

## 编译作业参考答案 第五次作业

自动机的题目每个人画法可能都不一样，参考答案给出的是去年助教个人习惯的做法

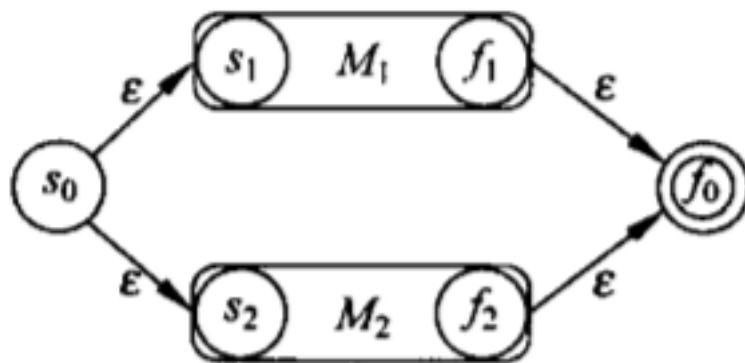
### P76 2

构造下列正则表达式相应的 DFA，并化简

- $\$1(0|1)^*|0\$$
- $1(1010^*|1(010)^*1)^*0$
- $1(0|1)^*101$

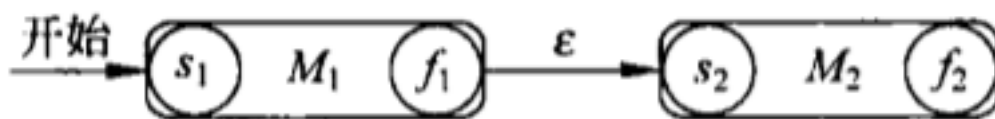
### 答案解析

这样的题目做法很多，最直接的方式是利用下面三种基本模型相互连接得到一个 NFA：

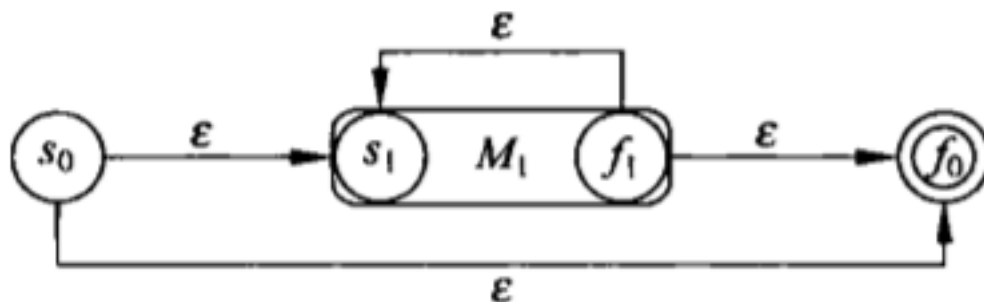


$M_1|M_2$ :

$M_1 M_2$ :



$M_1^*$ :



但是这样通常会得到一个过于巨大的 NFA，没有必要严格遵循上面的三种模板。

比如  $1(0|1)^*|0$ :

严格遵循模板会得到 NFA:

