# 因果图法

1. 引入

自动售货机的部分需求规格说明如下：

一个处理单价为5角钱的饮料自动售货机。其规格说明如下：若投入5角钱或1元的硬币，按下【橙汁】或【啤酒】的按钮，则相应的饮料就送出来。

若售货机没有零钱找，则一个显示【零钱找完】的红灯亮，这时候投入的1元硬币按下按钮后，饮料不送出而且1元硬币也退出来

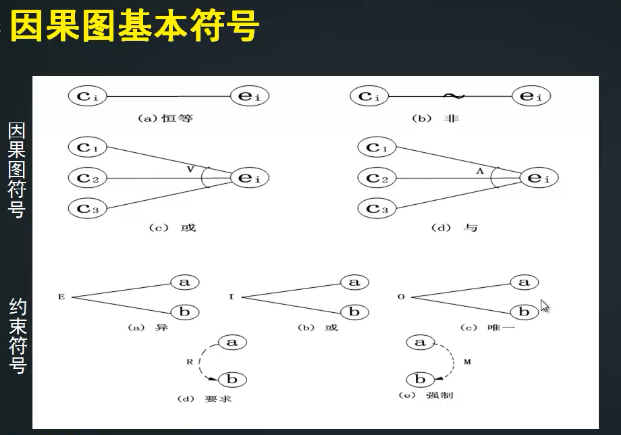
若有零钱找，则显示【零钱找完】的红灯灭，在送出饮料的同时退出5角硬币。

请使用因果图法设计相应的测试用例

1. 因果图法的定义

* 因果图提供了一个把（复杂逻辑关系）规格转化为判定表的系统方法，从该图中可以产生测试数据。其中，原因表示输入条件，结果是对输入执行一系列计算后得到的输出
* 因果图法最终生成的就是判定表。它适用于检查软件输入条件的各种组合情况
* 所以把因果图、判定表归结为一种方法。

1. 因果图的基本符合、因果关系



因果图法主要知道因果逻辑关系

第一类

因果关系（输入和输出之间的逻辑关系）

* 恒等（用横线表示）：如果满足条件a，则输出结果b

例如：如果是ts10班同学，1、3、5在7教室上课

* 非（横线上画~线）：如果满足条件a,则不能输出结果b

如果是ts10班同学，2、4、6不能在7教室上课

* 与：多个输入同时满足，才能得到某个结果

例如毕业要求：a条件：学历大专；b条件三个阶段考试合格；C考勤率90%

a,b,c三个条件同时满足则准予毕业。

* 或：多个输入条件只要有一个或一个以上得到满足就可以得到某个结果

例如入公司招聘：a条件会python语言、b条件会java、c条件会shell脚本语言

只要三种语言会一种可被录取

第二类

原因之间的关系（输入和输入之间的制约关系）

* E （exclusive互斥）的关系：多个输入至多有一个为真，可以同时不为真

比如拍卖会

* I（inclusive包容）的关系：多个输入至少有一个为真，可以同时为真，但不可以同时为假

比如自习室

* O（only唯一）的关系：多个输入有且只有一个为真

比如我们生日

* R（require要求）的关系：a、b两个条件，a为真则要求b为真，a为假则要求b为假

比如早教机构：a小盆友 b家长

a小盆友到早教机构上课则要求b家长陪护

a小盆友不到早教机构上课则要求b也不能来

第三类

结果之间的关系（输出和输出之间的制约关系）

M的关系（强制关系）：输出a、b 如a为真则强制b为假；a为假对b不做强制

比如：小明a；小亮b

小明去教室，小亮则不能去

小明不去教室，小亮可以去也可以不去

1. 因果图的使用步骤

步骤

1. 把大的系统规格划分成可以测试的规格片段
2. 分析规格片段，找出哪些是原因（输入），哪些是结果（输出）
3. 画出因果图（包括原因和结果、原因和原因、结果和结果）删除符合逻辑的关系
4. 把因果图转化为判定表
5. 简化判定表
6. 用判定表中的每一项生成测试用例

