

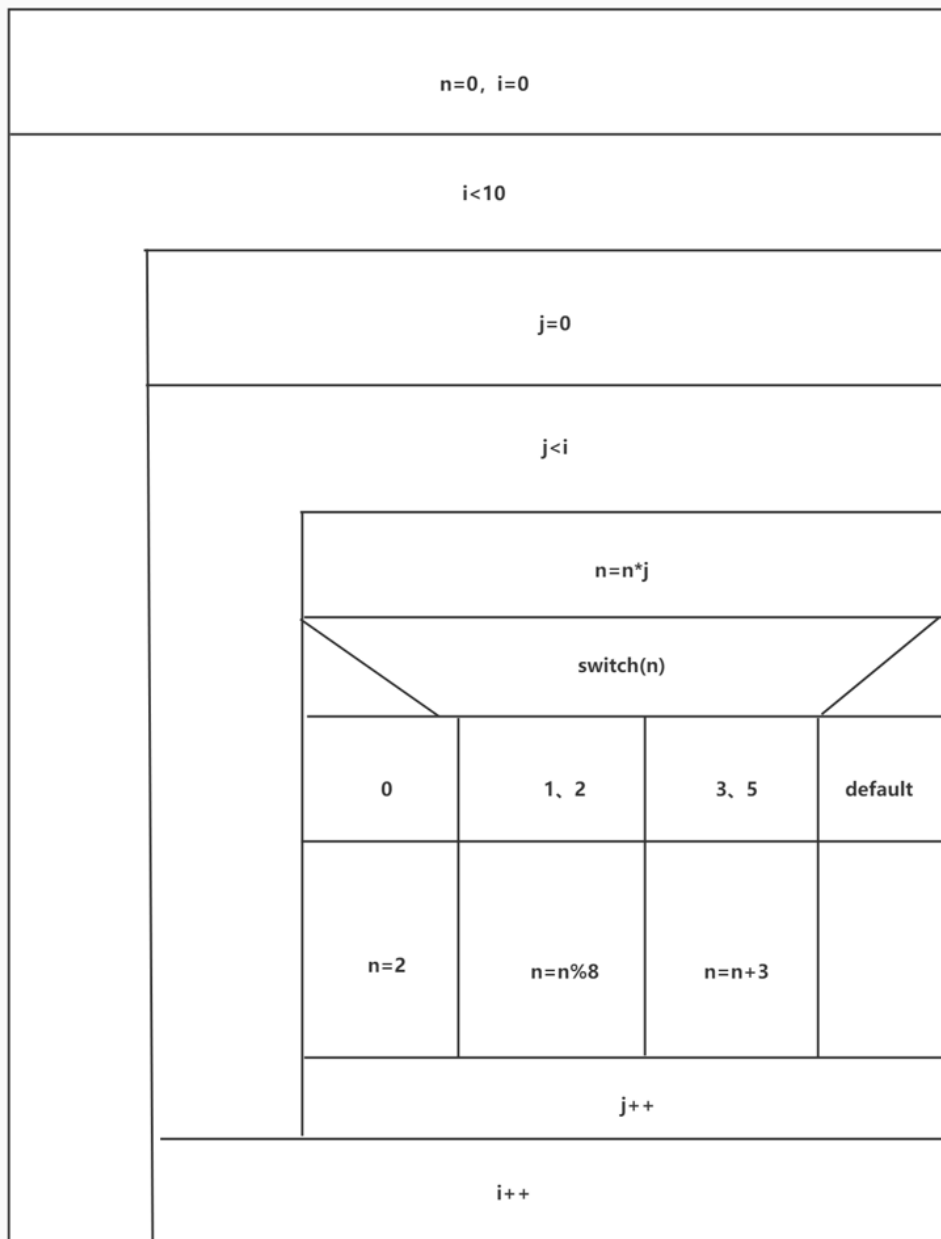
考察知识点：N-S图、状态转换图

1. 给出下面代码段的程序盒图，并使用状态转换图对该代码段中变量 **n** 的状态变化进行描述。

```
n=0;
for (i=0; i<10; i++)
    for (j=0; j<i; j++){
        n=n*j;
        switch(n){
            case 0: n=2; break;
            case 1:
            case 2: n=n%8; break;
            case 3:
            case 5: n=n+3; break;
            default: break;
        }
    }
}
```

