软件测试技术

主讲 黄轶文

第8个教学模块

黑盒测试技术

——因果图法

上节知识回顾

等价类划分法的应用步骤？

1、列出等价类划分表
1）有效等价类
2）无效等价类
标上编号
2、根据等价类表设计测试用例，覆盖所有的 等价类编号

边界值分析法

1、除了刚刚等于的边界，还有刚刚小于，刚刚大于的边界值
2、4n+1的推论个数，没有包含无效的边界值，在设计测试用例时要注意加上。
3、边界值分析法，是等价类划分法的好朋友，常常搭配使用。边界值确定了，那么等价类也就确定了。Nextdate的例子。

1、等价类划分就是将输入数据按照输入需求划分为若干个子集，这些子集称为（　　）。　　等价类

2、等价类划分法可将输入数据划分为（）和（）。

有效等价类　　无效等价类

3、（）通常作为等价类划分法的补充。

边界值分析法

2.3　因果图法

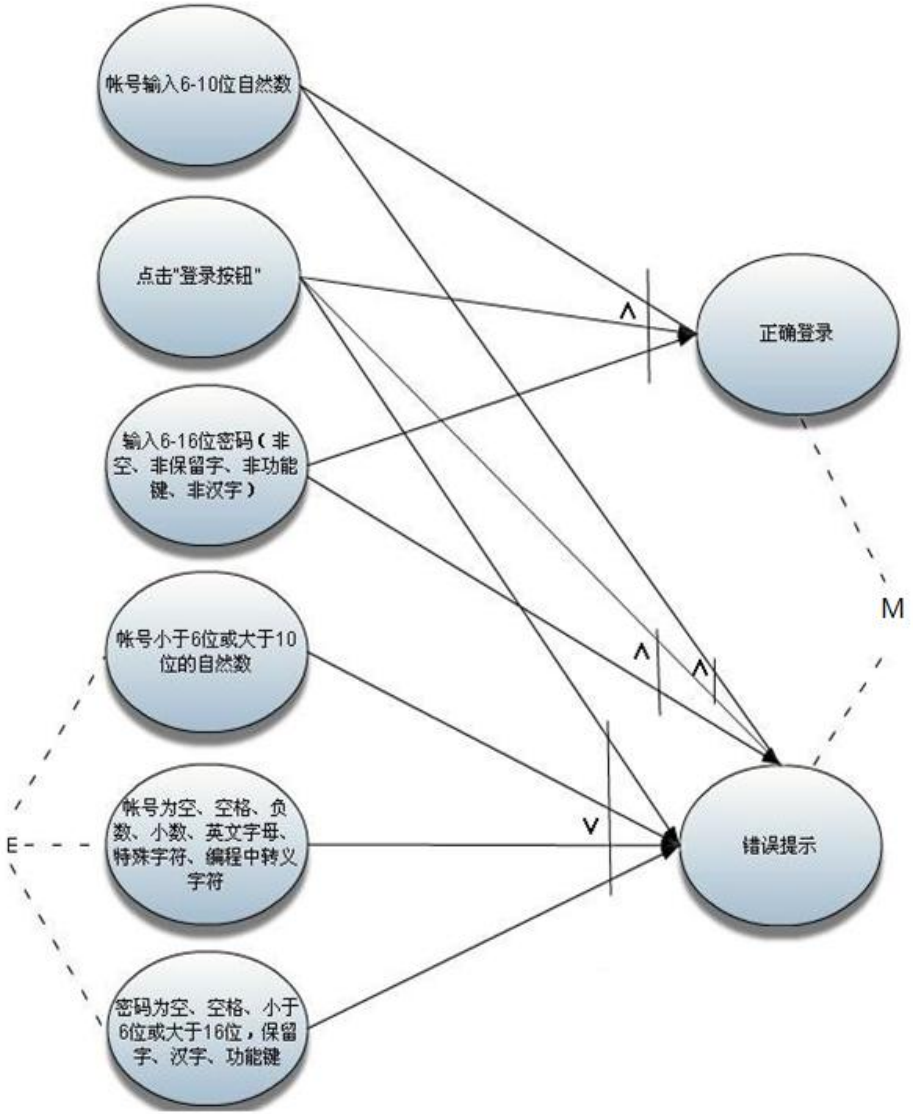
2.3.1因果图法产生的背景

等价类划分法和边界值分析方法都是着重考虑输入条件，但没有考虑输入条件的各种组合、输入条件之间的相互制约关系。这样虽然各种输入条件可能出错的情况已经测试到了，但多个输入条件组合起来可能出错的情况却被忽视了。

