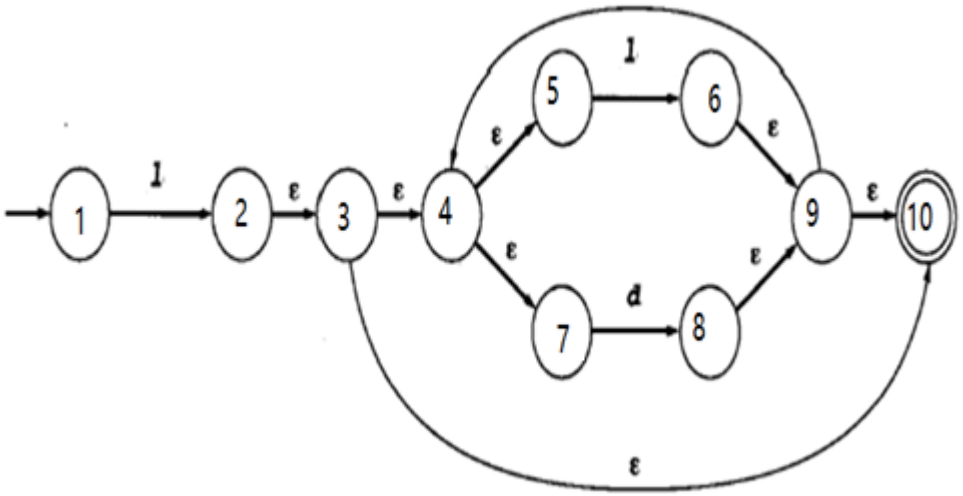
实验二

1.输入一行(一个)或多行(多个)正则表达式：

如：l(l|d)*

2.正则表达式→NFA

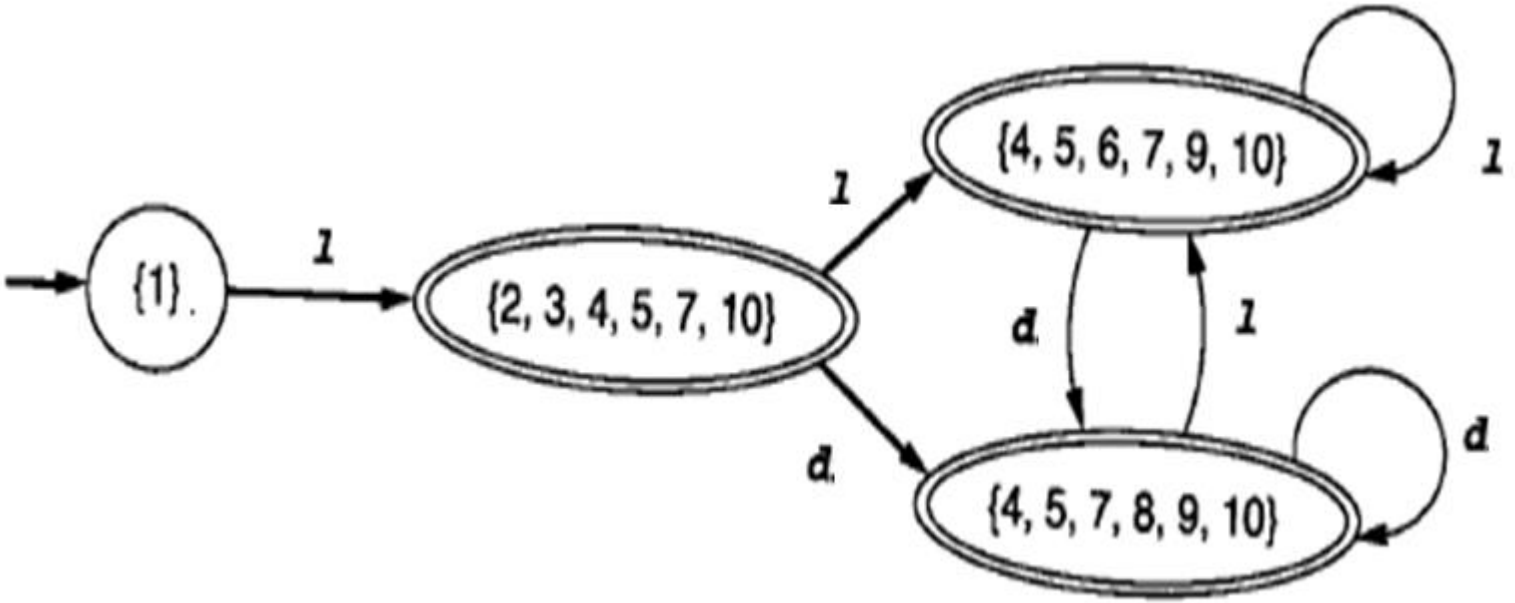
输出：如下图的NFA或该图对应的状态转换表



		l	d	#
→	1	2		
	2			3
	3			4, 10
	4			5, 7
	5	6		
	6			9
	7		8	
	8			9
	9			4, 10
←	10			

