第 3 章 词法分析习题答案

1．判断下面的陈述是否正确。

（1）有穷自动机接受的语言是正规语言。（√）

（2）若 r1 和 r2 是Σ上的正规式，则 r1|r2 也是Σ上的正规式。（√）

（3）设 M 是一个 NFA，并且 L(M)={x,y,z}，则 M 的状态数至少为 *4* 个。(×)

（4）设Σ={a,b}，则Σ上所有以 b 为首的符号串构成的正规集的正规式为 *b*(a|b)*。*(×)

（5）对任何一个 NFA M，都存在一个 DFA M’，使得 L(M’)=L(M)。（√）

（6）对一个右线性文法 G，必存在一个左线性文法 G’，使得 L(G)=L(G’)，反之亦然。（√）

（7）一个 DFA，可以通过*多条路*识别一个符号串。(×)

（8）一个 NFA，可以通过多条路识别一个符号串。（√）

（9）如果一个有穷自动机可以接受空符号串，则它的状态图*一定含有*ε边。(×)

（10）DFA 具有翻译单词的能力。(×)

2. 指与出正规式匹配的串.

（1）(ab|b)*c 与后面的那些串匹配？ababbc abab c babc aaabc

（2）ab*c*(a|b)c 与后面的那些串匹配？ acac acbbc abbcac abc acc

（3）(a|b)a*(ba)* 与后面的那些串匹配? ba bba aa baa ababa

答案

（1） ababbc c babc

（2） acac abbcac abc

（3） ba bba aa baa ababa

3. 为下边所描述的串写正规式，字母表是{0, 1}.

(1)以 01 结尾的所有串

(2)只包含一个 0 的所有串

(3) 包含偶数个 1 但不含 0 的所有串

(4)包含偶数个 1 且含任意数目 0 的所有串

(5)包含 01 子串的所有串

(6)不包含 01 子串的所有串

答案

注意 正规式不唯一

(1) (0|1)*01

(2) 1*01*

(3) (11)*

(4) (0*10*10*)*

(5) (0|1)*01(0|1)*

(6) 1*0*

4. 请描述下面正规式定义的串. 字母表{x, y}.

(1) x(x|y)*x

(2) x*(yx)*x*

(3) (x|y)*(xx|yy) (x|y)*

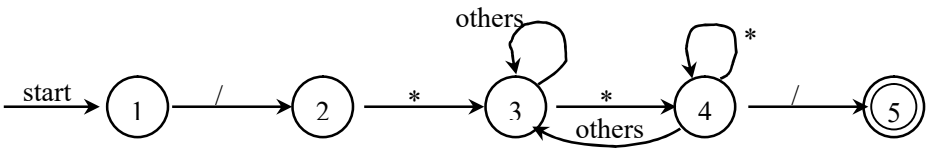
答案

(1)必须以 x 开头和x结尾的串

(2)每个 y 至少有一个 x 跟在后边的串

(3)所有含两个相继的x或两个相继的y的串

5．处于/* 和 */之间的串构成注解，注解中间没有*/。画出接受这种注解的DFA的状态转换图。



答案：见上图。标记为others的边是指字符集中未被别的边指定的任意其它字符。

分析 这个DFA的状态数及含义并不难确定，见下面的五个状态说明。

状态1：注释开始状态。

状态2：进入注释体前的中间状态。

状态3：表明目前正在注释体中的状态。

状态4：离开注释前的中间状态。

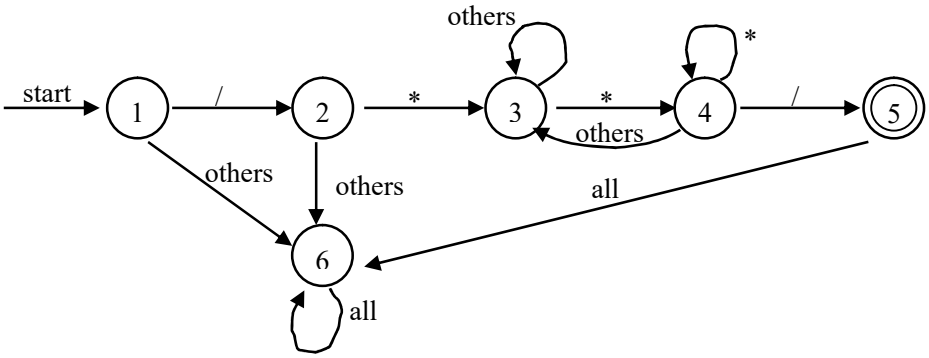
状态5：注释结束状态，即接受状态。

分析：在这个DFA中，最容易忽略的是状态4到本身的’*’转换。这个边的含义是：在离开注释前的中间状态，若下一个字符是’*’，那么把刚才读过的’*’看成是注释中的一个字符，而把这下一个字符看成可能是结束注释的第一个字符。若没有这个边，那么象

