

对于 t_1 , 谓词 p_c 和 p_i 的真值均为 false, 因此 t_1 无法发现 p_i 中的故障; 对于 t_2 , 谓词 p_c 真值为 true, 而 p_i 真值为 false, 因此 t_2 将发现 p_i 中的故障。

缺失/冗余布尔变量故障

在前面讨论的故障模型中, 没有考虑另外两类故障, 即缺失布尔变量故障和冗余布尔变量故障。

举个例子, 考虑某过程控制系统, 系统监测液体容器的压力 P 和温度 T , 并将结果传送给控制计算机。控制计算机中的紧急情况检测系统在条件 $T > T_{\max}$, $P > P_{\max}$ 任一为真时发出告警。将该告警需求规范转换为谓词 $p_r: T > T_{\max} \vee P > P_{\max}$ 。当 p_r 取值为真时, 计算机发出告警, 否则不告警。

将 p_r 写成布尔表达式形式 $a + b$, 其中 $a = T > T_{\max}$, $b = P > P_{\max}$ 。

现在假设, 编写控制软件时错误地将 p_r 编码为 a , 而不是 $a + b$ 。显然 p_r 编码中存在故障, 称该故障为缺失布尔变量故障。

同样假设, 编写控制软件时错误地将 p_r 编码为 $a + b + c$, 其中 c 是一个布尔变量, 它表示某个条件。显然 p_r 编码中存在另一类故障, 称该故障为冗余布尔变量故障。

本章介绍的各种测试用例生成方法, 都无法确保能够有效发现缺失/冗余布尔变量故障。第7章将介绍基于程序变异的测试用例生成方法, 在某些条件下能够确保发现这些故障。

2.7.3 谓词约束

用 BR 表示符号集合 $\{t, f, <, =, >, +\varepsilon, -\varepsilon\}$ 。“BR”是布尔和关系 (Boolean and Relational) 的缩写。集合 BR 中的元素称为 BR 符号。

一个 BR 符号定义了针对某个布尔变量或关系表达式的约束。例如, 用符号 “ $+\varepsilon$ ” 约束表达式 $E': e_1 < e_2$ 时, 要满足该约束就要求 E' 的某个测试用例确保 $0 < e_1 - e_2 \leq \varepsilon$ 。同样, 符号 “ $-\varepsilon$ ” 是对 E' 的另一个约束, 要满足该约束就要求 E' 的某个测试用例确保 $-\varepsilon \leq e_1 - e_2 < 0$ 。

对于 p_r 中的变量, 如果不存在满足约束 C 的输入取值, 则称约束 C 对谓词 p_r 是无效的 (infeasible)。例如, 对谓词 $a > b \wedge b > d$ 的约束 $(>, >)$, 要求简单谓词 $a > b$ 和 $b > d$ 都为真。但是, 如果 $d > a$ 为真的话, 则该约束无效。

例 2.22 这里举一个简单的约束例子, 考虑关系表达式 $E: a < c + d$ 以及 E 上的约束 $C: (=)$ 。当验证 E 的正确性时, 约束 C 要求测试集至少包含一个测试用例 $a = c + d$ 。这样, 测试用例 $\langle a=1, c=0, d=1 \rangle$ 满足 E 上的约束 C 。

另一个例子, 考虑 E 上的约束 $C: (+\varepsilon)$, 令 $\varepsilon=1$ 。满足约束 C 的测试用例要求 $0 < a - (c + d) \leq 1$ 。因此, 测试用例 $\langle a=4, c=2, d=1 \rangle$ 满足 E 上的约束 $(+\varepsilon)$ 。

同样, 对于布尔表达式 $E: b$, 约束 “t” 要求测试用例将变量 b 取值为 true。

BR 符号 t 和 f 用于定义布尔变量和布尔表达式的约束; 关系表达式上的约束可用 $<, =, >$ 三个符号来定义。当关系表达式被当作简单布尔变量时, 符号 t 和 f 也可用于定义关系表达式上的约束。例如, 可将表达式 $p_r: a < b$ 视作布尔变量 z , 此时, 可以使用 t 和 f 约束 p_r 。

现在, 定义对整个谓词的约束, 该谓词由布尔变量、关系表达式通过布尔运算符连接而成。

设 p_r 为包含 n ($n > 0$) 个 \wedge 和 \vee 运算符的谓词, p_r 的谓词约束 C 是由 $n+1$ 个 BR 符号组成的序列, 其中每个 BR 符号分别对应于 p_r 中的某个布尔变量或关系表达式。为方便起见, 将谓词约束简称为约束。

对于测试用例 t , 如果 p_r 的各组件都满足 C 中相应的约束, 称测试用例 t 满足 p_r 上的约束