3.NFA→DFA

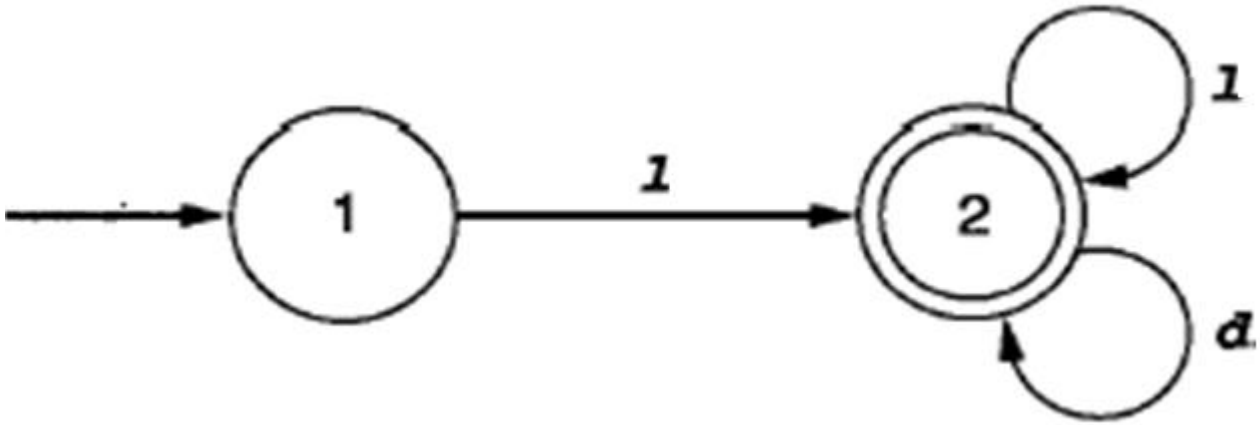
输出：如下图的DFA或对应的状态转换表(如下表)



	字符 / 状态集合	l	d
—	{ 1 }	{ 2, 3, 4, 5, 7, 10 }	
✚	{ 2, 3, 4, 5, 7, 10 }	{ 6, 9, 4, 7, 10, 5 }	{ 8, 9, 4, 7, 5, 10 }
✚	{ 6, 9, 4, 7, 10, 5 }	{ 6, 9, 4, 7, 10, 5 }	{ 8, 9, 4, 7, 5, 10 }
✚	{ 8, 9, 4, 7, 5, 10 }	{ 6, 9, 4, 7, 10, 5 }	{ 8, 9, 4, 7, 5, 10 }

4.DFA最小化

输出：如下图的最小化DFA或对应的状态转换表

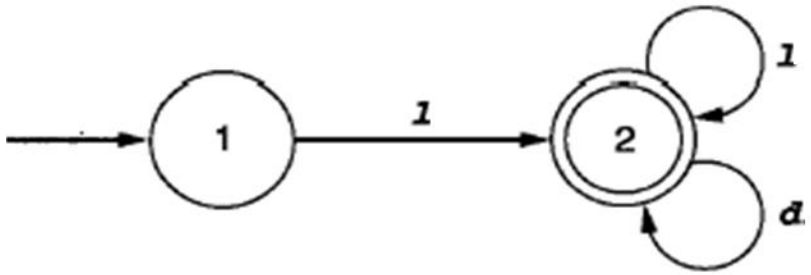


		l	d
−	1	2	
+	2	2	2

5.DFA→词法分析程序【注意：实验要求生成的分析程序是C/C++语言的表达方法，而非下面样例的类程序语言】

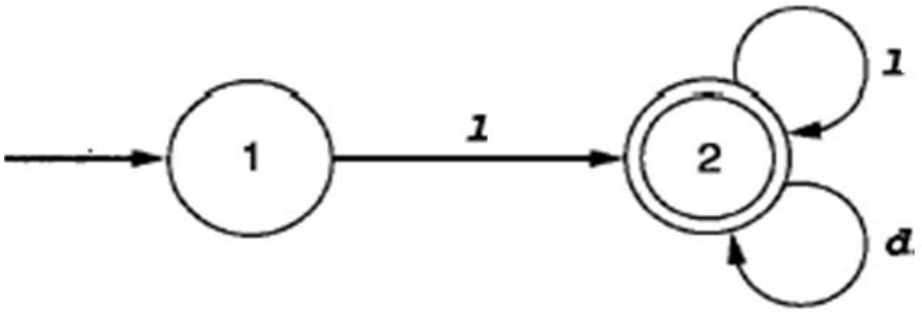
输出：生成对应的词法分析程序(方法一)。

```
{ starting in state 1 }
if the next character is a letter then
    advance the input;
   { now in state 2 }
   while the next character is a letter or a digit do
      advance the input; { stay in state 2 }
   end while;
   { go to state 3 without advancing the input }
   accept ;
else
   { error or other cases }
end if;
```



输出：生成对应的词法分析程序(方法二)

```
state := 1; { start }
while state = 1 or 2 do
  case state of
  1:  case input character of
      letter : advance the input;
               state := 2;
      else state := . . . { error or other };
      end case;
  2:  case input character of
      letter, digit: advance the input;
                     state := 2; { actually unnecessary }
      else state := 3;
      end case;
  end case;
end while;
if state = 3 then accept else error ;
```



实验需要提交的内容及注意事项

- 第2次实验作业的提交，只能使用RAR文件或ZIP压缩文件。
- 压缩文件内含文件夹及文件如下：
- （1）源程序文件夹：内含整个实验2的所有源程序文件和编译方法的说明介绍文件
- （2）文档文件夹：内含实验2的设计文档（PDF或DOC格式）（注：文档书写格式可参考百度云盘中课程实验文件夹下的格式）
- （3）测试数据文件夹：内含所有的测试数据文件(应该能测试所支持的每个运算符)和测试结果的汇报文件
- （4）可执行程序文件夹：内含实验2的可执行程序以及使用说明书。