因果图方法设计QQ登录界面的测试用例

我们看到有3个可以组合的项：QQ的帐号、QQ的密码、登录按钮。

第一步：画出因果图：


帐号输入6-10位自然数
点击"登录按钮"
输入6-16位密码（非空、非保留字、非功能键、非汉字）
帐号小于6位或大于10位的自然数
帐号为空、空格、负数、小数、英文字母、特殊字符、编程中转义字符
密码为空、空格、小于6位或大于16位，保留字、汉字、功能键
正确登录
错误提示
^
^
^
V
E
M

第二步：从因果图导出判定表：

		1	2	3	4	5	6	7
原因	输入6-10位自然数帐号	1	1	0	0	0	0	0
	点击"登录"按钮	1	1	1	1	1	1	1
	输入6-16位密码（非空、非保留字、非功能键、非汉字）	1	0	1	0	0	0	0
	输入<6的帐号或>10的帐号	0	0	0	0	1	0	0
	帐号为空、空格、负数、小数、英文字母、特殊字符、编程中转义字符	0	0	0	0	0	1	0
	密码为为空、空格、小于6位或大于15位，保留字、汉字、功能键	0	0	0	0	0	0	1
结果	正确登录	1	0	0	0	0	0	0
	错误提示	0	1	1	1	1	1	1

第三步：从判定表导出测试用例：

用例编号	用例操作	输入数据	预期结果
qq-001	输入QQ帐号，输入QQ密码，点击"登录"按钮	532666666 123abc!@#	正确登录
qq-002	输入QQ帐号，点击"登录"按钮	432298987	错误提示
qq-003	输入QQ密码，点击"登录"按钮	abcde123@%$	错误提示
qq-004	点击"登录"按钮		错误提示
qq-005	输入QQ帐号，点击"登录"按钮	23243或635343634363543635	错误提示
qq-006	输入QQ帐号，点击"登录"按钮	hunf@#$\n*-87	错误提示
qq-007	输入QQ密码，点击"登录"按钮	中国beijingAUX/\&#	错误提示

2.3.1测试任务

任务： 为一个自动饮料售货机的软件设计测试用例。为简化问题，假设售货机所有饮料的价格都是5角钱。其规格说明如下。

“若投入5角钱或1元钱的硬币，按下‘橙汁’或‘啤酒’按钮，则相应的饮料就送出来。然而，如果售货机没有零钱找，则‘零钱找完’红灯亮，这时在投入1元硬币并按下饮料按钮后，饮料不送出来而且1元硬币也退出来；若有零钱找，则‘零钱找完’红灯灭，在送出饮料的同时退还5角硬币”。

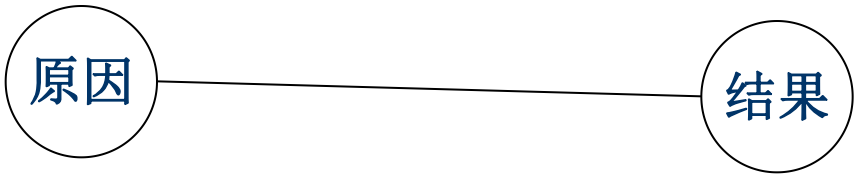
因果图法设计测试用例思想

- 首先从程序规格说明书的描述中，找出因(输入条件)和果(输出结果或者程序状态的改变)；
- 然后根据输入之间、输出之间、输入与输出之间的关系画出因果图；
- 然后通过因果图转换为判定表，最后为判定表中的每一列设计一个测试用例。

因果图法的定义：

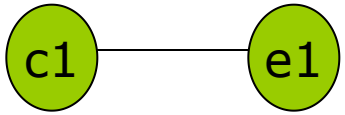
- 是一种利用图解法分析输入的各种组合情况，从而设计测试用例的方法，它适合于检查程序输入条件的各种组合情况。