(1)N-S图

[看不清楚图片，请戳我!](#) 

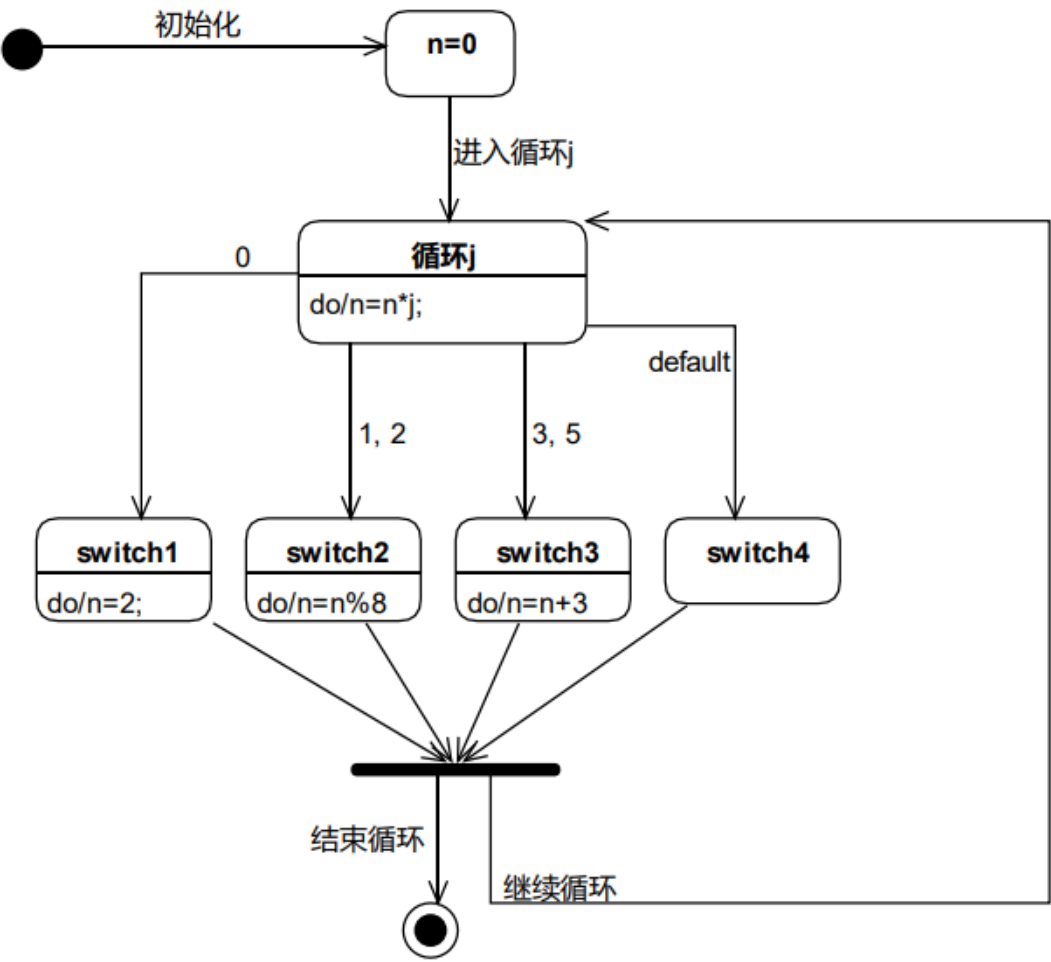


(2)状态转换图

do事件指定在该状态下的动作。

个人认为不会出如下图所示的状态转换图习题，因此直接给出参考答案。

状态转换图：



考察知识点：面向对象、状态图的理解

4. 图 11.16 所示是某杂志社稿件处理系统中稿件类 (Article) 的状态图, 根据该图回答问题。

(1) 给出 Article 类的定义, 包括属性和方法。

(2) 列出所有需要测试的类状态。

(3) 列出所有需要测试状态的转换。

(4) 从初始态开始 (各属性置为空值), 一篇名为 *A Good Paper* 的稿件投稿, 由于符合杂志领域范围, 通过初审, 然后送外审, 外审专家认为质量不错, 但需要继续修改, 论文修改后达到要求, 被录用。为以上场景开发一个测试驱动类, 编写代码。

