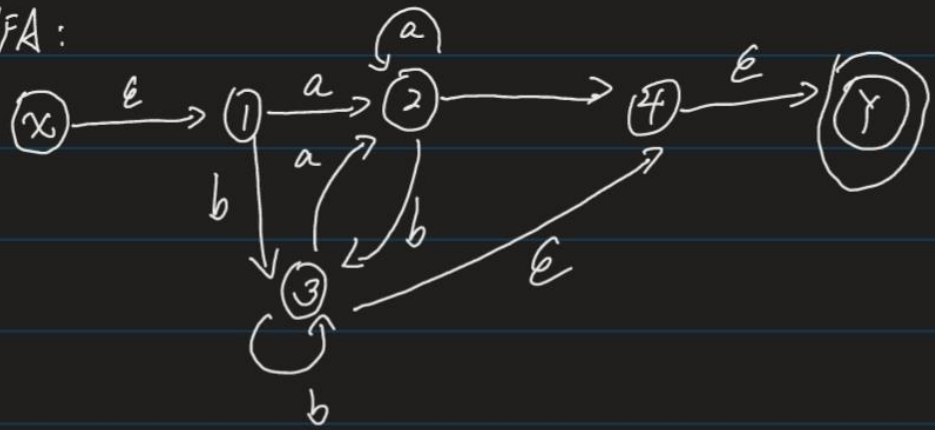
③构造识别的DFA

④构造识别L的最小DFA

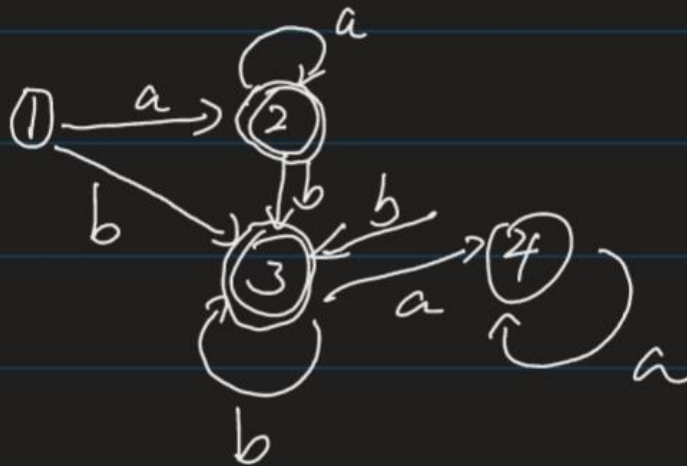
4). 正则表达式:  $a^* | bca^* | b^* )^*$

(2)  $L(G) = \{ a^* | bca^* | b^* )^* \}$

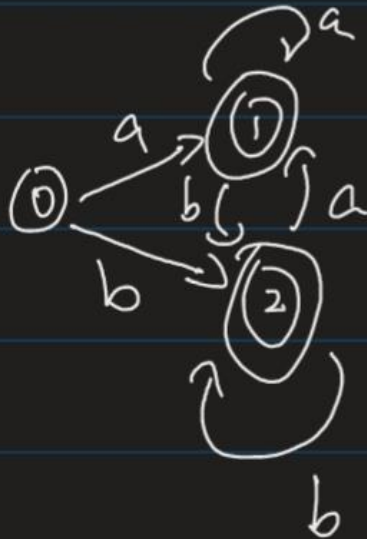
则 NFA:



(3)



4.