因果图中出现的基本符号

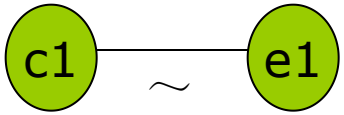


- 通常在因果图中用Ci表示原因，用Ei表示结果，各结点表示状态，可取值“0”或“1”。“0”表示某状态不出现，“1”表示某状态出现。

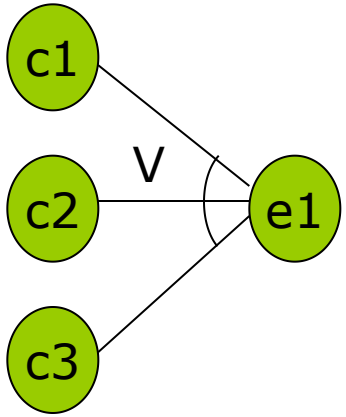
2.3.2主要的原因与结果之间的关系



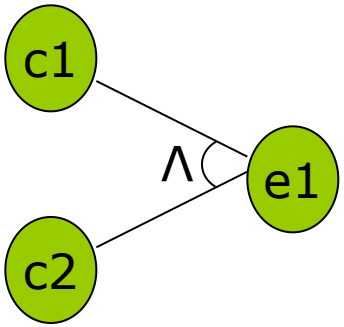
（a）恒等



（b）非

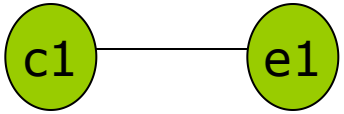


（c）或



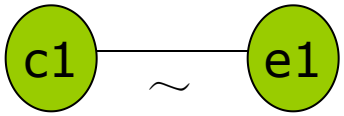
（d）与

- 恒等：若c1是1，则e1也为1，否则e1为0；



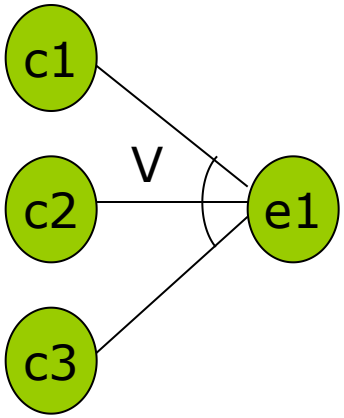
表示当原因C1为真时则得出E1。比如，如果得分大于等于60为真，则是否及格为真。

- 非：若c1是1，则e1为0，否则e1为1；



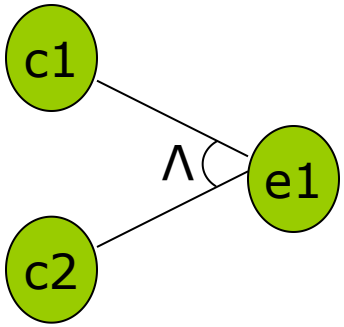
逻辑非的关系，表示当原因C1为假时则得出e1。
比如，如果得分大于等于60为假，则是否及格为假。