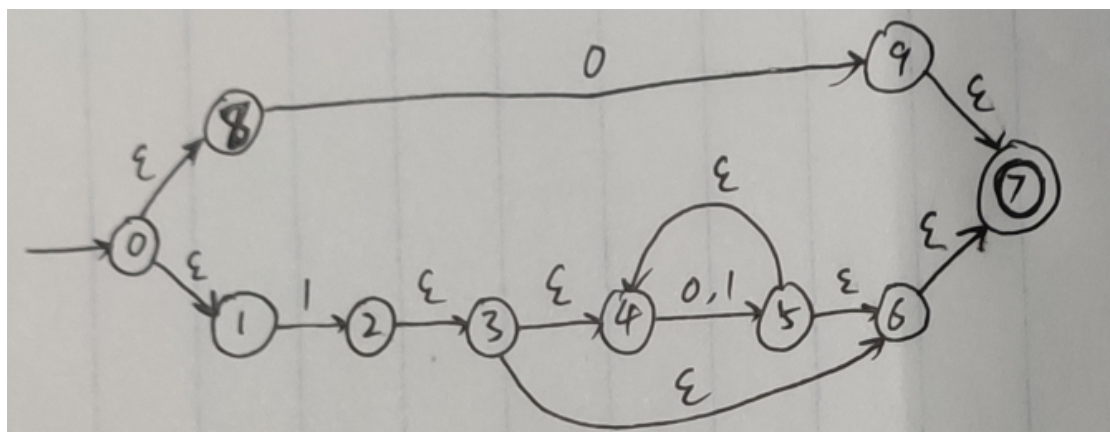


Figure 1: huge

稍微省略一些  $\epsilon$  边得到 NFA:

大道至简 NFA (看上去这么简单的 NFA，在确定化的时候可能会意外的麻烦):

