编译原理第一次作业

任凯 2020141460080

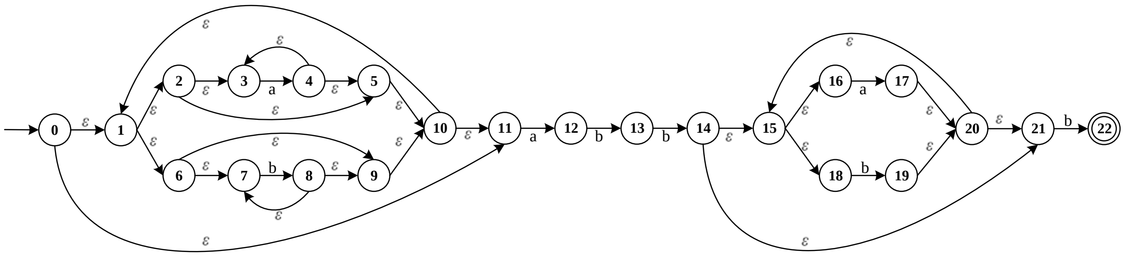
1. 已知 $\Sigma = \{a, b\}$ 上正则表达式为 $(a^*|b^*)^*abb(a|b)^*b$

（1） 该正则表达式所定义的语言是什么？

答：以 b 结尾的，且含有一个不包括结尾 b 的“abb”子串的 a，b 构成的串。

（2） 画出接收该语言的 NFA

答：该接收语言的 NFA 如下图所示：



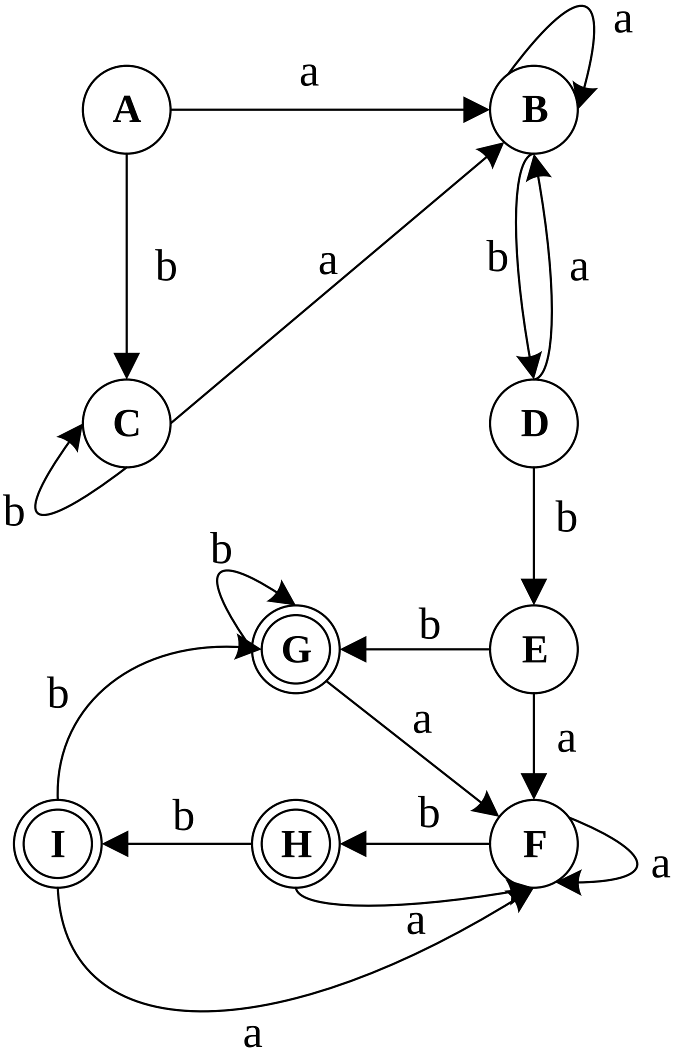
（3） 把该 NFA 转化成等价的 DFA

答：步骤如下

M	Set of NFA States	a	b
{0}	{0,1,2,3,5,6,7,9,10,11}	{4,12}	{8}
{4,12}	{1,2,3,4,5,6,7,9,10,11,12}	{4, 12}	{8,13}
{8}	{1,2,3,5,6,7,8,9,10,11}	{4, 12}	{8}
{8, 13}	{1,2,3,5,6,7,8,9,10,11,13}	{4, 12}	{8,14}
{8, 14}	{1,2,3,5,6,7,8,9,10,11,14,15,16,18,21}	{4,12,17}	{8,19,22}
{4,12,17}	{1,2,3,4,5,6,7,9,10,11,12,15,16,17,18,20,21}	{4,12,17}	{8,13,19,22}
{8,19,22}	{1,2,3,5,6,7,8,9,10,11,15,16,18,19,20,21,22}	{4,12,17}	{8,19,22}
{8,13,19,22}	{1,2,3,5,6,7,8,9,10,11,13,15,16,18,19,20,21,22}	{4,12,17}	{8,14,19,22}
{8,14,19,22}	{1,2,3,5,6,7,8,9,10,11,14,15,16,18,19,20,21,22}	{4,12,17}	{8,19,22}

NFA State	DFA State	a	b
{0}	A	B	C
{4,12}	B	B	D
{8}	C	B	C
{8, 13}	D	B	E
{8, 14}	E	F	G
{4,12,17}	F	F	H
{8,19,22}	G	F	G
{8,13,19,22}	H	F	I
{8,14,19,22}	I	F	G

进而可得等价的 DFA 如下图：

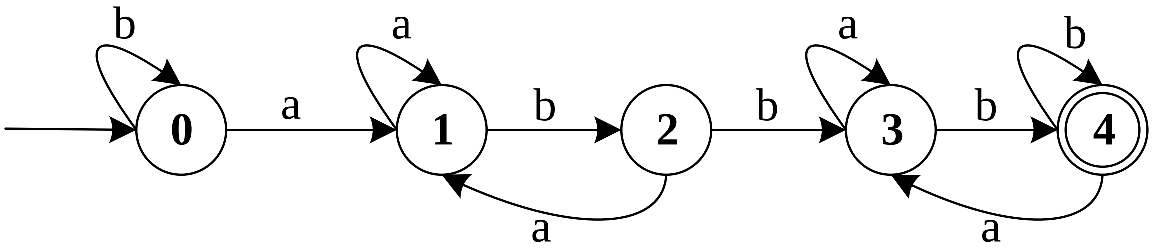


（4） 对该 DFA 进行状态最小化

答：

- {A,B,C,D,E,F} {G,H,I}
- {A,B,C,D} {E,F} {G,H,I}
- {A,B,C}{D}{E,F}{G,H,I}{A,C}(0) {B}(1) {D}(2) {E,F}(3) {G,H,I}(4)

最小化 DFA 如下图：



（5） 用双层 case 嵌套实现（4）中的 DFA

答：代码如下

```
state = 0; // {start}

while(1){
    switch(state){
        case 0:
            ch = nextChar();
            switch(ch){
                case 'a': state = 1;
                case 'b': state = 0;
                else fail();
                break;
            }
        case 1:
            ch = nextChar();
            switch(ch){
                case 'a': state = 1;
                case 'b': state = 2;
                else fail();
                break;
            }
        case 2:
            ch = nextChar();
            switch(ch){
                case 'a': state = 1;
                case 'b': state = 3;
                else fail();
                break;
            }
        case 3:
            ch = nextChar();
            switch(ch){
                case 'a': state = 3;
                case 'b': state = 4;
                else fail();
                break;
            }
        case 4:
            ch = nextChar();
            switch(ch){
                case 'a': state = 3;
                case 'b': state = 4;
                else fail();
                break;
            }
       if state == 4 {
           return(accept);
       }
    }
}
```