- 或：若c1或c2或c3是1，则e1是1，否则e1为0，“或”可有任意个输入；



逻辑或的关系，表示当原因C1、C2、C3其一为真时则得出E1。比如，招聘单位招聘条件为：至少精通DB2、ORACLE、SQLSERVER一种数据库。

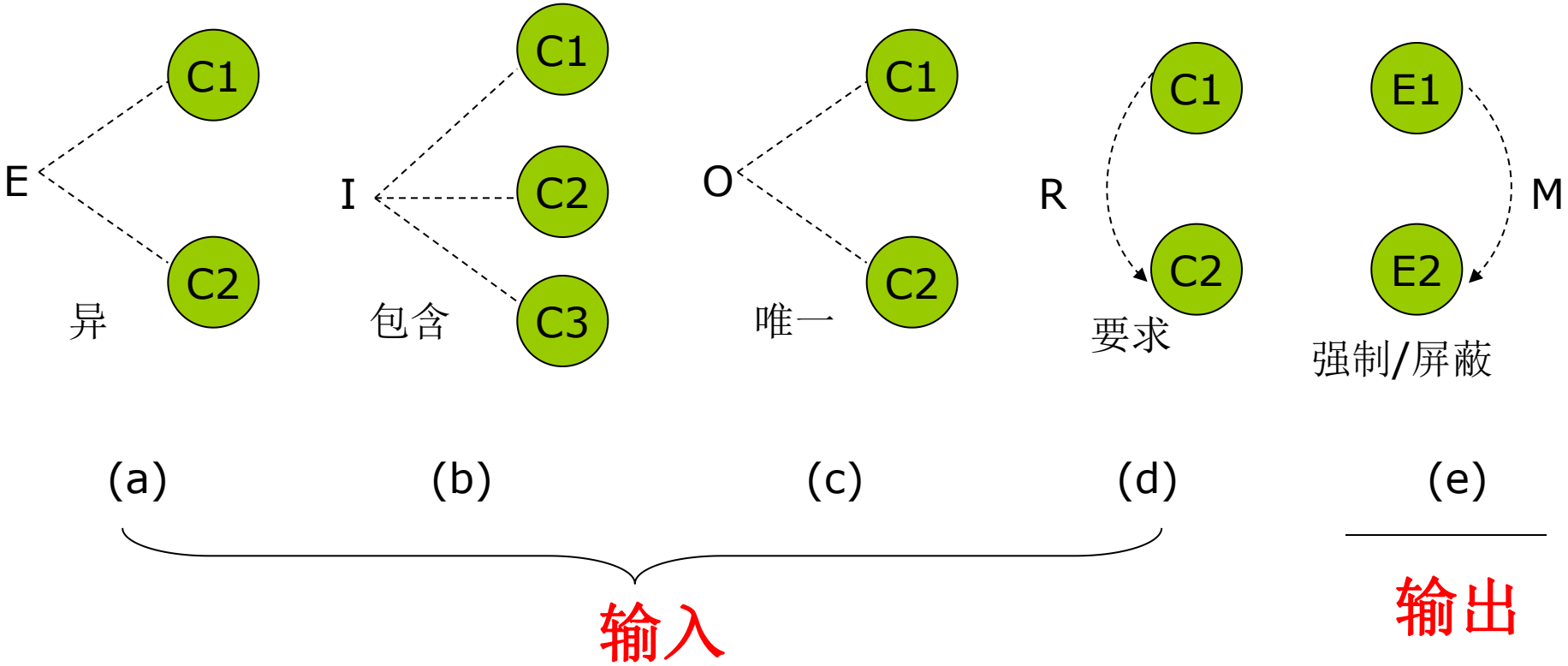
- 与：若c1和c2都是1，则e1为1，否则e1为0，“与”也可有任意个输入。



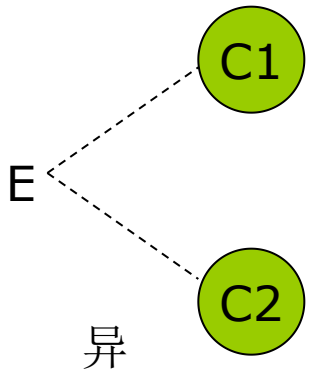
逻辑与的关系，表示当原因C1、C2全部为真时则得出E1。比如，招聘单位招聘条件为：C1大本以上学历，C2精通JAVA。

- 恒等：若c1是1，则e1也为1，否则e1为0；
- 非：若c1是1，则e1为0，否则e1为1；
- 或：若c1或c2或c3是1，则e1是1，否则e1为0，“或”可有任意个输入；
- 与：若c1和c2都是1，则e1为1，否则e1为0，“与”也可有任意个输入。

- 在实际问题当中输入/输出状态相互之间还可能存在某些依赖关系，称为“约束”

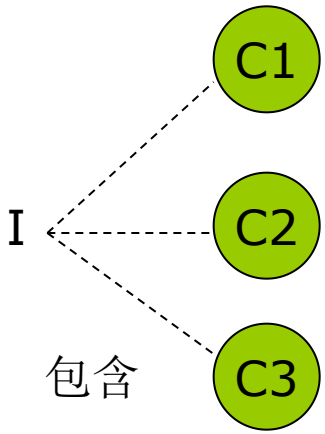


- E约束（异）：C1和C2中最多有一个可能为1，即a和b不能同时为1；



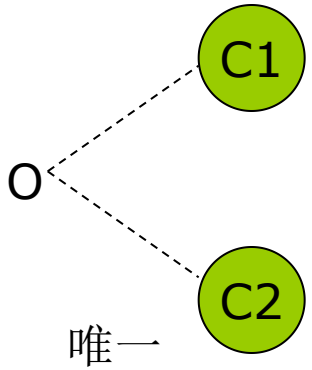
逻辑异关系，表示C1与C2最多只能有一个为真，但均可为假。想吃C1鱼还是想吃C2熊掌？鱼与熊掌不可兼得，却可都不得。

- I约束（包含）：C1、C2、C3中至少有一个必须是1，即C1、C2、C3不能同时为0；



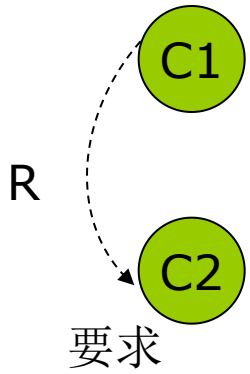
逻辑或关系，表示C1,C2与C3最多只能有一个为假，但均可为真。比如，商店收款时，可以C1支持刷银联卡支付方式？可以C2支持现金支付方式？可以C3购物卡支付？显然，可以都支持，却不可以都不支持。