/**** This is a comment ****/

这样的注释就被拒绝。

另外，上面的状态转换图并不完整。例如，对于状态1，没有指明遇到其它字符怎么办。要把状态转换图画完整，还需引入一个死状态6，进入这个状态就再也出不去了。因为它不是接受状态，因此进入这个状态的串肯定不被接受。完整的状态转换图见下图，其中all表示任意字符。在能够说清问题时，通常我们省略死状态和所有到它的边。



6．一个C语言编译器编译下面的函数时，报告parse error before ‘else’。这是因为else的前面少了一个分号。但是如果第一个注释

```
        /* then part */
```
误写成
```
        /* then part
```
那么该编译器发现不了遗漏分号的错误。这是为什么？
```
long gcd(p,q)
long p,q;
{
            if (p%q == 0)
            /* then part */
            return q
            else
            /* else part */
            return gcd(q, p%q);
}
```

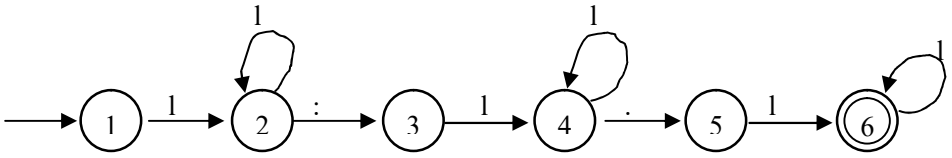
答案：此时编译器认为
```
/* then part
        return q
    else
        /* else part */
```
是程序的注释，因此它不可能再发现else 前面的语法错误。

分析 这是注释用配对括号表示时的一个问题。注释是在词法分析时忽略的，而词法分析器对程序采取非常局部的观点。当进入第一个注释后，词法分析器忽略输入符号，一直到出现注释的右括号为止，由于第一个注释缺少右括号，所以词法分析器在读到第二个注释的右括号时，才认为第一个注释处理结束。

为克服这个问题，后来的语言一般都不用配对括号来表示注释。例如Ada语言的注释始于双连字符（--），随行的结束而终止。如果用Ada语言的注释格式，那么上面函数应写成
```
long gcd(p,q)
long p,q;
{
            if (p%q == 0)
            --  then part
            return q
            else
            -- else part
            return gcd(q, p%q);
}
```

7．某操作系统下合法的文件名为 device:name.extension，其中第一部分（device:）和第三部分（.extension）可缺省，若 device，name 和 extension 都是字母串，长度不限，但至少为 1，画出识别这种文件名的 DFA。（用 l 表示任意字母）。

答案：这个 DFA 和无符号数的 DFA 有类似的地方。首先考虑 device:和.extension 全都出现的情况。（即：device:name.extension）这时的 DFA 比较容易构造。



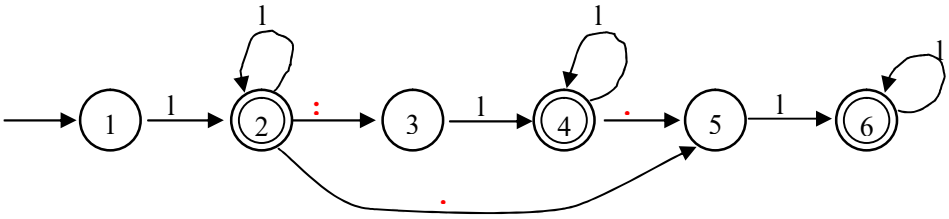
文件名的三部分都出现的 DFA

然后考虑缺省情况：

（1）因为.extension 可缺省（即：device:name），因此把状态 4 也作为接受状态。

（2）因为 name 和 device 一样，都是字母序列，所以在 device:缺省时，把到状态 2 为止得到的字母序列看成是 name。由于 device:和.extension 都可缺省（即： name），因此把状态 2 也作为接受状态。

（3）(即：name.extension)因为 name 和 device 一样，都是字母序列，因此在 device:缺省时，把到状态 2 为止得到的字母序列看成是 name，所以从状态 2 画一条转换边到状态 5，标记为’.’。