另一种思路是首先人工化简正则表达式，之后再转化为 NFA; 比如利用正则式的结合律将  $1(10^*|11^*)$  化为  $11(0^*|1^*)$ 。

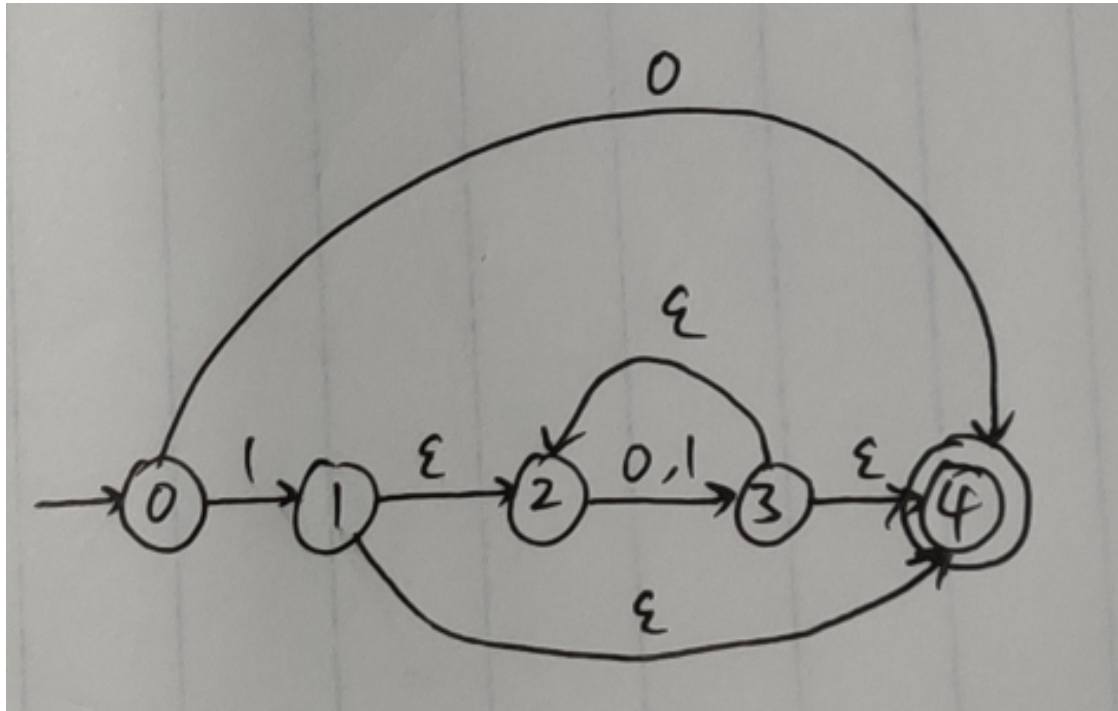


Figure 2: huge

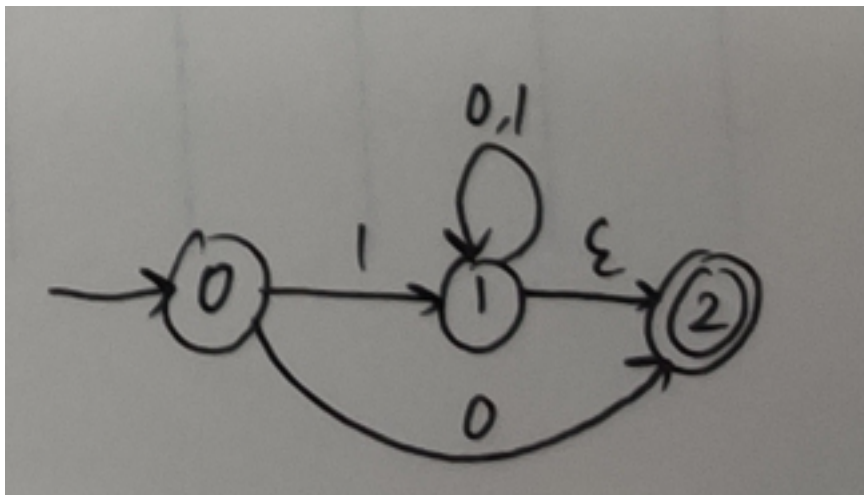


Figure 3: huge

但是这样得到的简化 NFA，确定化为 DFA 的过程不一定简单，有时候甚至是多此一举。

再一种思路是将一个大型正则式/NFA 进行分而治之的处理，这样可以在一定程度上避免 NFA 确定化时状态数过快增长的请款，但是每一次合并子问题都要重新确定化。

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

首先画出一个 NFA：

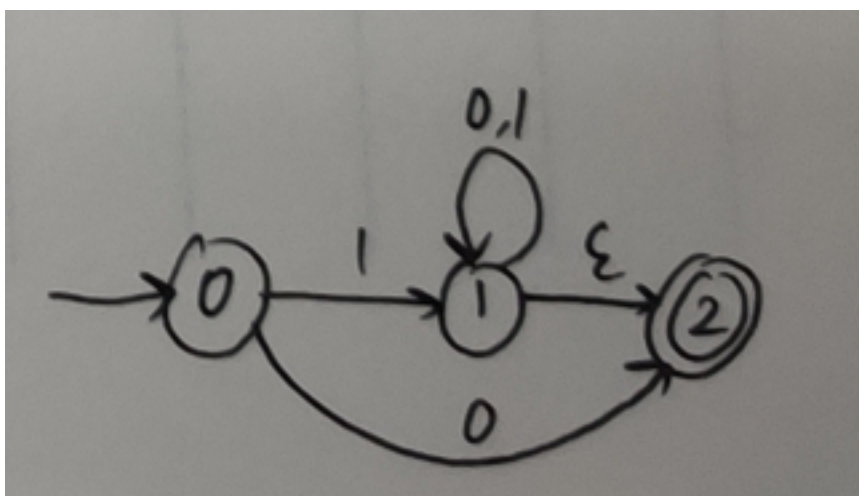


Figure 4: huge

进行确定化和重新编号：

最小化（终止状态红色斜体加粗）：

state	input 0	input 1
0	<b>2</b>	<b>1</b>
<b>1</b>	<b>1</b>	< style="color:red"> <b>1</b>
<b>2</b>	-	-