- O约束（唯一）：C1和C2必须有一个且仅有一个为1；



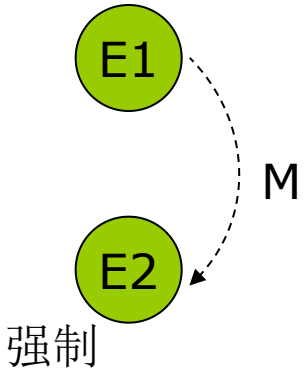
表示唯一关系，C1与C2只能有一个为真。比如单位领导找你谈话，你是想出国锻炼发展呢，还是想留在国内升职？

- R约束（要求）：C1是1时，C2必须是1；不可能C1出现，C2不出现。



表示要求关系，当C1为真时，则C2必须为真，比如windows操作系统的TCP/IP属性配置，当手工指定IP地址为真时，则手工指定DNS地址必然为真。

- M约束（强制/屏蔽）：若结果E1是1，则结果E2强制为0，当E1为0时，E2的值不定。

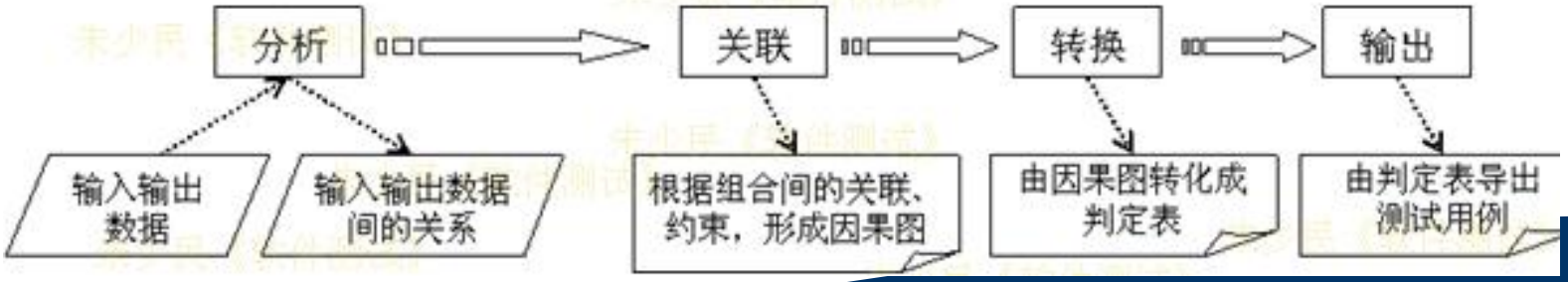


表示强制关系，当E1为真时，则E2必须为假，TCP/IP属性配置比如windows操作系统的TCP/IP属性配置，当手工指定IP地址为真时，则自动获得DNS地址必然为假。

- **对于输入条件的约束有4种：**
  - E约束（异）：C1和C2中最多有一个可能为1，即C1和C2不能同时为1；
  - I约束（包含）：C1、C2、C3中至少有一个必须是1，即C1、C2、C3不能同时为0；
  - O约束（唯一）：C1和C2必须有一个且仅有一个为1；
  - R约束（要求）：C1是1时，C2必须是1；
- **对于输出条件的约束只有M约束**
  - M约束（强制/屏蔽）：若结果E1是1，则结果E2强制为0。

2.3.3设计步骤

- 分析软件规格说明书中的输入输出条件，将每个输入输出赋予一个标志符
- 分析规格说明中的语义，通过这些语义来找出多个输入因素之间的关系。
- 找出输入因素与输出结果之间的关系，将对应的输入与输出之间的关系关联起来，并将其中不可能的组合情况标注成约束或者限制条件，形成因果图。
- 由因果图转化成判定表，任何由输入与输出之间关系构成的路径，形成判定表的一列，即来判定表的一条规则。
- 将判定表的每一列拿来作为测试用例的设计依据，一列对应一个测试用例。