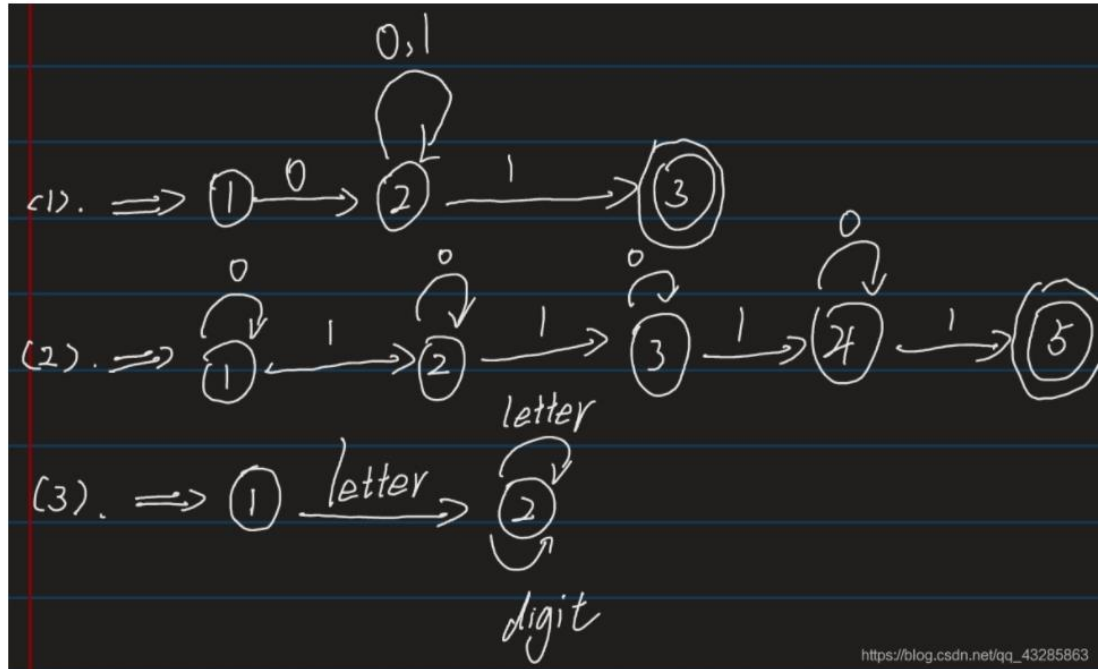


## 2.构造下列正规式相应的DFA，用状态转换图表示

①  $1(1|0)^*1$

②  $0^*10^*10^*10^*1$

③  $\text{letter}(\text{letter}|\text{digit})^*$



## DFA极小化

### 解题步骤

可列状态表也可不列

我主要使用的是划分的方法

将所有状态先划分为两个集合，终极符集和非终极符集

再对集合进一步划分，例如：

终极符集中，每一个状态过边的集合内的每一个元素可指向的状态是终极状态还是非终极状态，可将集合内继续进行划分

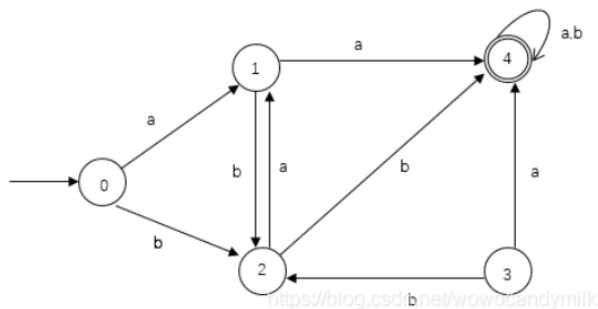
最终划分就是极小化的最终结果

划分内的每一个集合是极小化自动机的状态结点

### 例题

#### 例题一

将下图所示的DFA M进行最小化





划分  $\Pi_0 = (\{4\}, \{0, 1, 2, 3\})$

$G = \{0, 1, 2, 3\}$  中

$f(1, a) = 4$     $f(3, a) = 4$    指向终态

$f(0, a) = 1$     $f(2, a) = 1$    指向非终态

划分可以进行更改

$\Pi_1 = (\{4\}, \{0, 2\}, \{1, 3\})$

$G = \{0, 2\}$  中

$f(0, b) = 2$     $f(2, b) = 4$    0指向非终态, 2指向终态

划分可以进行修改

$\Pi_2 = (\{4\}, \{0\}, \{2\}, \{1, 3\})$

$G = \{1, 3\}$  中

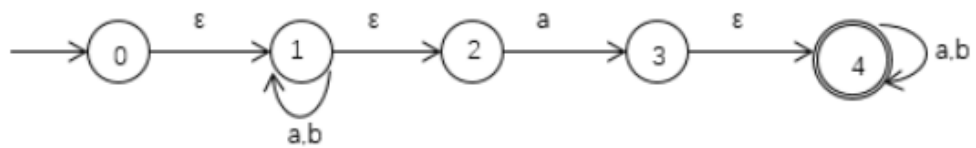
$f(1, b) = 2$     $f(3, b) = 2$    均指向非终态, 且均为2

故划分不修改

$\Pi = (\{4\}, \{0, 2\}, \{1, 3\})$

因此最终会有四个状态结点, 图可以进行更改, 删去3状态, 将1状态换成 $\{1, 3\}$ 状态, 图就不画了。

## 例题二



将这个非确定有限自动机确定化

$$T_0 = \mathcal{E}\text{-closure}(0) = \{0, 1, 2\}$$

$$\mathcal{E}\text{-closure}(\text{move}(T_0, a)) = \mathcal{E}\text{-closure}(\{1, 3\}) = \{1, 2, 3, 4\} = T_1$$

$$\mathcal{E}\text{-closure}(\text{move}(T_0, b)) = \mathcal{E}\text{-closure}(\{1\}) = \{1, 2\} = T_2$$

$$\mathcal{E}\text{-closure}(\text{move}(T_1, a)) = \mathcal{E}\text{-closure}(\{1, 3, 4\}) = \{1, 2, 3, 4\} = T_1$$

$$\mathcal{E}\text{-closure}(\text{move}(T_1, b)) = \mathcal{E}\text{-closure}(\{1, 4\}) = \{1, 2, 4\} = T_3$$