没有可合并的等价状态，说明这个 DFA 已经是最小化的了。

最终结果是  $M_1 = (S, \Sigma, \delta, s_0, Z)$ ：

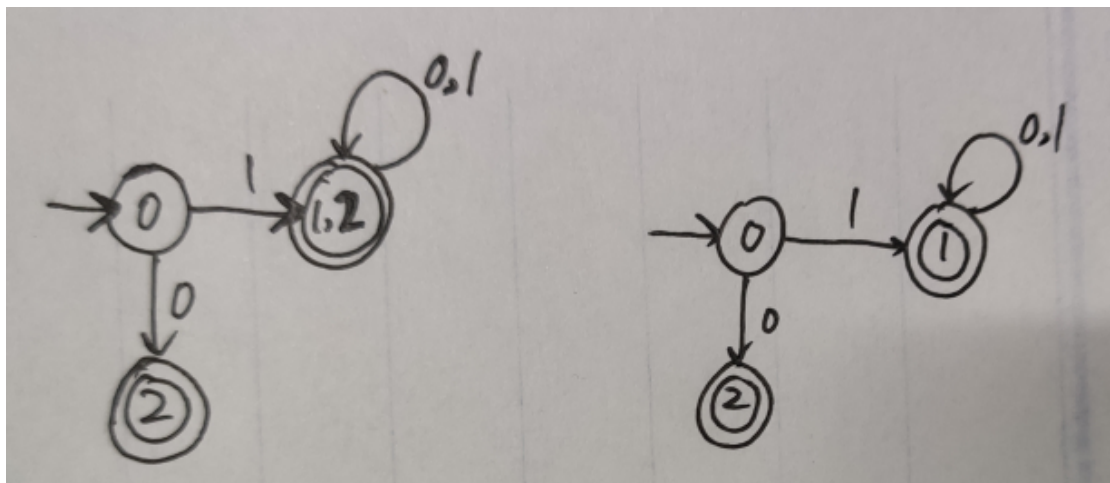


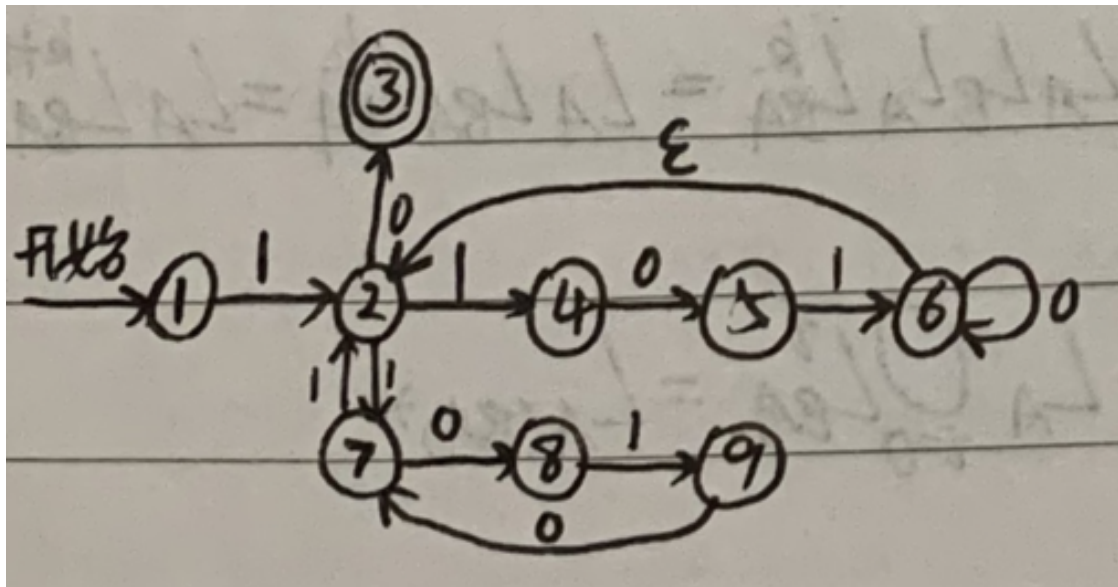
Figure 5: huge

- 有穷状态集  $S = \{0, 1, 2\}$
- 输入字母表  $\Sigma = \{0, 1\}$
- 状态转换函数:  $\delta(0, 0) = 2, \delta(0, 1) = 1, \delta(1, 0) = 1, \delta(1, 1) = 1$
- 开始状态  $s_0 = 0$
- 终止状态集  $Z = \{1, 2\}$