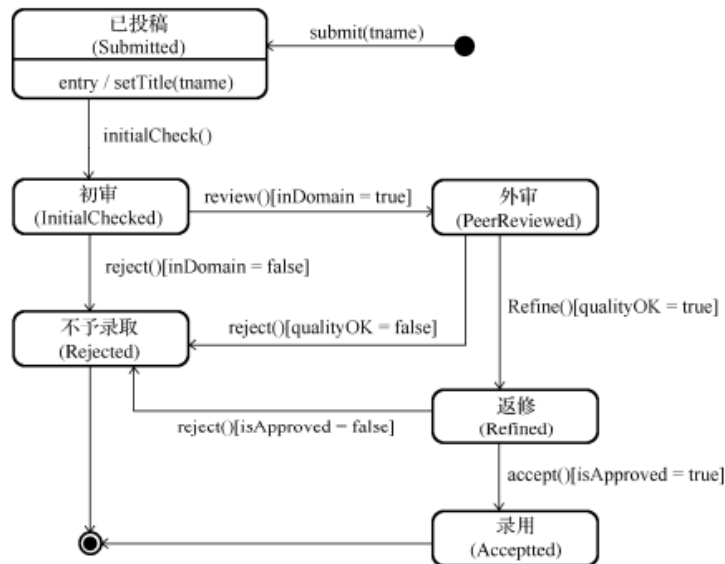


图 11.16 稿件类的状态图

参考解答:

(1)

根据稿件类的状态图可知Article类的属性有tname,inDomain,qualityOK,isApproved;方法有submit()、initCheck()、review()、reject()、Refine()、accept()故Article类的定义【Java实现】如下:

```
public class Article{
    private String tname;
    private boolean inDomain;
    private boolean qualityOK;
    private boolean isApproved;

    // 无参构造方法、有参构造方法
    // get、set方法此处省略

    public void submit(String tname){
        this.tname=tname;
    }

    public void initCheck(){

    }

    public void review(){

    }

    public void reject(){
```

```

    }

    public void Refine(){

    }

    public void accept(){

    }
}

```

(2)所有需要测试的类状态:

Submitted、InitialChecked、PeerReviewed、Rejected、Refined、Accepted

(3)所有需要测试状态的转换

| 状态A | 事件 | 状态B |
|----------------|----------------|----------------|
| Submitted | initialCheck() | InitialChecked |
| InitialChecked | review() | PeerReviewed |
| | reject() | Rejected |
| PeerReviewed | Refine() | Refined |
| | reject() | Rejected |
| Refined | accept() | Accepted |
| | reject() | Rejected |

(4)属于后续章节的内容，故可忽略

参考答案如下图所示:

(4)

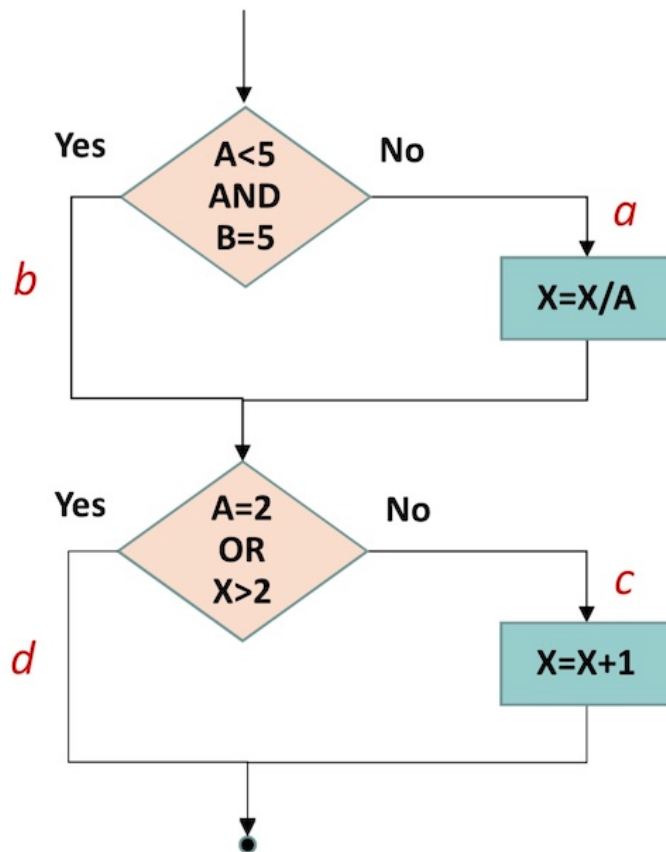
```

public class ArticleTest {
    public static void main(String[] args) {
        Article article = new Article();
        article.inDomain = true;
        article.qualityOK = true;
        article.isApproved = true;

        article.submit("A Good Paper");
        article.initialCheck();
        article.review();
        article.refine();
        article.accept();
        return;
    }
}