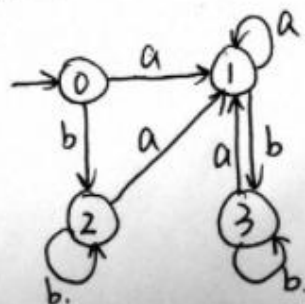
$$\mathcal{E}\text{-closure}(\text{move}(T_2, a)) = \mathcal{E}\text{-closure}(\{1, 3\}) = \{1, 2, 3, 4\} = T_1$$

$$\mathcal{E}\text{-closure}(\text{move}(T_2, b)) = \mathcal{E}\text{-closure}(\{1\}) = T_2$$

$$\mathcal{E}\text{-closure}(\text{move}(T_3, a)) = \mathcal{E}\text{-closure}(\{1, 3, 4\}) = T_1$$

$$\mathcal{E}\text{-closure}(\text{move}(T_3, b)) = \mathcal{E}\text{-closure}(\{1, 4\}) = T_3$$

I	Ia	Ib	状态
$\{0, 1, 2\}$	$\{1, 2, 3, 4\}$	$\{1, 2\}$	0.
$\{1, 2, 3, 4\}$	$\{1, 2, 3, 4\}$	$\{1, 2, 4\}$	1
$\{1, 2\}$	$\{1, 2, 3, 4\}$	$\{1, 2\}$	2
$\{1, 2, 4\}$	$\{1, 2, 3, 4\}$	$\{1, 2, 4\}$	3.



### 例题一

文法:

$S \rightarrow MH|a$

$H \rightarrow LSo|\epsilon$

$K \rightarrow dML|\epsilon$

$L \rightarrow eHf$

$M \rightarrow K|bLM$

判断 G 是否为 LL(1) 文法, 如果是, 构造 LL(1) 分析表。

FIRST集和FOLLOW集

	FIRST	FOLLOW
S	{a,d,b,ε,e}	{#,o}
M	{d,ε,b}	{e,#,o}
H	{ε,e}	{#,f,o}
L	{e}	{a,d,b,e,o,#}
K	{d,ε}	{e,#,o}

PREDICT集

	SELECT
$S \rightarrow MH$	{d,b,e,#,o}
$S \rightarrow a$	{a}
$H \rightarrow LSo$	{e}
$H \rightarrow \epsilon$	{#,f,o}
$K \rightarrow dML$	{d}
$K \rightarrow \epsilon$	{e,#,o}
$L \rightarrow eHf$	{e}
$M \rightarrow K$	{d,e,#,o}
$M \rightarrow bLM$	{b}

## 预测分析表

	a	o	d	e	f	b	#
S	->a	->MH	->MH	->MH		->MH	->MH
M		->K	->K	->K		->bLM	->K
H		-> $\epsilon$		->LS <sub>0</sub>	-> $\epsilon$		-> $\epsilon$
L				->eHf			
K		-> $\epsilon$	->dML	-> $\epsilon$			-> $\epsilon$

预测分析表看出没有多重入口，所以是LL(1)文法

### 例题二

$E \rightarrow TE'$

$E' \rightarrow +E \mid \epsilon$

$T \rightarrow FT'$

$T' \rightarrow T \mid \epsilon$

$F \rightarrow PF'$

$F' \rightarrow *F' \mid \epsilon$

$P \rightarrow (E) \mid a \mid b \mid ^$

(1) 计算这个文法的每个非终结符的 FIRST 集和 FOLLOW 集。

(2) 证明这个方法是 LL(1) 的。

(3) 构造它的预测分析表。

解：

(1) 计算这个文法的每个非终结符的 FIRST 集和 FOLLOW 集。

FIRST 集合有：

$FIRST(E) = FIRST(T) = FIRST(F) = FIRST\$ = \{(, a, b, ^)\}$ ;

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

$FIRST(T) = FIRST(F) = FIRST\$ = \{(, a, b, ^)\}$ ;

$FIRST(T') = FIRST(T) \cup \{\epsilon\} = \{(, a, b, ^, \epsilon)\}$ ;

$FIRST(F) = FIRST\$ = \{(, a, b, ^)\}$ ;

$FIRST(F') = FIRST\$ = \{undefined, \epsilon\}$ ;

$FIRST\$ = \{(, a, b, ^)\}$ ;

FOLLOW 集合有：

$FOLLOW(E) = \{, \#\}$ ;

$FOLLOW(E') = FOLLOW(E) = \{, \#\}$ ;  $FOLLOW(T) = FIRST(E') \cup FOLLOW(E) = \{+, , \#\}$ ; // 不包含  $\epsilon$

$FOLLOW(T') = FOLLOW(T) = FIRST(E') \cup FOLLOW(E) = \{+, , \#\}$ ;  $FOLLOW(F) = FIRST(T') \cup FOLLOW(T) = \{(, a, b, ^, +, , \#\}$ ; // 不包含  $\epsilon$