接受文件名的 DFA

8．有一台自动售货机，接收 1 分和 2 分硬币，出售 3 分钱一块的硬糖。顾客每次向机器中投放≥3 分的硬币，便可得到一块糖(注意：只给一块并且不找钱)。

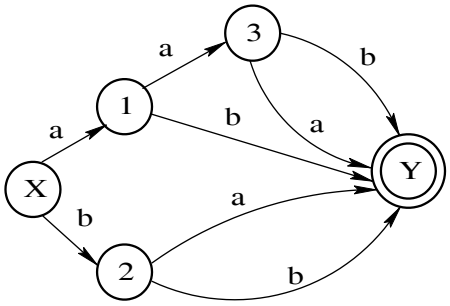
（1）写出售货机售糖的正规表达式；

（2）构造识别上述正规式的最简 DFA。

答案：

（1）设 a=1，b=2，则售货机售糖的正规表达式为 a (b|a(a|b))|b(a|b)。

（2）对应的 DFA 状态图如下：



DFA 最小化的过程如下：

I	I_a	I_b
X	1	2
1	3	Y
2	Y	Y

3	Y	Y
Y	∅	∅

（a）将所有状态分成两个子集：非终态集 S1={X，1，2，3}和终态集 S2={Y}；

（b）考虑非终态集 S1

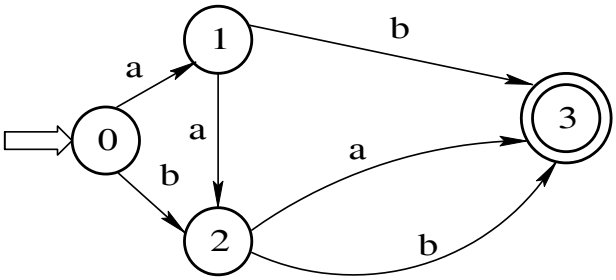
对于输入符号 a，$Ia^{\{X,1\}}=\{1,3\}\subset S1$，$Ia^{\{2,3\}}=\{Y\}\subseteq S2$

对于输入符号 b，$I_b^{\{X\}}=\{2\}\subset S1$ ， $I_b^{\{1\}}=\{Y\}\subset S2$，$I_b^{\{2,3\}}=\{Y\}\subseteq S2$，

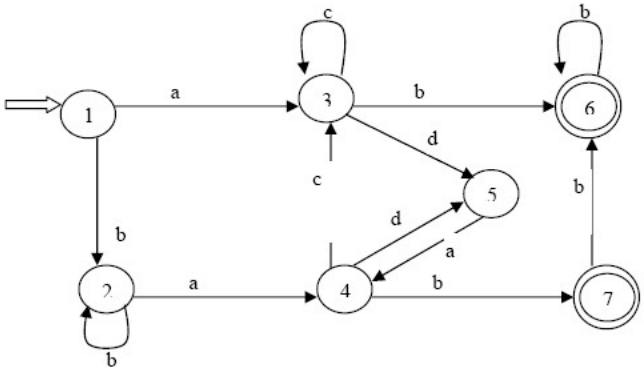
所以子集 S1 中的状态 X 和 1 不等价，状态 2 和 3 等价，从而需要划分为三个集合 S11={X}、S12={1}、S13={2，3}。

（c） 考虑终态集 S2 不需要继续划分。

（d）划分结束，得到四个子集{X}、{1}、{2，3}和{Y}。按顺序重新命名为 0、1、2、3，得到最简 DFA 状态图如下：



9. 将下图的 DFA 最小化，并用正规式描述它所识别的语言。（清华教材 73 页第 9 题）



答案：

DFA 最小化的过程如下：

I	I_a	I_b	I_c	I_d
1	3	2	∅	∅
2	4	2	∅	∅
3	∅	6	3	5

4	∅	7	3	5
5	4	∅	∅	∅
6	∅	6	∅	∅
7	∅	6	∅	∅

（a）将所有状态分成两个子集：非终态集 S1=｛1, 2, 3, 4, 5｝和终态集 S2=｛6, 7｝;