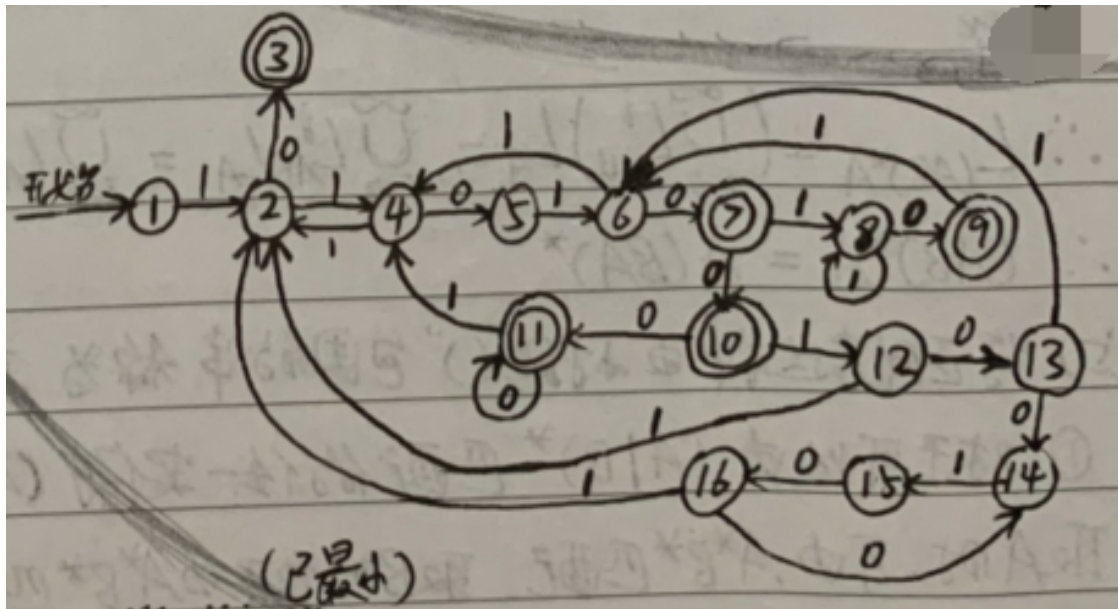
列出五元式的方式不限, 建议好懂一些, 比如  $M_1 = (\{0, 1, 2\}, \{0, 1\}, \delta, 0, \{1, 2\})$ ,  $\delta$  使用最小化的表格

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

NFA:



DFA:



最小化 (终止状态为斜体加粗):

state	input 0	input 1
1	-	2
2	<b>3</b>	4
4	5	2

state	input 0	input 1
5	-	6
6	7	4
8	<b>9</b>	8
12	13	2
13	14	6
14	-	15
15	16	-
16	14	2
<b>3</b>	-	-
7	<b>10</b>	8
<b>9</b>	-	6
<b>10</b>	<b>11</b>	12
<b>11</b>	<b>11</b>	4

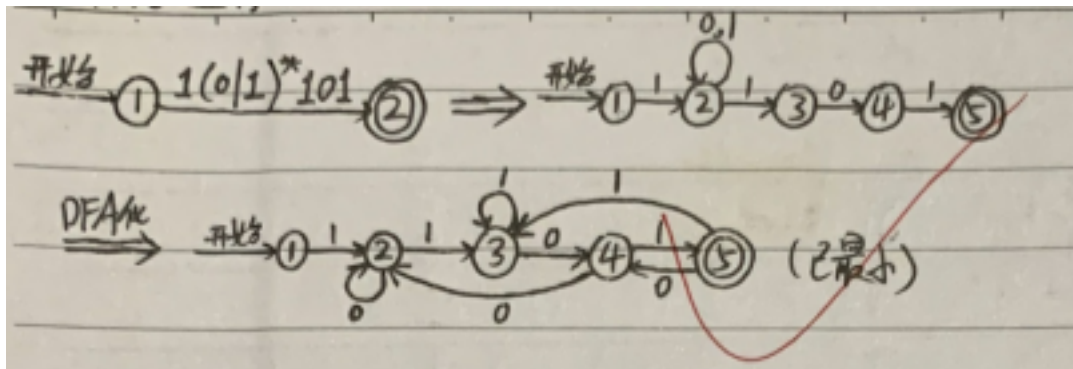
没有可合并的等价状态，说明这个 DFA 已经是最小化的了。

最终结果是  $M_2 = (S, \Sigma, \delta, s_0, Z)$ :