$FOLLOW(F') = FOLLOW(F) = FIRST(T') \cup FOLLOW(T) = \{(, a, b, ^, +, , \#\}$ ;  $FOLLOW\$ = FIRST(F') \cup FOLLOW(F) = \{(, a, b, ^, +, , \#\}$ ; // 不包含  $\epsilon$

(2)证明这个方法是 LL(1)的。

各产生式的 SELECT 集合有：

SELECT( $E \rightarrow TE'$ )=FIRST( $T$ )= $\{ (, a, b, ^ \}$ ;

SELECT( $E' \rightarrow +E$ )= $\{ + \}$ ;

SELECT( $E' \rightarrow \epsilon$ )=FOLLOW( $E/$ )= $\{ ), \# \}$

SELECT( $T \rightarrow FT'$ )=FIRST( $F$ )= $\{ (, a, b, ^ \}$ ;

SELECT( $T' \rightarrow T$ )=FIRST( $T$ )= $\{ (, a, b, ^ \}$ ;

SELECT( $T' \rightarrow \epsilon$ )=FOLLOW( $T/$ )= $\{ +, ), \# \}$ ;

SELECT( $F \rightarrow PF'$ )=FIRST( $\$$ )= $\{ (, a, b, ^ \}$ ;

SELECT( $F' \rightarrow F'$ )= $\{ \}$ ;

SELECT( $F' \rightarrow \epsilon$ )=FOLLOW( $F'$ )= $\{ (, a, b, ^, +, ), \# \}$ ;

SELECT( $P \rightarrow (E)$ )= $\{ ( \}$

SELECT( $P \rightarrow a$ )= $\{ a \}$

SELECT( $P \rightarrow b$ )= $\{ b \}$

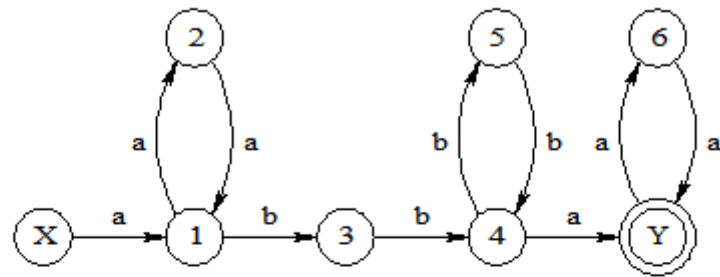
SELECT( $P \rightarrow \epsilon$ )= $\{ \}$

可见，相同左部产生式的 SELECT 集的交集均为空，所以文法 G[E]是 LL(1)文法。

(3)构造它的预测分析表。 文法 G[E]的预测

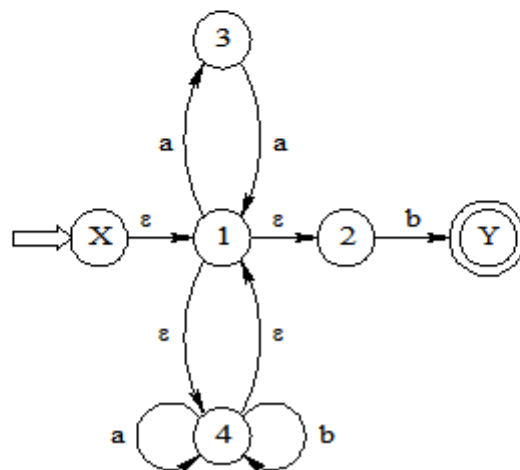
	+	*	(	)	a	b	^	#
E			$\rightarrow TE'$		$\rightarrow TE'$	$\rightarrow TE'$	$\rightarrow TE'$	
$E'$	$\rightarrow +E$			$\rightarrow \epsilon$				$\rightarrow \epsilon$
T			$\rightarrow FT'$		$\rightarrow FT'$	$\rightarrow FT'$	$\rightarrow FT'$	
$T'$	$\rightarrow \epsilon$		$\rightarrow T$	$\rightarrow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

1. 构造正规表达式 $a(aa)^*bb(bb)a(aa)$  的NFA。



解：

2. 构造正规表达式 $((a|b)^*|aa)^*b$ 的NFA。



解：

5. 给定文法： $S \rightarrow (L) \mid a$

$L \rightarrow L, S \mid S$

请书写语义规则，求输出句子中每一个a的括号嵌套深度。

解：用继承属性depth表示嵌套深度，则

1	$S' \rightarrow S$	$S.depth = 0$
2	$S \rightarrow (L)$	$L.depth = S.depth + 1$
3	$S \rightarrow a$	$print(S.depth)$
4	$L \rightarrow L(1), S$	$L(1).depth = L.depth; S.depth = L.depth$
5	$L \rightarrow S$	$S.depth = L.depth$

10. 将下面的语句翻译成四元式序列：

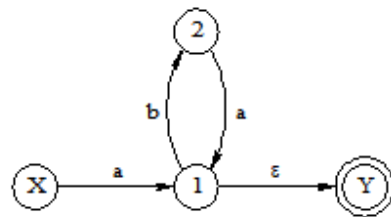
if ( $x > y$ )  $m = 1$ ;

else  $m = 0$ ;

