**Lab6 Assignment 2: Design Tests**

1. **Input space partition**

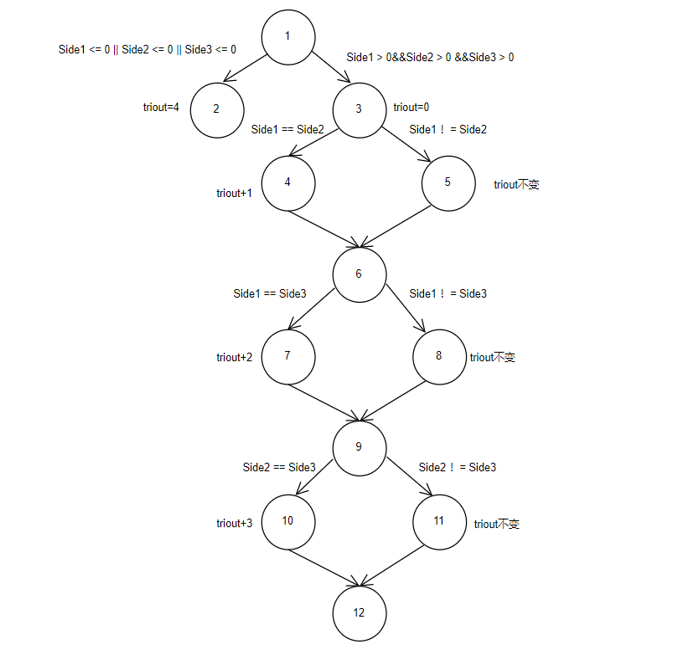
**(1) Each choice coverage**

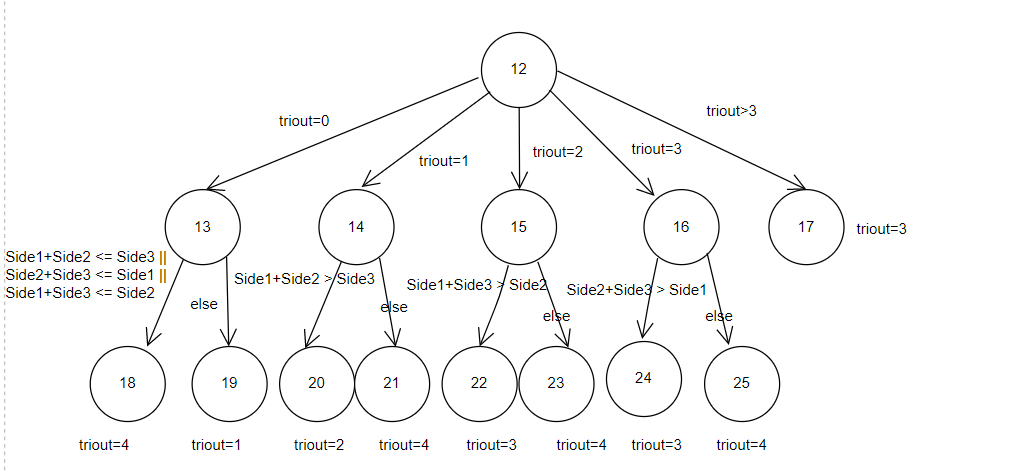
|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | EO |
| 1 | 1 | 1 | 3 |
| 2 | 3 | 4 | 1 |
| -1 | -1 | -1 | 4 |
| 1 | 2 | 2 | 2 |
| 0 | 0 | 0 | 4 |

**(2) Pair-wise coverage**

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | EO |
| -1 | -1 | 0 | 4 |
| -1 | 0 | 1 | 4 |
| -1 | 1 | 1 | 4 |
| -1 | 1 | -1 | 4 |
| 0 | -1 | 1 | 4 |
| 0 | 0 | 1 | 4 |
| 0 | 1 | -1 | 4 |
| 0 | 1 | 0 | 4 |
| 1 | 2 | 1 | 4 |
| 1 | 1 | 1 | 3 |
| 1 | 1 | -1 | 4 |
| 1 | 0 | 1 | 4 |
| 2 | 0 | 0 | 4 |
| 2 | 3 | 4 | 1 |
| 2 | 2 | 2 | 3 |
| 2 | 3 | 3 | 2 |