* 例题：

某文件修改需求：

如果想对文件进行修改，必须遵守以下规则：

输入的第一列字符必须是A或B；

输入的第二列字符必须是一个数字；

如果第一列字符不正确，则给出信息L;

如果第二列字符不正确，则给出信息M;

如果两列字符输入正确，则修改文件。

xx

蓝色第一列A/B

红色第二列（0~9）

原因： 第一列是A吗？ c1

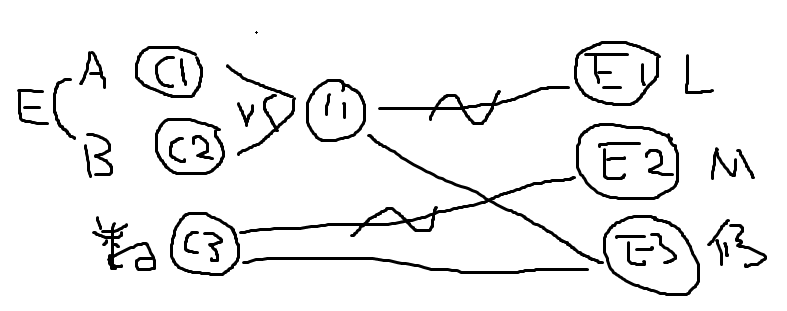
第一列是B吗？ c2

第二列是数字吗？ c3

结果： 信息L E1

信息M E2

修改文件 E3





5．因果图法的应用

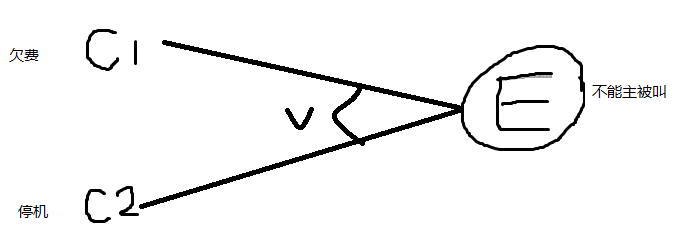
1）手机欠费或停机则不能被主被叫

原因1：欠费 取值：1欠费 0不欠费

原因2：停机 取值：1停机 0不停机

原因1与原因2是I（inclusive包容）的关系

结果：不能主被叫





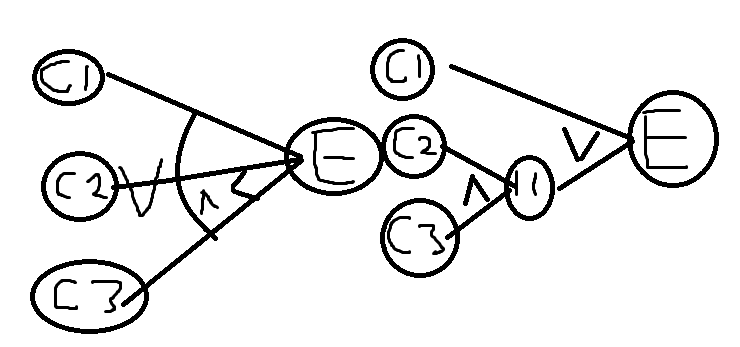
2）有一个需求描述如下“……对于运行10年以上的机器、或功率大于50马力且维修记录不全的机器，给予全面维修，对于其他机器只进行一般维修处理“

原因1：10年以上 10年以上：1 10年及以下：0

原因2：50马力以上 50马力以上：1 50马力及以下：0

原因3：维修记录不全 维修记录不全 ：1 维修记录全：0

结果：全面维修



中间结点作用:

* 当多个原因之间不是单纯的与或关系，我们可以利用中间几点存储中间结果，使得因果逻辑看上去更清晰。
* 当多个原因都在描述同一件事情时，我们可以利用中间节点归并逻辑。