解：

1		1	( $j >$ , $x, y, 3$ )
2		2	( $j$ , $_{-}$ , $_{-}$ , 5)
3		3	( $=$ , 1, $_{-}$ , $m$ )
4		4	( $j$ , $_{-}$ , $_{-}$ , 6)
5		5	( $=$ , 0, $_{-}$ , $m$ )
6		6	:

10. 将以下DFA 最小化。（8分）



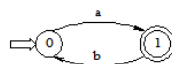
解：

1. 将状态分解为
  - 终态集{Y}
  - 非终态集{X,1,2}
2. 考察非终态集{X,1,2}接收a字符
  - X接收a字符到1 （1属于非终态集）
  - 1不接收字符
  - 2接收a字符到1 （1属于非终态集）
 故将非终态集{X,1,2}, 分为{X,2}, {1}

3. 考察{X,2}接收b字符
  - X接收不接受b
  - 2接收不接受b

4. 按顺序重新命名状态子集{X,2}, {1}, {Y}为0, 1, 2得到状态转换矩阵

s	a	b
0	1	空
1	空	0
2		



5. 根据状态转换矩阵绘制化简后的状态转换图



14. 试构造下述文法的LL(1)分析表。(15分)

$G[S]: S \rightarrow (L) \mid a$

$L \rightarrow L, S \mid S$

解：消除左递归：

$G(S): S \rightarrow (L) \mid a$

$L \rightarrow SL'$

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

构造FIRST集，如下：

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

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

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

构造FOLLOW集如下：

(1)  $FOLLOW(S) = \{#, , , )\}$

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

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

LL(1)分析表

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

15. 判断文法 $G[S]: S \rightarrow BA$

$A \rightarrow BS \mid d$

$B \rightarrow aA \mid bS \mid c$

是否为LL(1)文法.

解：对于该文法求其FIRST集如下：

$FIRST(S) = \{a, b, c\}$ ;

$FIRST(A) = \{a, b, c, d\}$ ;

$FIRST(B) = \{a, b, c\}$

求其FOLLOW集如下：

$FOLLOW(S) = \{a, b, c, d, \#\}$ ;

$FOLLOW(A) = \{a, b, c, d, \#\}$ ;

$FOLLOW(B) = \{a, b, c, d, \#\}$

由 $A \rightarrow BS \mid d$ 得：

$FIRST(BS) \cap FIRST(d) = \{a, b, c\} \cap \{d\} = \Phi$

由 $B \rightarrow aA \mid bS \mid c$ 得

$FIRST(aA) \cap FIRST(bS) \cap FIRST(c) = \{a\} \cap \{b\} \cap \{c\} = \Phi$

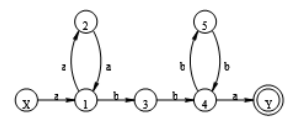
由于文法 $G[S]$ 不存在形如 $\beta \rightarrow \varepsilon$ 的产生式，故无需求解形如 $FIRST(a) \cap FOLLOW(A)$ 的值，也即文法 $G[S]$ 是一个LL(1)文法。

18. 将下面的语句翻译成四元式序列：

```
1 | while(a<b)
2 | if (c>d)  x=y+z
```

```
1 | 解: 100 (j<,a,b,102)
2 | 101 (j,_,_,107)
3 | 102 (j>,c,d,104)
4 | 103 (j,_,_,106)
5 | 104 (+,y ,z ,t)
6 | 105 (=,t ,_ ,x)
7 | 106 (j,_,_,100)
8 | 107:
```

19. 构造正规表达式 $a(aa)^*bb(bb)^*a$  的最小化的确定有限自动机 $M'$ 。  
解：先画出正规式相应的NFA  $M$ 状态图，如下图所示。

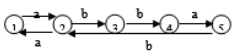


用子集法构造状态转换矩阵，如下表所示。

I	I <sub>a</sub>	I <sub>b</sub>
{x}	{1}	-
{1}	{2}	{3}
{2}	{1}	-
{3}	-	{4}
{4}	{5}	{5}
{5}	-	{4}
{Y}	-	-

将状态分为终态集 $\{Y\}$ 和非终态集 $\{X,1,2,3,4,5\}$   
因为  $\{X,1,2,3,4,5\} \xrightarrow{a} \{1,2,1,,Y\}$   
所以非终态集分为 $\{X,1,2\},\{3,5\},\{4\}$   
因为 $\{X,1,2\} \xrightarrow{b} \{3,\}$ ,所以分为  
最后得到集合 $\{X,2\},\{1\},\{3,5\},\{4\},\{Y\}$ 重新命名为1, 2, 3, 4, 5得到最小化的DFA  $M'$ 状态转换矩阵和状态转换图如下图所示。