2. 构造一个最小 DFA，能够识别被 3 整除的无符号十进制整数

答：类比上课所提到的整除题目，被能够识别被三整除的无符号十进制整数的 DFA 应该有以下三种状态：

1) $\boldsymbol{S_0}$：除 3 余 0 状态；

2) $\boldsymbol{S_1}$：除 3 余 1 状态；

3) $\boldsymbol{S_2}$：除 3 余 2 状态；

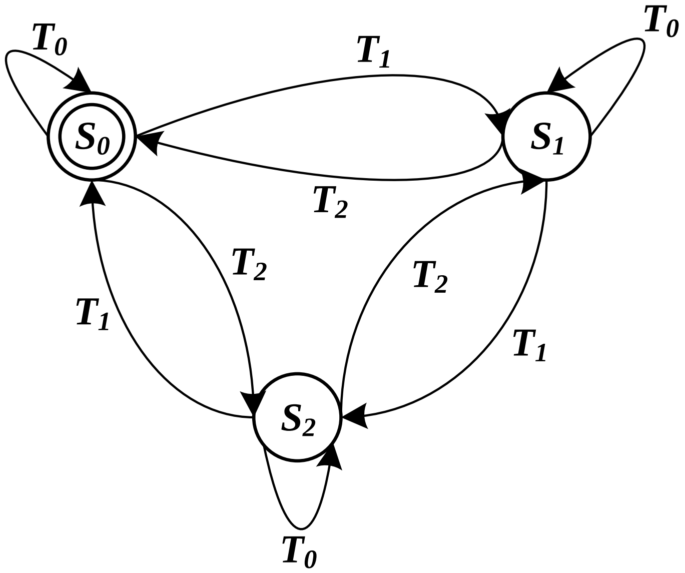
将转移条件分为以下三类：

1) $\boldsymbol{T_0}$：0 或 3 或 6 或 9

2) $\boldsymbol{T_1}$：1 或 4 或 7

3) $\boldsymbol{T_2}$：2 或 5 或 8

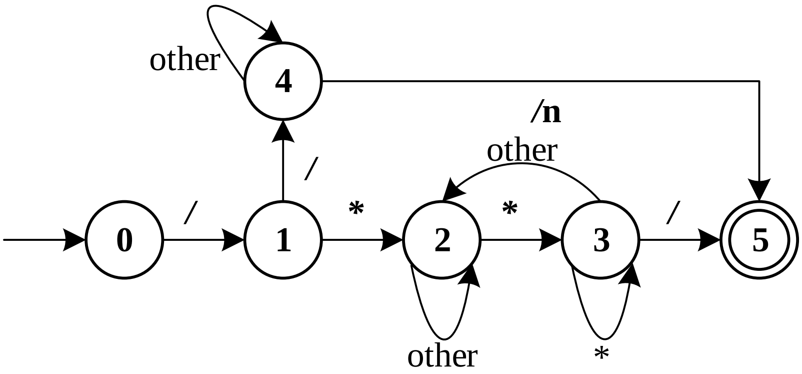
根据以上假设可以得到如下最小 DFA：



3. 编写一个程序，将一个 C 语言源程序中位于注释外的所有保留字全部大写

（1） 构建相应的 DFA

答：如下图



（2） 用表驱动方法实现（1）中的 DFA （写出伪代码）

答：驱动表格如下：

input char / state	'/'	'*'	'\n'	other	Accepting
0	1				no
1	4	2			no
2	2	3	2	2	no
3	5	3	2	2	no
4	4	4	5	4	no
5					yes

```
state := 0;
ch := next input char;
while not Accept[state] and not error(state) do
     newstate := T[state, ch];
     if Advance[state, ch] then
         ch := next input char;
         state := newstate;
end while;
if Accept[state] then accept;
```