```

考察知识点：基本路径测试



- 根据程序流程图

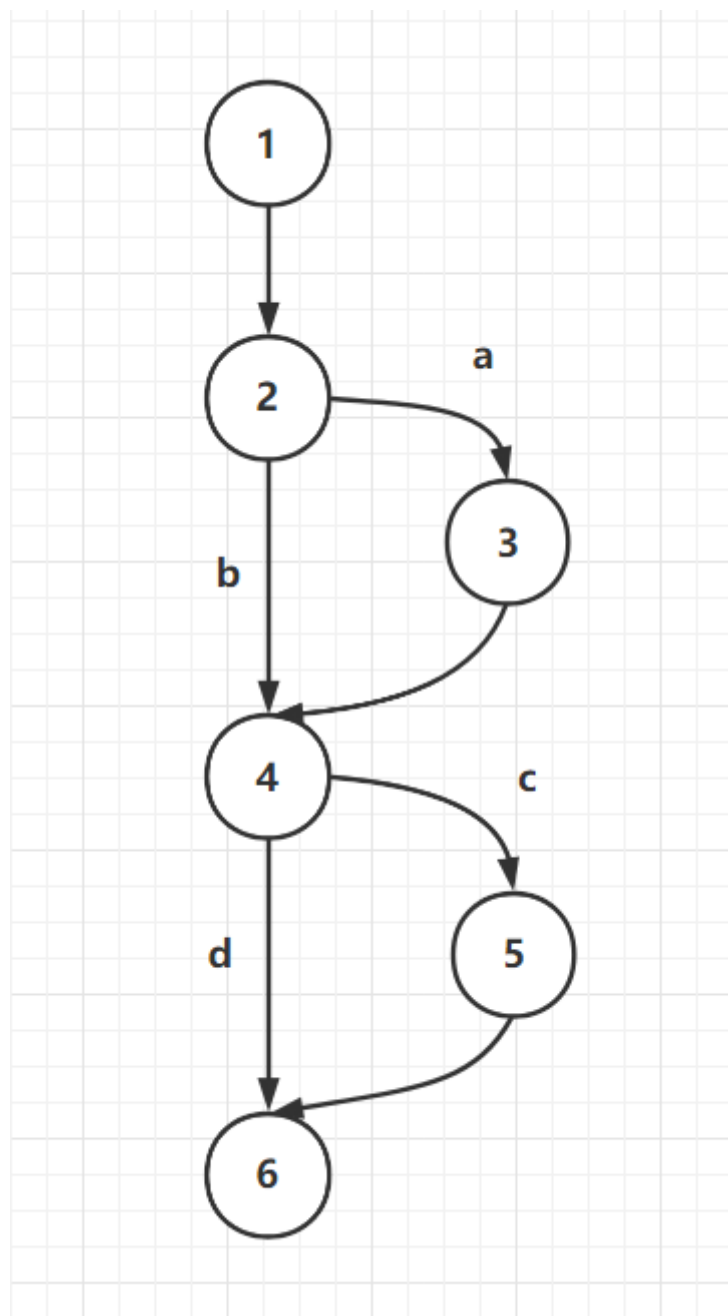
- (1) 画出相应的程序控制流图;
- (2) 计算 McCabe 环形复杂度;
- (3) 找出程序的一个独立路径集合;
- (4) 根据独立路径设计测试用例。