- 有穷状态集  $S = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16\}$
- 输入字母表  $\Sigma = \{0, 1\}$
- 状态转换函数见上表
- 开始状态  $s_0 = 1$
- 终止状态集  $Z = \{3, 7, 9, 10, 11\}$

$1(0|1)^*101$

画 NFA 与确定化:



最小化 (终止状态为斜体加粗):

state	input 0	input 1
1	-	2
2	2	3
3	4	3
4	2	<b>5</b>
<b>5</b>	4	3

没有可合并的等价状态, 说明这个 DFA 已经是最小化的了。

最终结果是  $M_3 = (S, \Sigma, \delta, s_0, Z)$ :

- 有穷状态集  $S = \{1, 2, 3, 4, 5\}$
- 输入字母表  $\Sigma = \{0, 1\}$
- 状态转换函数见上表
- 开始状态  $s_0 = 1$
- 终止状态集  $Z = \{5\}$

### 批改标准

- 中间步骤如果有画出 (比如 NFA 和非最小的 DFA), 则不能有错
- 最终得到的自动机和参考答案等价, 且已经最小化

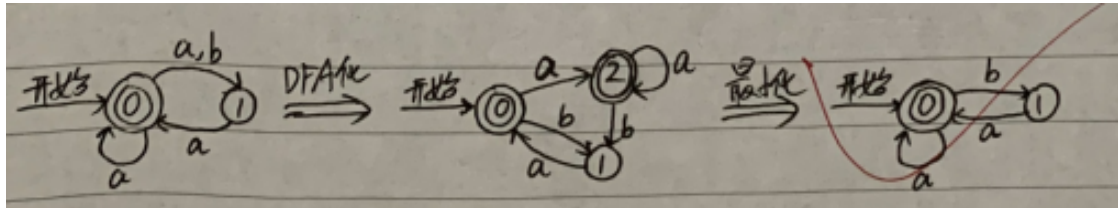


## P76 4

把两个自动机分别确定化和最小化

### 答案解析

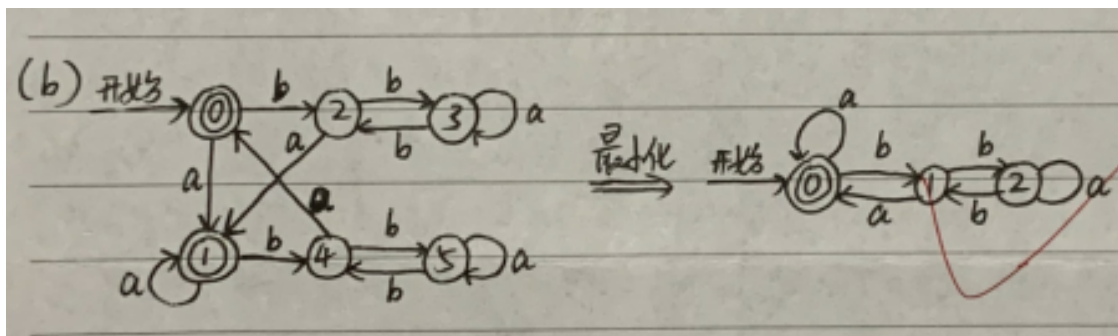
(a)



最终结果是  $M_4 = (S, \Sigma, \delta, s_0, Z)$ :

- 有穷状态集  $S = \{0, 1\}$
- 输入字母表  $\Sigma = \{a, b\}$
- 状态转换函数  $\delta(0, a) = 0, \delta(0, b) = 1, \delta(1, a) = 0$
- 开始状态  $s_0 = 0$
- 终止状态集  $Z = \{0\}$