**2.Graph coverage**

首先画控制流图(图片太大分成了两个)



由于这次控制流图是自上往下没有出现循环或等等情况，所以edge coverage，node coverage，prime path coverage的测试用例可以通用。

1. **Node coverage**

节点覆盖:[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,20,21,22,23,24,25]

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | EO |
| 0 | 1 | 1 | 4 |
| 1 | 1 | 1 | 3 |
| 2 | 3 | 2 | 2 |
| 2 | 2 | 3 | 2 |
| 2 | 4 | 2 | 4 |
| 3 | 2 | 2 | 2 |
| 4 | 2 | 2 | 4 |
| 1 | 2 | 3 | 4 |
| 2 | 3 | 4 | 1 |
| 2 | 3 | 3 | 2 |

1. **edge coverage**

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | EO |
| 0 | 1 | 1 | 4 |
| 1 | 1 | 1 | 3 |
| 2 | 3 | 2 | 2 |
| 2 | 2 | 3 | 2 |
| 2 | 4 | 2 | 4 |
| 3 | 2 | 2 | 2 |
| 4 | 2 | 2 | 4 |
| 1 | 2 | 3 | 4 |
| 2 | 3 | 4 | 1 |
| 2 | 3 | 3 | 2 |

1. **prime path coverage**

主路径有：

[1,2][1,3,4,6,7,9,10,12,13,18][1,3,4,6,7,9,10,12,13,19] [1,3,4,6,7,9,10,12,14,20][1,3,4,6,7,9,10,12,14,21][1,3,4,6,7,9,10,12,15,22] [1,3,4,6,7,9,10,12,15,23]等等共81条主路径

经过排查有10条可以设计测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | EO |
| 0 | 1 | 1 | 4 |
| 1 | 1 | 1 | 3 |
| 2 | 3 | 2 | 2 |
| 2 | 2 | 3 | 2 |
| 2 | 4 | 2 | 4 |
| 3 | 2 | 2 | 2 |
| 4 | 2 | 2 | 4 |
| 1 | 2 | 3 | 4 |
| 2 | 3 | 4 | 1 |
| 2 | 3 | 3 | 2 |

1. **all-use coverage**

对于triout=4所有变量都可以达到使用

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | EO |
| -1 | 1 | 1 | 4 |
| 2 | 2 | 4 | 4 |
| 2 | 4 | 2 | 4 |
| 4 | 2 | 2 | 4 |
| 2 | 3 | 5 | 4 |

**3.Logic coverage**

**(1) Predicate coverage**

**T F**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | EO |  | A | B | C |  |
| 0 | 0 | 0 | 4 | 1 | 1 | 1 | 3 |
| 1 | 1 | 1 | 3 | 1 | 2 | 2 | 2 |
| 1 | 1 | 1 | 3 | 1 | 2 | 2 | 2 |
| 1 | 1 | 1 | 3 | 2 | 1 | 2 | 2 |
| 1 | 2 | 3 | 4 | 1 | 1 | 1 | 3 |
| 1 | 2 | 3 | 4 | 2 | 3 | 4 | 1 |
| 1 | 1 | 1 | 3 | 2 | 2 | 3 | 2 |
| 2 | 2 | 3 | 2 | 2 | 2 | 4 | 4 |
| 2 | 3 | 2 | 2 | 3 | 4 | 2 | 4 |
| 3 | 2 | 2 | 2 | 4 | 2 | 2 | 4 |