(1) 程序控制流图:

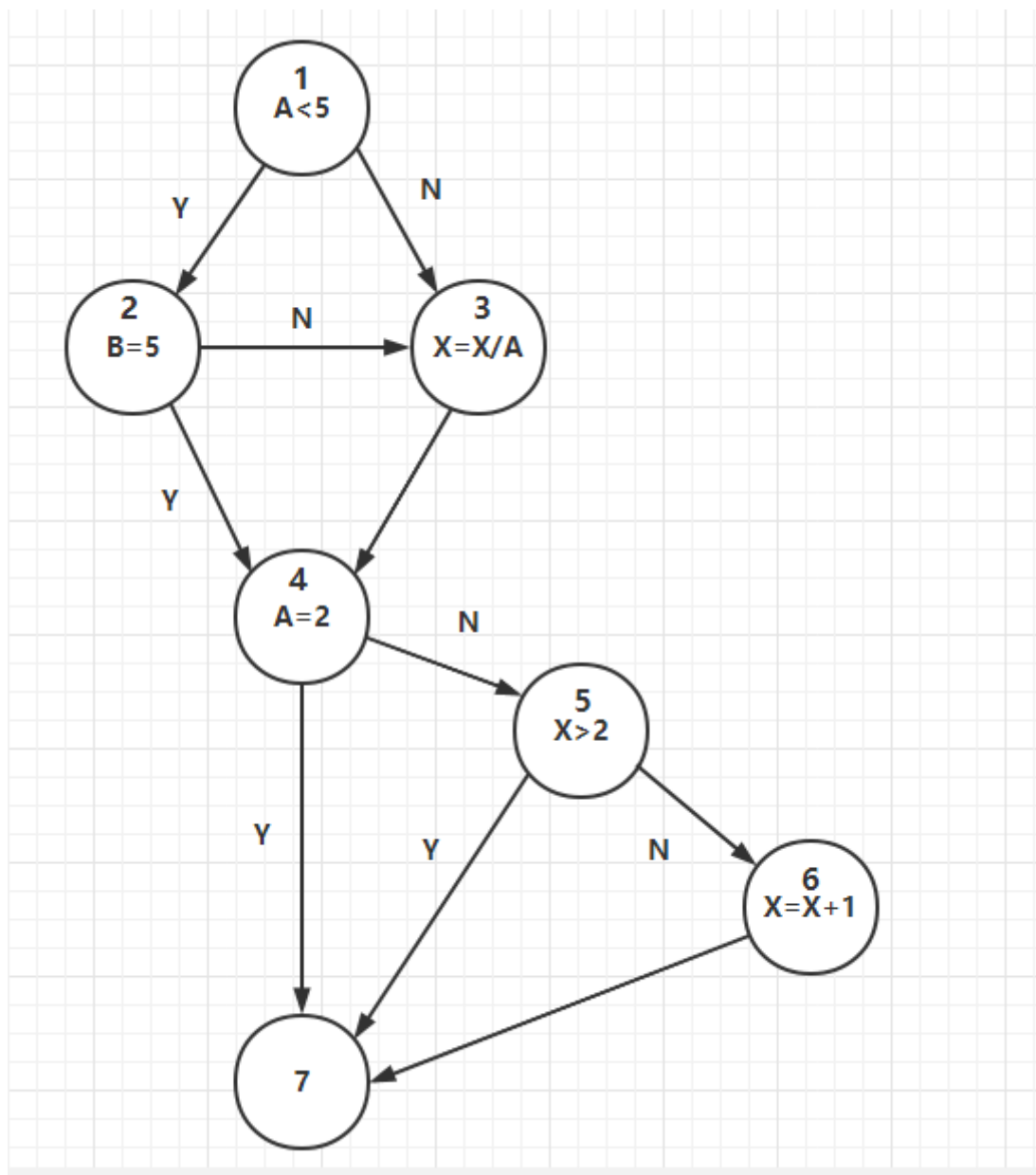
当我做到这道题目的时候，跟复习题(2)的第1题进行了比较，最后纠正了复习题(2)的第1题流图绘制的错误。与此同时，当我再次翻看流图绘制的讲解的时候，又发现了其中的一个坑点。