(b)



最终结果是  $M_5 = (S, \Sigma, \delta, s_0, Z)$ :

- 有穷状态集  $S = \{0, 1, 2\}$
- 输入字母表  $\Sigma = \{a, b\}$

- 状态转换函数  $\delta(0, a) = 0, \delta(0, b) = 1, \delta(1, a) = 0, \delta(1, b) = 2, \delta(2, a) = 2, \delta(2, b) = 1$
- 开始状态  $s_0=0$
- 终止状态集  $Z = \{0\}$

### 批改标准

- 由于题目有歧义，4(a) 可以不最小化。
- 确定化得到的 DFA 和参考答案等价即可
- 最小化得到的 DFA 和参考答案等价即可

### P76 5

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

### 答案解析

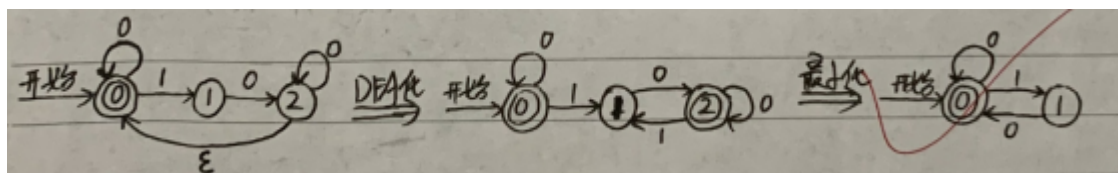
首先根据要求，写出正则：

每一个 1 后面都有 0 直接跟在右边，说明 1 接下来至少要有 1 个 0，则有  $100^*$ （拓展正则表达： $10^+$ ）。

将上述正则式视为一个基本单元，满足题目要求的字符串，可以有任意个上述单元出现，也可以是空串，则有  $(100^*)^*$ 。

由于全是 0 的串也可以满足题目要求，并且  $(100^*)^*$  前有前导 0 也满足，得到最终结果为： $0^*(100^*)^*$ 。

接下来根据正则式画出 NFA：



最终结果是  $M_6 = (S, \Sigma, \delta, s_0, Z)$ :

- 有穷状态集  $S = \{0, 1\}$
- 输入字母表  $\Sigma = \{0, 1\}$
- 状态转换函数  $\delta(0, 0) = 0, \delta(0, 1) = 1, \delta(1, 0) = 0$
- 开始状态  $s_0 = 0$
- 终止状态集  $Z = \{0\}$

## 批改标准

- 最终得到的自动机和参考答案等价即可，无需最小化

## P85 1

给出文法规则如下：

```

1  E -> E '+' T | T
2  T -> TF      | F
3  F -> F '*'   | P
4  P -> 'a'     | 'b'

```

- (1) 对这个文法提取左公因子
- (2) 提取左公因子后该文法能适用于自顶向下分析吗？
- (3) 提取左公因子后，从文法中消除左递归
- (4) 得到的文法适用于自顶向下分析吗？

## 答案解析

由于这个文法不存在任何一条规则的右侧有公因子，所以提取左公因子后还是和原来一样：

```

1  E -> E '+' T | T
2  T -> TF      | F
3  F -> F '*'   | P
4  P -> 'a'     | 'b'

```

由于规则： $E \rightarrow E' + T$ 、 $T \rightarrow TF$ 、 $F \rightarrow F' *'$ 都存在直接左递归，因此不适用于自顶向下分析。

为了消除左递归，改写文法：

```
1 E -> T{'+'T}
2 T -> F{F}
3 F -> P{'*'}
4 P -> 'a' | 'b'
```

此时任何一个规则都不存在任何形式的左递归了，并且对于  $P \rightarrow 'a' | 'b'$ ，由于  $FIRST(a) = \{a\} \neq \{b\} = FIRST(b)$ ，因此也不存在回溯问题。

此时的文法适用于自顶向下分析。

### 批改标准

- 做法不唯一，消除了左递归，并且没有回溯就行