I	I <sub>a</sub>	I <sub>b</sub>
1	2	-
2	1	3
3	-	4
4	5	3
5	-	-



25. 把下列语句翻译为四元式序列：

```
1  while (A > B)
2      if (C > D)
3  X = Y * Z
4      else
5          X = Y + Z
```

解：

```
1  (1) (j>, A, B, 3)
2  (2) (j, _, _, 11)
3  (3) (j>, C, D, 5)
4  (4) (j, _, _, 8)
5  (5) (*, Y, Z, T1)
6  (6) (=, T1, _, X)
7  (7) (j, _, _, 1)
8  (8) (+, Y, Z, T2)
9  (9) (=, T2, _, X)
10 (10) (j, _, _, 1)
11 (11)
```

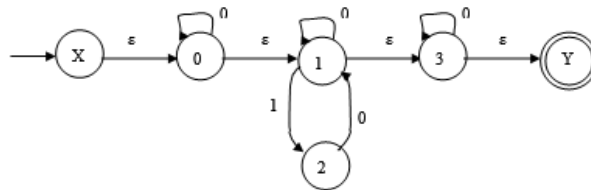
26. 构造一个DFA，它接受 $\Sigma = \{0, 1\}$ 上所有满足如下条件的字符串：每个1后面都有0直接跟在右边。

解：

(1)  $0^*(0|10)^*0$  或者  $(0|10)^*$

(2)

①NFA (2分)



②子集法确定化

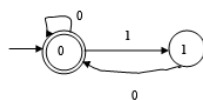
I	I <sub>0</sub>	I <sub>1</sub>
{X, 0, 1, 3, Y}	{0, 1, 3, Y}	{2}
{0, 1, 3, Y}	{0, 1, 3, Y}	{2}
{2}	{1, 3, Y}	-
{1, 3, Y}	{1, 3, Y}	{2}

重新命名状态，即得：

S	0	1
1	2	3
2	2	3
3	4	-
4	4	3

③ 最小化

首先分为终态集和非终态集 {3} {1, 2, 4} 因为  $10 = 2$   $20 = 2$   $40 = 4$  状态均属于集合{1, 2, 4}，所以对于输入符号0不能区分1, 2, 4三个状态； $11 = 3$   $21 = 3$   $41 = 3$ 状态均属于集合{3}，所以对于输入符号1也不能区分1, 2, 4三个状态；因此最终的状态划分即为: {3} {1, 2, 4}，其对应的DFA如下图所示：



27. 已知文法G(S):

$$S \rightarrow S^*aP \mid aP \mid ^*aP$$

$$P \rightarrow +aP \mid +a$$

(1) 将文法G(S)改写为LL(1)文法G'(S);

(2) 写出文法G'(S)的预测分析表。

解: (1) 消除左递归, 文法变为:

$$S \rightarrow aPS' \mid ^*aPS'$$

$$S' \rightarrow aPS' \mid \varepsilon$$

$$P \rightarrow +aP \mid +a$$

提取公共左因子, 文法变为G'(S):

$$S \rightarrow aPS' \mid ^*aPS'$$

$$S' \rightarrow aPS' \mid \varepsilon$$

$$P \rightarrow +aP'$$

$$P' \rightarrow P \mid \varepsilon$$

(2) 计算每个非终结符的FIRST集和FOLLOW集:

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

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

$$FIRST\$_ = \{+\}$$

$$FIRST(P') = \{+, \varepsilon\}$$

$$FOLLOW(S) = \{\#\}$$

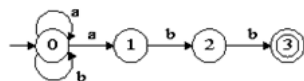
$$FOLLOW(S') = \{\#\}$$

$$FOLLOW\$_ = \{, \#\}$$

$$FOLLOW(P') = \{, \#\}$$

30. 对正规式(a|b)\*abb构造其等价的NFA。

【解】



31. 下面的文法产生0和1的串, 即二进制的正整数, 请给出决定每个二进制数的值(十进制形式)的语法制导定义。

【解】定义值属性为val, 翻译方案如下:

$$B \rightarrow B_1 0 \{ B.val = B_1.val \times 2 \}$$

$$B \rightarrow B_1 1 \{ B.val = B_1.val \times 2 + 1 \}$$

$$B \rightarrow 1 \{ B.val = 1 \}$$

$$B \rightarrow 0 \{ B.val = 0 \}$$

39. 程序的文法如下:

$P \rightarrow D$

$D \rightarrow D; D \mid id : T \mid proc\ id; D; S$

写一语法制导定义, 打印该程序一共声明了多少个id

【解】 属性num表示id个数

1	$P \rightarrow D$	<code>print(D.num)</code>
2	$D \rightarrow D(1); D(2)$	<code>D.num = D(1).num + D(2).num</code>
3	$D \rightarrow id : T$	<code>D.num = 1</code>
4	$D \rightarrow proc\ id; D(1); S$	<code>D.num = D(1).num + 1</code>
5		

57. 有一语法制导定义如下:

