Input space partition

Each choice coverage

Side1	Side2	Side3
2	2	2
1	1	1
0	0	0
-1	-1	-1

Pair-wise coverage

Side1	Side2	Side3
2	2	2
1	2	1
0	2	0
-1	2	-1
1	1	2
0	1	1
-1	1	0
2	1	-1
0	0	2
-1	0	1
2	0	0
1	0	-1
-1	-1	2

2	-1	1
1	-1	0
0	-1	-1

Graph coverage

Node coverage

Side1	Side2	Side3
-1	-1	-1
2	2	2
2	1	2

Edge coverage

Side1	Side2	Side3
-1	-1	-1
2	2	3
2	2	2
2	2	5
2	5	2
1	2	2
5	2	2
1	2	5

Prime path coverage

Side1	Side2	Side3
-1	-1	-1
2	2	5
2	2	2
2	5	2
5	2	2
1	2	5
2	2	2

All-use coverage

Side1	Side2	Side3
3	2	2
2	3	2
2	2	3

Logic coverage

```
triOut == 3 && Side2+Side3 > Side1
```

Predicate coverage

	Side1	Side2	Side3
true	2	2	2
false	2	2	5

Clause coverage

Clause1:

```
triOut == 1 && Side1+Side2 > Side3
```

clause2:

```
triOut == 2 && Side1+Side3 > Side2
```

Side1	Side2	Side3
2	2	1

Clause1:true clause2:true

Side1	Side2	Side3
2	3	4

Clause1:true clause2:false

Side1	Side2	Side3
2	1	1

Clause1:false clause2:false

Side1	Side2	Side3
2	2	2

Clause1:false clause2:true

Correlated active clause coverage

Major clause: triout==1

Side1	Side2	Side3
2	2	1

Major clause: true

Minor clause: true

Predicate: true

Side1	Side2	Side3
1	2	5

Major clause: false

Minor clause: false

Predicate: false

Mutation testing

TritypMutantOne

Side1	Side2	Side3
2	2	5

Expected:4

Actually:2

TritypMutantTwo

Side1	Side2	Side3
2	2	4

Expected:4

Actually:2