(1)分析：

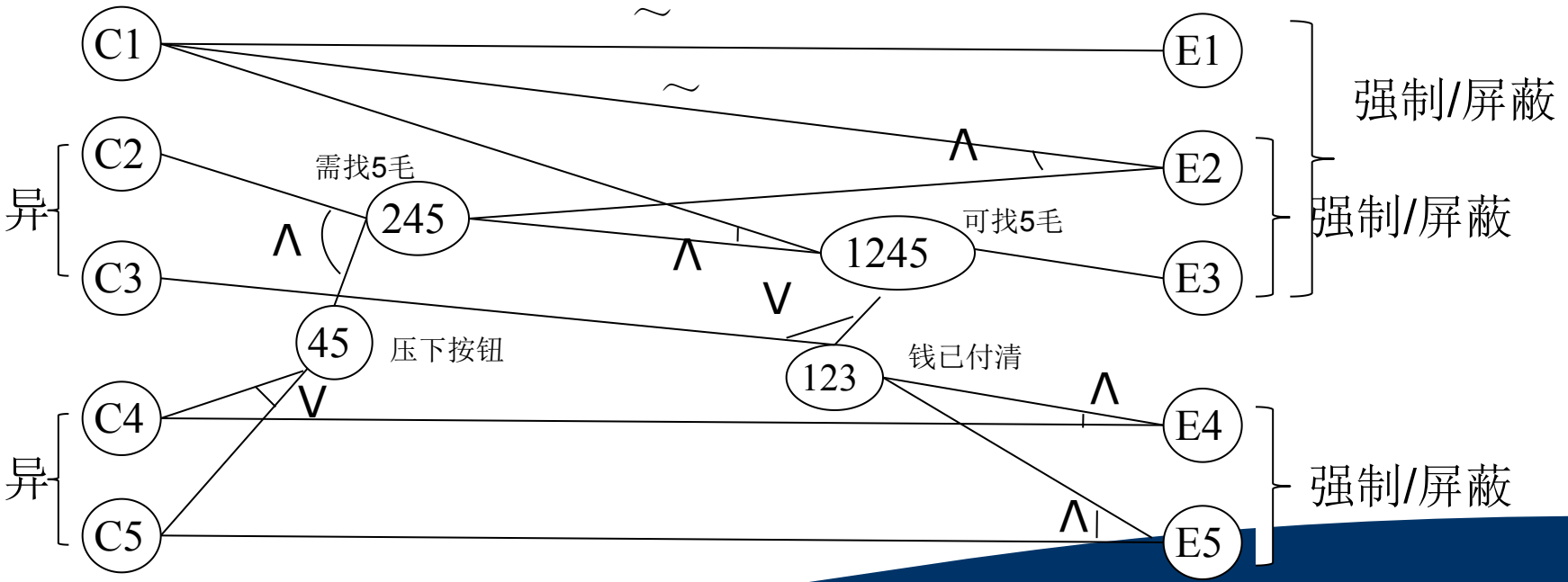
C1：售货机有零钱
C2：投入1元硬币
C3：投入5角硬币
C4：压下橙汁按钮
C5：压下啤酒按钮

E1：售货机“零钱找完”红灯亮
E2：退还1元硬币
E3：找回5角硬币
E4：送出橙汁饮料
E5：送出啤酒饮料

中间节点：

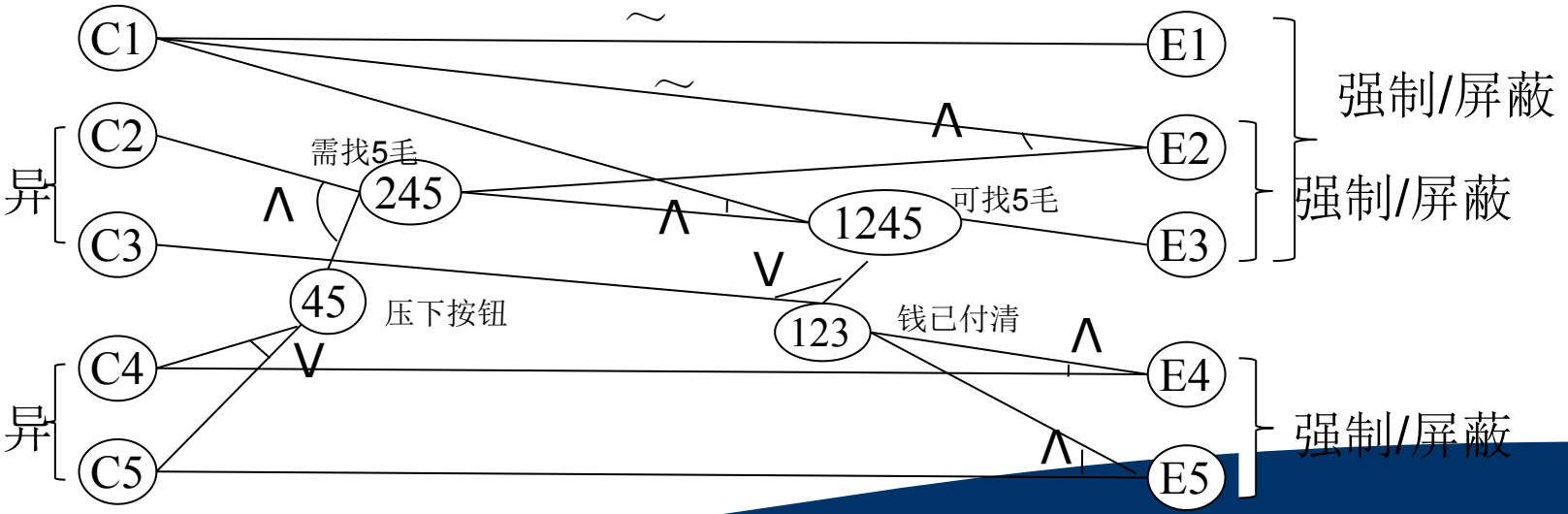
245：投入 1 元硬币且按下饮料按钮 → 需找5毛
45：按下“橙汁”或“啤酒”按钮 → 压下按钮
1245：应当找 5 角零钱并且售货机有零钱找 → 可找5毛
123：钱已付清 → 投入5毛或投入1元找回5毛