（b）考虑非终态集 S1

对于输入符号 a，$Ia^{\{1,\ 2,\ 5\}}$={3，4}⊂S1 ，$Ia^{\{3,\ 4\}}$=∅

对于输入符号 b，$I_b^{\{1,\ 2\}}$={2}⊂S1，$I_b^{\{3,\ 4\}}$={6，7}⊂S2，$I_b^{\{5\}}$=∅

对于输入符号 c，$I_c^{\{1,\ 2\}}$=∅，$I_c^{\{3,\ 4\}}$={3}⊂S1，$I_c^{\{5\}}$=∅

对于输入符号 d，$I_d^{\{1,\ 2\}}$=∅，$I_d^{\{3,\ 4\}}$={5}⊂S1，$I_d^{\{5\}}$=∅

所以子集 S1 中的｛1,2｝，｛3,4｝，｛5｝不等价，需要划分为三个集合 S11=｛1,2｝、S12=｛3,4｝和 S13=｛5｝。

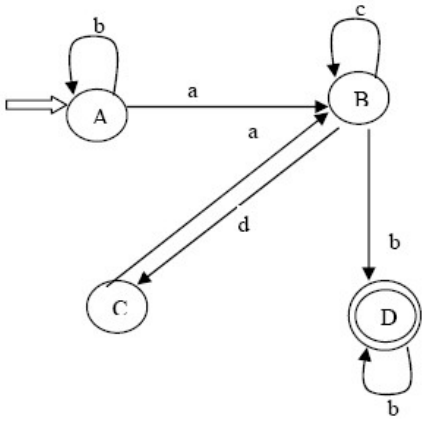
（c）考虑终态集 S2

对于输入符号 b，$I_b^{\{6,7\}}$={6}

对于其他输入符号 a，c，d，$I^{\{6,\ 7\}}$=∅

所以子集 S2 中的状态 6 和 7 等价，不需要继续划分。

（d）划分结束，得到四个子集｛1,2｝，｛3,4｝，｛5｝，｛6，7｝，分别重新命名为新状态 A，B，C，D，得到最小化的 DFA 状态图如下：



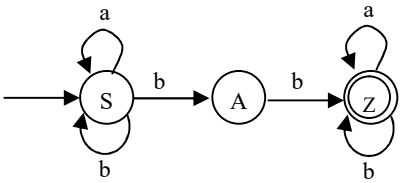
正规式：r=b*a(c|da)*bb*

10. 已知 NFA 如右图所示：

（1）以上状态图所表示的语言有什么特点？

（2）写出表示该语言的正规式。

（3）将 NFA 确定化为 DFA 并最小化。



答案：

（1）L(G)={X|X 是至少含有两个连续 b 的 ab 串}

（2）正规式为：(a|b)*bb(a|b)*

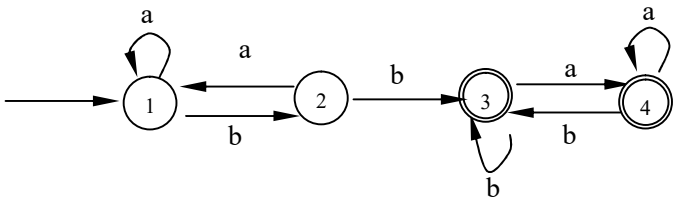
（3）（请按照以下步骤书写求解过程）

① NFA 确定化为 DFA 过程如下表所示：

NFA 确定化过程

状态	Ia	Ib
{S}	{S}	{S，A}
{S，A}	{S}	{S，A，Z}
{S，A，Z}	{S，Z}	{S，A，Z}
{S，Z}	{S，Z}	{S，　A，Z}

将表中第一列的每个状态子集分别用 1,2,3，4 表示，得到新的 DFA 状态图如下：



② DFA 最小化的过程如下：

（a）将所有状态分成两个子集：非终态集 S1={1，2}和终态集 S2={3，4}；

（b）考虑非终态集 S1

对于输入符号 a，$Ia^{\{1,2\}}=\{1\}$

对于输入符号 b，$I_b^{\{1\}}=\{2\}\subset S1$，$I_b^{\{2\}}=\{3\}\subset S2$

所以子集 S1 中的状态 1 和 2 不等价，需要划分为两个集合 S11={1}和 S12={2}。

（c） 考虑终态集 S2

对于输入符号 a，$Ia^{\{3,4\}}=\{4\}\subset S2$

对于输入符号 b，$I_b^{\{3,4\}}=\{3\}\subset S2$

所以子集 S2 中的状态 3 和 4 等价，不需要继续划分。

（d）划分结束，得到三个子集{1}、{2}、{3，4}。将{3，4}重新命名为新状态 3，得到最小化的 DFA 状态图如下：