1	$S \rightarrow bAb$	<code>print "1"</code>
2	$A \rightarrow ( B$	<code>print "2"</code>
3	$A \rightarrow a$	<code>print "3"</code>
4	$B \rightarrow aA)$	<code>print "4"</code>

若输入序列为b ( a ( a ( aa ) ) ) b, 且采用自下而上的分析方法, 则输出序列为(34242421\_\_ )。

59. 为文法

$G[S]: S \rightarrow (L) \mid a$

$L \rightarrow L, S \mid S$

写一语法制导定义, 它输出句子中括号嵌套的最大层次数。

解: 使用num属性描述括号的嵌套最大层次数

$S' \rightarrow S$	<code>print (S.num) ←</code>
$S \rightarrow (L)$	<code>S.num=L.num+1 ←</code>
$S \rightarrow a$	<code>S.num=0 ←</code>
$L \rightarrow L^{(i)}, S$	<code>L.num=if L<sup>(i)</sup>.num&gt;S.num then L<sup>(i)</sup>.num else S.num ←</code>
$L \rightarrow S$	<code>L.num= S.num ←</code>

## 作业一

1、考虑下面的CFG:  $S \rightarrow SS+ \mid SS* \mid a$  以及串  $aa+a*$

- (1) 给出这个串的一个最左推导;
- (2) 画出这个串的语法树;
- (3) 分析该文法是否为二义性的;
- (4) 描述这个文法产生的语言。

2、对下面的每一对文法和串重复练习1:

- (1)  $S \rightarrow 0S1 \mid 01$  和串000111;
- (2)  $S \rightarrow +SS \mid *SS \mid a$  和串+\*aaa;
- (3)  $S \rightarrow S(S)S \mid \varepsilon$  和串( $()()$ );
- (4)  $S \rightarrow aSbS \mid bSaS \mid \varepsilon$  和串aabbab;

50

## 作业二

3、对如下文法（把斜体部分都看做 $V_T$ ）:

$\langle \text{条件语句} \rangle \rightarrow IF \text{ 条件 } THEN \langle \text{语句} \rangle ELSE \langle \text{语句} \rangle$

$\langle \text{条件语句} \rangle \rightarrow IF \text{ 条件 } THEN \langle \text{语句} \rangle$

$\langle \text{语句} \rangle \rightarrow \langle \text{条件语句} \rangle \mid \text{其他语句}$

- (1) 证明该文法是二义的;
- (2) 将其改造成无二义的。

文法G[S]:

$S \rightarrow a | b | (T) \quad T \rightarrow T \wedge S | S$

1. 将其变换成等价的LL(1)文法;
2. 用伪代码写出不带回溯的递归子程序;
3. 构造预测分析表;
4. 写出对句子((a) ∧ (b))的非递归预测分析过程。

❖ (P191) 下列文法由开始符号S产生一个带有“小数点”的二进制数:

$S \rightarrow L . L | L$

$L \rightarrow L B | B$

$B \rightarrow 0 | 1$

设计一个SDD来求S.val, 即输入串对应的十进制数值。例如, 输入串为101.101时应该输出5.625。

方法1:使用L.val、L.len属性

$S \rightarrow L \quad \{ S.val = L.val \}$

$S \rightarrow L_1 . L_2 \quad \{ S.val = L_1.val + L_2.val / 2^{L_2.len} \}$

$L \rightarrow B \quad \{ L.val = B.c, L.len = 1 \}$

$L \rightarrow L_1 B$

$\{ L.val = 2 * L_1.val + B.c, L.len = L_1.len + 1 \}$

$B \rightarrow 0 \quad \{ B.c = 0 \}$

$B \rightarrow 1 \quad \{ B.c = 1 \}$



## 方法2：使用L.lval、L.w属性

```
S → L      { S.val = L.lval }
S → L1.L2 { S.val = L1.lval + L2.lval / L2.w }
L → B      { L.lval = B.c, L.w = 2 }
L → L1B
              { L.lval = 2 * L1.lval + B.c, L.w = 2 * L1.w }
B → 0      { B.c = 0 }
B → 1      { B.c = 1 }
```

## 方法3：使用继承属性L.side来指明一个二进制位在小数点的哪一边（自己练习）

❖ 在C语言中，自增运算符只能作用于“左值”（如变量名），而3++和( id + id )++这样的表达式在编译时都会得到如下的错误提示：

invalid lvalue in increment

现有如下简化的C语言表达式文法：

$E \rightarrow E + E \mid ( E ) \mid E ++ \mid id \mid number$

写出一个语法制导定义或翻译方案，它检查++的运算对象是否合法。

## 作业二

❖ (P195) 设计一个SDD，将一个带有+和\*的中缀表达式翻译成没有冗余括号的表达式。比如，因为两个运算符都是左结合的，并且\*的优先级高于+，所以

$((a*(b+c))*(d))$

可翻译为：

$a*(b+c)*d$

注意：为了降低难度，可以无二义的左递归文法作为基础文法进行分析。

将下面的语句翻译成某种形式的中间代码（四元式、三地址赋值、类汇编…）。

```
while A<C and B>D do
    if A=1 then C:=C+1
    else B:=B+5*A
```