当过程设计中包含复合条件时，生成流图的方法会稍微复杂一些。所谓复合条件，就是在条件中包含了一个或多个布尔运算符(逻辑OR, AND, NAND, NOR)。在这种情况下，应该把复合条件分解为若干个简单条件，每个简单条件对应流图中一个结点。.....(书本P137、P139图6.17)

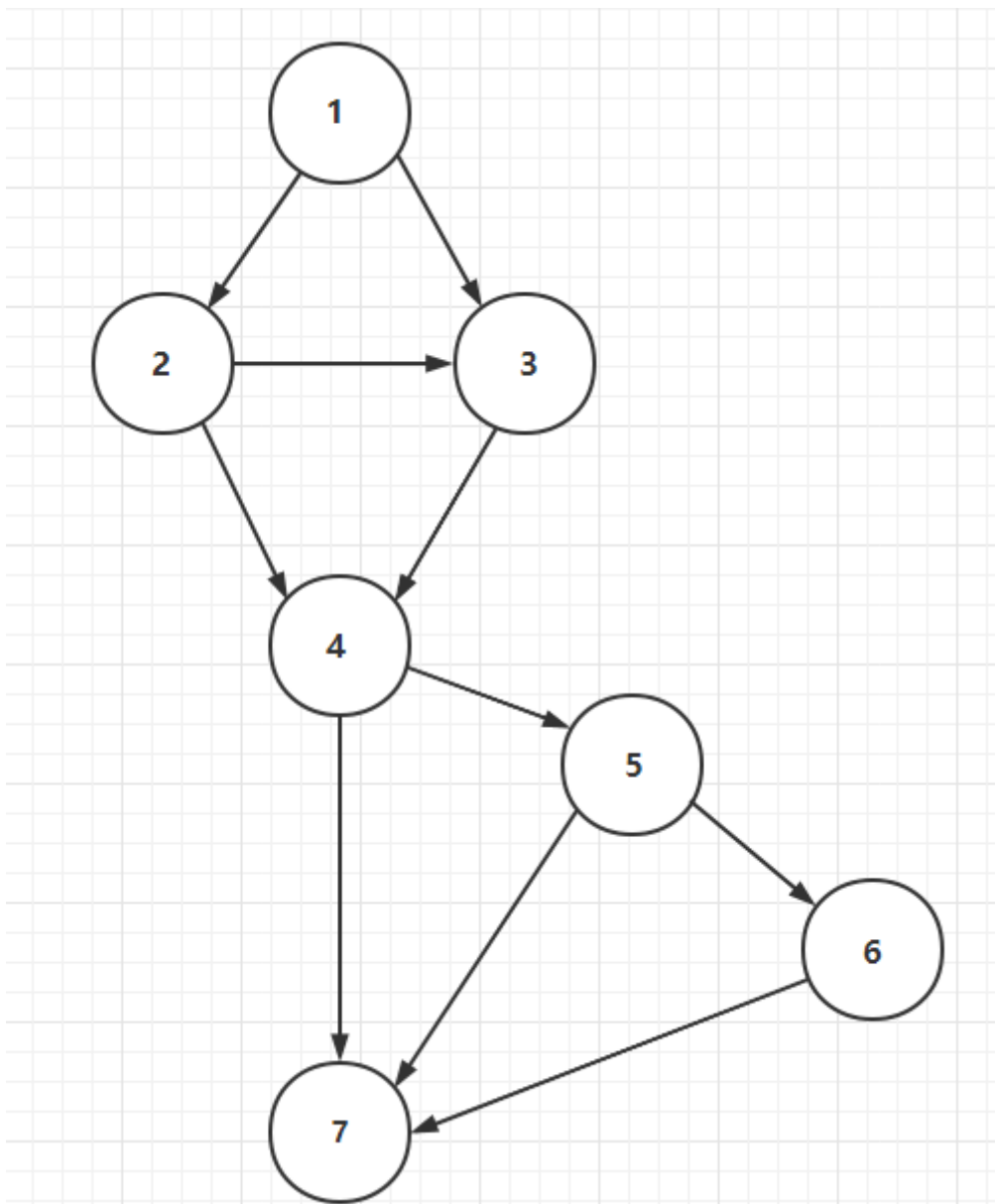
错误画法:



正确画法：



可以先画出上图，接着再将结点内的判定条件以及边上的逻辑值去除，便可得到规范的流图，具体如下图所示：



(2)

环形复杂度=流图中边的条数-流图中结点数+2

流图中边的条数为10，流图中结点数为7

∴ 环形复杂度=10-7+2=5

【或者环形复杂度=流图中判定结点的个数+1，

流图中判定结点的个数为4，分别为编号为1、2、4、5的结点

∴ 环形复杂度=4+1=5】

(3) 程序的一个独立路径的集合

由于环形复杂度为5，所以找出5条独立路径。

路径1: ①—②—④—⑦

路径2: ①—②—④—⑤—⑦

路径3: ①—②—④—⑤—⑥—⑦

路径4: ①—③—④—⑤—⑦

路径5: ①—②—③—④—⑦

补充：有同学可能会有疑问，路径不还有

- 1) — 3) — 4) — 7)
- 1) — 3) — 4) — 5) — 6) — 7)
- 1) — 2) — 3) — 4) — 5) — 7)
- 1) — 2) — 3) — 4) — 5) — 6) — 7)

嘛？

答：

3、基本路径集

- 程序的环路复杂度给出了程序基本路径集中的独立路径条数。
- 从控制流图来看，一条独立路径是至少包含有一条在其它独立路径中从未有过的边的路径。

一般情况下，由短到长
或，由关键路径再逐步扩充
每次增加一条尚未走过的边

但是，不可达
前后有关联性的判断时需结合条件构造基本路径

上述的路径没满足独立路径的定义，至少包含有一条在其它独立路径中从未有过的边的路径。

- 1) — 3) — 4) — 7) 的边在路径4和路径1中均出现过了
- 1) — 3) — 4) — 5) — 6) — 7) 的边在路径4和路径3中均出现过了
- 1) — 2) — 3) — 4) — 5) — 7) 的边在路径5和路径2中均出现过了
- 1) — 2) — 3) — 4) — 5) — 6) — 7) 的边在路径5和路径3中均出现过了

程序的环形复杂度决定了程序中独立路径的数量，而且这个数是确保程序中所有语句至少被执行一次所需的测试数量的上界【环形复杂度=独立路径数】

(4) 根据独立路径设计测试用例

| 路径名 | 输入数据 | 预期结果 |
|-----|----------------|---------------|
| 路径1 | A=2, B=5, X=1 | A=2, B=5, X=1 |
| 路径2 | A=3, B=5, X=4 | A=3, B=5, X=4 |
| 路径3 | A=3, B=5, X=2 | A=3, B=5, X=2 |
| 路径4 | A=5, B=1, X=15 | A=5, B=1, C=3 |
| 路径5 | A=2, B=4, X=8 | A=2, B=4, X=4 |

考察知识点：状态转换图(状态图)

2. IP 长途卡的使用过程如下：未使用前，卡密封在塑料卡袋中，密码被遮挡住；一旦开始使用，使用者必须打开卡袋、刮开密码，通过座机绑定使用长途卡；一旦卡上余额不足或者超出长途卡的使用日期范围，长途卡将无法使用。根据以上描述，分析 IP 长途卡的状态，绘制其状态图。

参考解答：

IP长途卡的状态有：未使用、使用、无法使用

引起"未使用状态->使用状态"的事件：打开卡密、刮开密码；进入使用状态的动作是通过座机绑定

引起"使用状态->无法使用状态"的事件：余额不足、超出长途卡的使用日期范围

注：

在状态图中，初态用实心圆表示，终态用一对同心圆(内圆为实心圆)表示。

中间状态用圆角矩形表示，可以用两条水平横线将它分成上、中、下3个部分。

上面部分为状态的名称，这部分是必须有的；中间部分为状态变量的名字和值，

这部分是可选的；下面部分是活动表，这部分也是可选的。

状态图：

看不清楚图片，请戳我！ 