(2)画因果图：



（3）根据因果图，就可以转化为判定表。这里根据条C2 与C3、C4与C5的E约束（互斥），可以减少组合

输入	C1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	C2	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1
	C3	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0
	C4	0	1	0	0	1	0	0	1	0	1	0	0	1	0	0	1
	C5	1	0	0	1	0	0	1	0	1	0	0	1	0	0	1	0
中间结果	245	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
	45	1	1	0	1	1	0	1	1	1	1	0	1	1	0	1	1
	1245	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
	123	0	0	1	1	1	0	0	0	0	0	1	1	1	0	1	1
结果	E1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
	E2	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
	E3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
	E4	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1
	E5	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0



（4）得到测试用例：

从右到左列出测试用例

输入	C1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	C2	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1
	C3	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0
	C4	0	1	0	0	1	0	0	1	0	1	0	0	1	0	0	1
	C5	1	0	0	1	0	0	1	0	1	0	0	1	0	0	1	0
中间结果	245	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
	45	1	1	0	1	1	0	1	1	1	1	0	1	1	0	1	1
	1245	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
	123	0	0	1	1	1	0	0	0	0	0	1	1	1	0	1	1
结果	E1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
	E2	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
	E3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
	E4	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1
	E5	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0

表 17-6　自动饮料售货机软件的测试用例

序　号	输　入	输　出
1	有零钱找，投入 1 元，按下“橙汁”按钮	灯不亮，找回 5 角，出橙汁饮料
2	有零钱找，投入 1 元，按下“啤酒”按钮	灯不亮，找回 5 角，出啤酒饮料

（4）得到测试用例(续):

输入	C1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	C2	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1
	C3	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0
	C4	0	1	0	0	1	0	0	1	0	1	0	0	1	0	0	1
	C5	1	0	0	1	0	0	1	0	1	0	0	1	0	0	1	0
中间结果	245	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
	45	1	1	0	1	1	0	1	1	1	1	0	1	1	0	1	1
	1245	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
	123	0	0	1	1	1	0	0	0	0	0	1	1	1	0	1	1
结果	E1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
	E2	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
	E3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
	E4	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1
	E5	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0

续表

序　号	输　入	输　出
3	有零钱找，投入1元，未按按钮	灯不亮，不找钱，不出饮料
4	有零钱找，投入5角，按下“橙汁”按钮	灯不亮，不找钱，出橙汁饮料
5	有零钱找，投入5角，按下“啤酒”按钮	灯不亮，不找钱，出啤酒饮料
6	有零钱找，投入5角，未按按钮	灯不亮，不找钱，不出饮料
7	有零钱找，未投币，按下“橙汁”按钮	灯不亮，不找钱，不出饮料
8	有零钱找，未投币，按下“啤酒”按钮	灯不亮，不找钱，不出饮料
9	无零钱找，投入1元，按下“橙汁”按钮	灯亮，退回1元，不出饮料
10	无零钱找，投入1元，按下“啤酒”按钮	灯亮，退回1元，不出饮料
11	无零钱找，投入1元，未按按钮	灯亮，不退钱，不出饮料
12	无零钱找，投入5角，按下“橙汁”按钮	灯亮，不退钱，出橙汁饮料
13	无零钱找，投入5角，按下“啤酒”按钮	灯亮，不退钱，出啤酒饮料
14	无零钱找，投入5角，未按按钮	灯亮，不退钱，不出饮料
15	无零钱找，未投币，按下“橙汁”按钮	灯亮，不退钱，不出饮料
16	无零钱找，未投币，按下“啤酒”按钮	灯亮，不退钱，不出饮料