1. **clause coverage**

**T F**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | EO |  | A | B | C | EO |
| 0 | 1 | 1 | 4 | 1 | 1 | 1 | 3 |
| 1 | 0 | 1 | 4 | 1 | 1 | 1 | 3 |
| 1 | 1 | 0 | 4 | 1 | 1 | 1 | 3 |
| 2 | 3 | 6 | 4 | 2 | 3 | 4 | 1 |
| 6 | 2 | 3 | 4 | 2 | 3 | 4 | 1 |
| 2 | 6 | 3 | 4 | 2 | 3 | 4 | 1 |
| 2 | 2 | 3 | 2 | 2 | 3 | 2 | 2 |
| 2 | 2 | 3 | 2 | 2 | 2 | 5 | 4 |
| 2 | 3 | 2 | 2 | 3 | 2 | 2 | 2 |
| 2 | 3 | 2 | 2 | 2 | 5 | 2 | 4 |
| 1 | 2 | 2 | 2 | 1 | 2 | 1 | 4 |
| 3 | 2 | 2 | 2 | 5 | 2 | 2 | 4 |

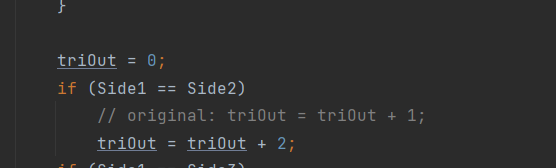
**(3) Correlated Active Clause Coverage**

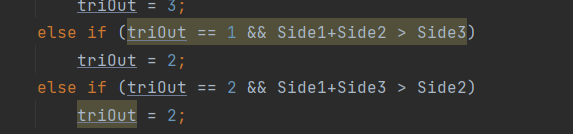
|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | EO |
| 0 | 1 | 1 | 4 |
| 1 | 1 | 1 | 3 |
| 1 | 0 | 1 | 4 |
| 1 | 1 | 0 | 4 |
| 2 | 3 | 6 | 4 |
| 2 | 3 | 4 | 1 |
| 6 | 2 | 3 | 4 |
| 2 | 6 | 3 | 4 |
| 2 | 2 | 3 | 2 |
| 2 | 3 | 3 | 2 |
| 2 | 2 | 5 | 4 |
| 2 | 3 | 3 | 2 |
| 2 | 3 | 3 | 2 |
| 2 | 5 | 2 | 4 |
| 3 | 2 | 2 | 2 |
| 1 | 2 | 2 | 4 |
| 5 | 2 | 2 | 4 |

**4. mutation testing**

Kill the mutant：

**(1) TritypMutantOne**

TritypMutantOne这个mutant program在这个地方更改，影响的是在这段if判断后triout的值，当side1=side2时triout=2而不=1,后续就不会出现triout == 1 && Side1+Side2 > Side3的情况，只有triout == 2 && Side1+Side3 > Side2的情况



所以根据这个设计测试用例。

原本情况：

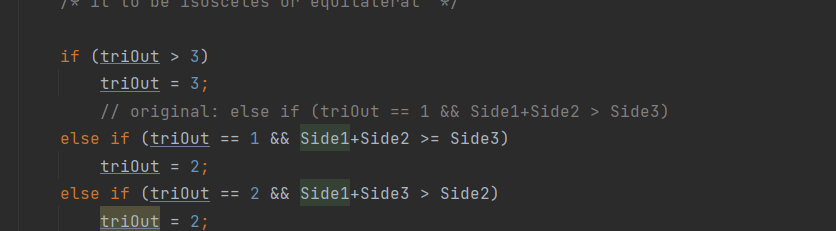
|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | EO |
| 2 | 2 | 4 | 4 |

现在情况:

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | EO |
| 2 | 2 | 4 | 2 |

**(2) TritypMutantTwo**

TritypMutantTwo这个mutant program在这个地方更改



(triout == 1 && Side1+Side2 > Side3)变成了 (triout == 1 && Side1+Side2 >= Side3)根据这个设计测试用例。

原本情况：

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | EO |
| 2 | 2 | 4 | 4 |

现在情况:

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | EO |
| 2 | 2 | 4 | 2 |

很巧的是 2 4 2这个测试用例在两个程序中都因mutant而发生结果的改变。