例子3：修改Notes账户密码，要求如下，首先输入正确的原始密码;输入两次一致的新密码；并且新密码具有一定的复杂度（8~15位；包含大写字母、小写字母、数字、其它符号）

原因1 原密码 1正确 0不正确

原因2 复杂度 1够 0不够

原因3 密码是否一致 1一致 0不一致

结果：密码修改成功

6．因过图的优缺点

优点：

等价类法尽管各个输入条件可能出错的情况都考虑到的，但是多个输入条件组合起来出错的情况确被忽略了。

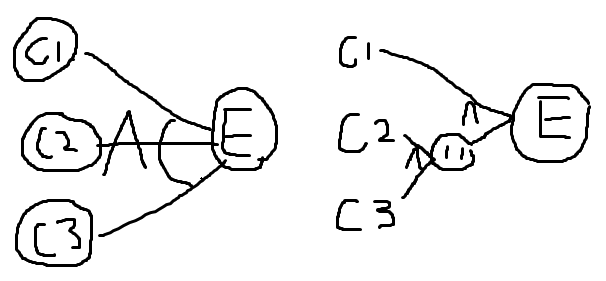
因果图法能够帮助我们按照一定步骤，高效的选择测试用例，设计多个输入条件组合用例

因果图分析还能为我们指出，程序规格说明描述中存在什么问题

缺点：

输入条件与输出结果的关系，有时难以从软件需求规格说明书得到

即使得到了这些因果关系，也会因为关系复杂导致因果图庞大，测试用例数目及其庞大。



例子4，自动售货机的部分需求规格说明如下：

一个处理单价为5角钱的饮料自动售货机。其规格说明如下：若投入5角钱或1元的硬币，按下【橙汁】或【啤酒】的按钮，则相应的饮料就送出来。

若售货机没有零钱找，则一个显示【零钱找完】的红灯亮，这时候投入的1元硬币按下按钮后，饮料不送出而且1元硬币也退出来

若有零钱找，则显示【零钱找完】的红灯灭，在送出饮料的同时退出5角硬币。

原因和原因之间的制约关系，通常是原因描述同一件事的时候。

投一元和投5角是E（互斥的关系）

选饮料：选橙汁和选啤酒也是E（互斥的关系）

1) 分析这一段说明，列出原因和结果

原因：

1.售货机有零钱找

2.投入1元硬币

3.投入5角硬币

4.押下橙汁按钮

5.押下啤酒按钮

结果：

21.售货机〖零钱找完〗灯亮

22.退还1元硬币

23.退还5角硬币

24.送出橙汁饮料

25.送出啤酒饮料

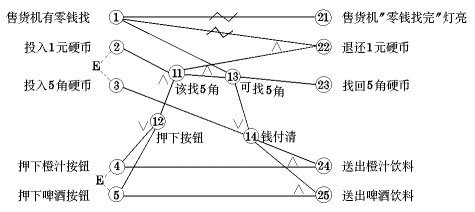
2)画出因果图，如图所示。所有原因结点列在左边，所有结果结点列在右边。建立中间结点，表示处理的中间状态。中间结点：

11. 投入1元硬币且押下饮料按钮

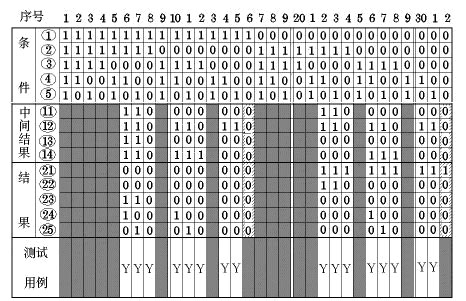
                12. 押下〖橙汁〗或〖啤酒〗的按钮

                13. 应当找5角零钱并且售货机有零钱找

                14. 钱已付清



3)转换成判定表：



4) 在判定表中，阴影部分表示因违反约束条件的不可能出现的情况，删去。第16列与第32列因什么动作也没做，也删去。最后可根据剩下的16列作为确定测试用例的依据。