2.3.4 因果图法测试练习

08实训练习第1题

- 实例 用因果图法测试以下程序。

  程序的规格说明要求：输入的第一个字符必须是#或*，第二个字符必须是一个数字，此情况下进行文件的修改；如果第一个字符不是#或*，则给出信息N，如果第二个字符不是数字，则给出信息M。

- 解题步骤：

（1）分析程序的规格说明，列出原因和结果。

（2）找出原因与结果之间的因果关系、原因与原因之间的约束关系，画出因果图。

（3）将因果图转换成判定表。

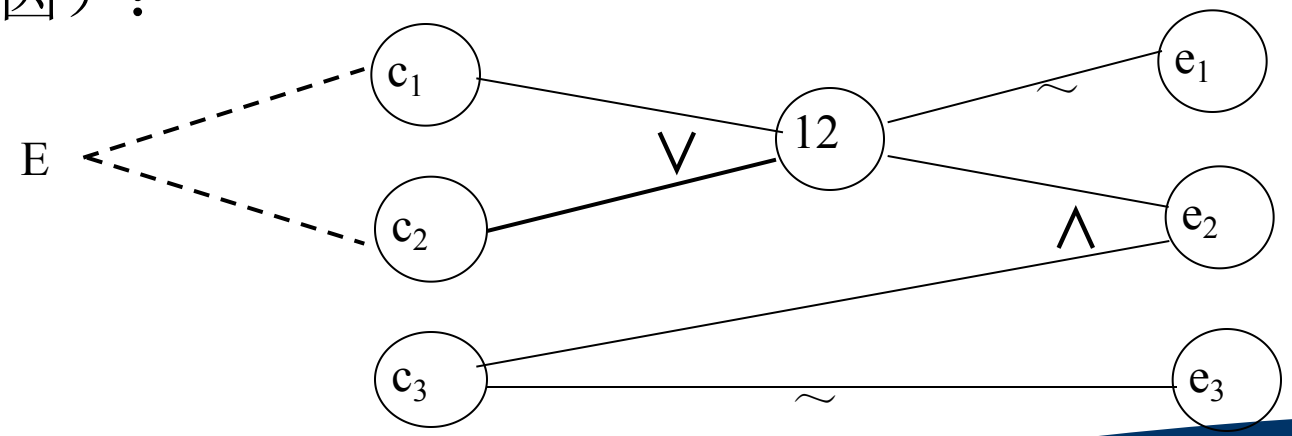
（4）根据（3）中的判定表，设计测试用例的输入数据和预期输出。

因果图法测试举例（续）

（1）分析程序规格说明中的原因和结果：

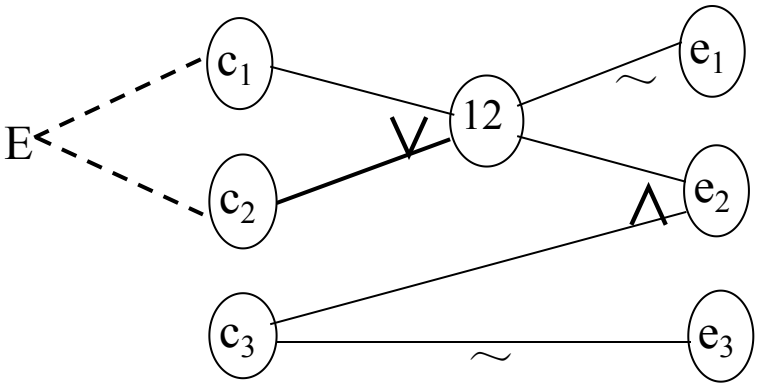
原因	结果
c1：第一个字符是#	e1：给出信息N
c2：第一个字符是*	e2：修改文件
c3：第二个字符是一个数字	e3：给出信息M

（2）画出因果图（编号为12的中间结点是导出结果的进一步原因）：



因果图法测试举例（续）

（3）将因果图转换成如下所示的判定表：



选项 \ 规则	7	6	5	4	3	2	1	0
C1第一个字符是#	1	1	1	1	0	0	0	0
C2第一个字符是*	1	1	0	0	1	1	0	0
C3第二个字符是一个数字	1	0	1	0	1	0	1	0
e1给出信息N							√	√
e2修改文件			√		√			
e3给出信息M				√		√		√
不可能	√	√						
测试用例			#3	#A	*6	*B	A1	GT

因果图法测试举例（续）

（4）根据判定表中的每一列设计测试用例：

测试用例编号	输入数据	预期输出
1	#3	修改文件
2	#A	给出信息M
3	*6	修改文件
4	*B	给出信息M
5	A1	给出信息N
6	GT	给出信息N和信息M