Pascal语言有如下语法：

```
var a,b : array[1..100] of integer;
```

    a:=b           ——允许数组之间直接赋值

若a和b是同一类型记录，也允许

但是C语言中数组不行，结构体可以

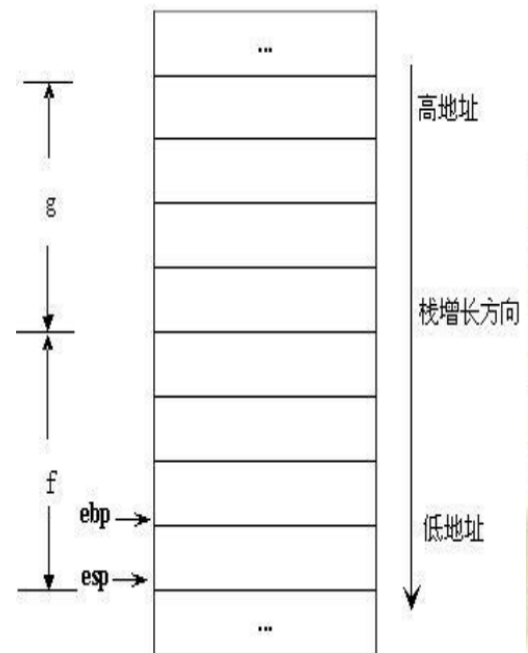
要求：为Pascal的这种数组赋值设计一种语法制导定义

某C程序中有如下两个函数：

```
int f(int x){
    // f函数只被g函数调用
    int i ;
    //在此处设置断点
    return i+1 ;
}

int g(int y){
    int j ;
    j = f(y+1) ;
    return j ;
}
```

在x86/Linux上运行该程序至断点时，活动记录栈中关于f，g 的部分如右图所示，请在图中填写栈里每一单元的内容。



(本题60分，每小题20分) 分析如下的C代码片段：

```
{
    int x,y;
    if(x>0) y=1;
    else y= -1;
}
```

写出其经过：

- (1) 词法分析
- (2) 语法分析
- (3) 代码翻译

处理之后的结果。

### 1. (简答题)

设  $\Sigma = \{a,b\}$ ,  $L = \{ \text{包含子串 aba 的串} \}$ , 求：

- (1) 能表达语言L的正规式；
- (2) 能识别语言L的NFA；
- (3) 能识别语言L的DFA；
- (4) 能识别语言L的状态最少的DFA。

注：对复杂部分如 (3)、(4)要写出分析过程。

1. (简答题)

(40分) 考虑下面的CFG  $G[S]: S \rightarrow SS+ \mid SS* \mid \text{number}$

- (1) 给出串 $35+2*$ 的一个最左推导和一个最右推导;
- (2) 画出与上述推导过程相对应的语法树;
- (3) 分析该文法是否为二义性的;
- (4) 该文法表示什么语言。

2. (简答题) (30分) 写一个CFG, 产生语言: { 能被5整除且不以0开头的无符号十进制整数 }

3. (简答题)

(30分) 把下面文法改写为无二义的:  $G[S] \quad S \rightarrow SS \mid aSb \mid ab$

要求: 分析该文法的二义性来源, 并论述如何消除二义性。

**正确答案:**

答案不唯一, 参考答案如下:

- (1) 证明二义性: 考查句子 $ababab$ 即可。实际上考查超过2个的并列 $ab$ 的串都一样。
- (2) 限制 $S$ 产生连续 $ab$ 对的能力, 使用 $ST$ 或者 $TS$ 来取代 $SS$ , 打破对称性;
- (3) 改造:  $G[S]: S \rightarrow TS \mid T \quad T \rightarrow aSb \mid ab$ 。

1. (简答题, 30分) 对于文法 $G[S]: S \rightarrow a \mid b \mid (T) \quad T \rightarrow T \wedge S \mid S, V_n = \{S, T\}, V_t = \{a, b, (, ), \wedge\}$ , 把该文法改造为可做自上而下语法分析的LL(1)文法; 注意: 要写分析过程。

2. (简答题, 30分) 针对上述文法, 用伪代码写出不带回溯的递归子程序; 注意: 是伪代码。

3. (简答题, 20分) 构造上述文法的预测分析表;

2. (简答题)

(50分) 以作业中曾经处理过的后缀表达式文法为基础:

$S \rightarrow SS+ \mid SS* \mid \text{id}$

设计一个语法制导定义 (SDD), 将每一个输入的后缀表达式转换为等价的中缀表达式, 但不带冗余括号。如: 输入  $ab*cd++$ , 输出  $a*b+(c+d)$ ; 输入  $ab*cd+*e+$ , 输出  $a*b*(c+d)+e$ 。