

특론 Lab 2 report
21700675 정지훈
21700554 이은강

1. Base-cases in tcas.c

```
80     if (probe("X1", upward_preferred))  
  
98     if (probe("X2", upward_preferred))  
  
131     if (probe("(A&&((B&&C) || D))", (probe("A", enabled) && probe("(B&&C) || D)",  
           (probe("B&&C)", (probe("B", tcas_equipped) && probe("C", intent_not_known))) ||  
           probe("D", !tcas_equipped))))  
  
135     if (probe("Y&&Z", (probe("Y", need_upward_RA) && probe("Z", need_downward_RA))) ||  
           (probe("Y", need_upward_RA) && probe("Z", need_downward_RA)))  
  
137     else if (probe("Y'", need_upward_RA))  
  
139     else if (probe("Z'", need_downward_RA))  
  
155     if (probe("M1", argc < 13))
```

- Total 7 decisions and 11 basic conditions in tcas.c

2. Truth table for decision on line 135

	A	B	C	D	Result	Short Circuit
1	T	T	T	T	T	A,B,C
2	T	T	T	F	T	A,B,C
3	T	T	F	T	T	A,B,C,D
4	T	T	F	F	F	A,B,C,D
5	T	F	T	T	T	A,B,D
6	T	F	T	F	F	A,B,D
7	T	F	F	T	T	A,B,D
8	T	F	F	F	F	A,B,D
9	F	T	T	T	F	A
10	F	T	T	F	F	A
11	F	T	F	T	F	A
12	F	T	F	F	F	A
13	F	F	T	T	F	A
14	F	F	T	F	F	A
15	F	F	F	T	F	A
16	F	F	F	F	F	A

- Case 2, 3, 4, 6, 9 makes up a MC/DC coverage for this decision.

3. Random test generator

```

for(int j = 0; j<100000; j++){
    for(int i = 1; i<=12; i++) {
        if(i == 2 || i == 3)
            fprintf(fs, "%d ", rand() % 2);
        else if(i == 7)
            fprintf(fs, "%d ", rand() % 4);
        else if(i == 10 || i == 12)
            fprintf(fs, "%d ", rand() % 3);
        else if(i == 11)
            fprintf(fs, "%d ", rand() % 2 + 1);
        else
            fprintf(fs, "%d ", rand() % 9999 - 5000);
    }
    fprintf(fs, "\n");
}

```

This program randomly generates 100000 test inputs for tcas. It considers the possible values each field can have.

1	3175	0	0	-4363	-3282	664	2	1668	-3452	0	1	1
2	-3550	0	1	954	4891	-1517	3	-566	190	1	2	2
3	-2143	1	0	-1546	-4713	-971	2	3463	-4095	0	2	2
4	-3014	1	0	116	-2818	3623	1	-1368	-88	2	1	2
5	-281	1	1	4909	-4277	4056	2	-1419	2424	2	1	2
6	2769	1	0	-4743	168	-4977	3	2155	1450	1	2	2
7	2125	1	0	-1379	1566	1598	2	-3713	-4336	0	1	2
8	-1949	0	0	-2942	848	1751	1	201	598	0	1	2
9	-4283	0	0	3749	3423	-1640	1	549	3690	1	2	1
10	-2423	0	0	-177	-244	-4093	2	2807	-1985	2	2	0
11	-740	0	1	4858	77	-2311	1	2376	-807	1	1	1
12	74	1	0	-1235	-2347	-4497	1	1812	-2153	2	1	1
13	4639	1	0	-2345	-3315	1861	2	945	-3754	0	1	2
14	-1977	1	1	2216	-2109	-3589	2	2965	4668	0	1	1
15	-1748	0	1	1100	-2348	518	0	-2708	4694	0	2	0
16	4142	0	0	-4611	-1870	1294	3	2735	2472	0	2	2
17	-10	1	1	-341	-4298	-1774	1	-1046	3727	0	2	0
18	2965	0	0	4242	2623	-1633	3	3349	1164	0	1	0
19	2409	1	0	1464	-2804	149	1	3768	-1898	2	2	0
20	-285	1	1	25	1021	-4021	1	569	1321	0	1	1
21	3192	1	1	-643	3133	-801	3	2125	-2184	0	1	1
22	-4419	0	1	-1317	72	-4624	1	-3630	533	1	1	2
23	-4028	1	1	2293	4210	103	0	2404	-1295	1	2	0
24	-2519	0	1	1880	593	-862	0	1174	4304	0	2	1
25	-4201	0	0	2914	-4496	-1273	2	-3524	4694	2	2	2
26	3028	1	0	-3266	-1961	-3762	2	2103	-3599	2	2	2

This is part of the test inputs the program generated.

4. Test shell script

```
1 #!/bin/bash
2 gcc -o tcas tcas.c
3
4 while read p
5 do
6     ./tcas $p
7 done < rand.txt > log.txt
```

This shell script builds the tcas program and for each line in the test input, it writes the output to a log file.

```
1 8175 0 0 -4363 -3282 664 2 1668 -3452 0 1 1
2 M1:0 A:0 (A&&((B&&C) || D)):0 0
3 -
4 -3550 0 1 954 4891 -1517 3 -566 190 1 2 2
5 M1:0 A:0 (A&&((B&&C) || D)):0 0
6 -
7 -2143 1 0 -1546 -4713 -971 2 3463 -4095 0 2 2
8 M1:0 A:0 (A&&((B&&C) || D)):0 0
9 -
10 -3014 1 0 116 -2818 3623 1 -1368 -88 2 1 2
11 M1:0 A:0 (A&&((B&&C) || D)):0 0
12 -
13 -281 1 1 4909 -4277 4056 2 -1419 2424 2 1 2
14 M1:0 A:0 (A&&((B&&C) || D)):0 0
15 -
16 2769 1 0 -4743 168 -4977 3 2155 1450 1 2 2
17 M1:0 A:1 B:0 (B&&C):0 D:1 ((B&&C) || D):1 (A&&((B&&C) || D)):1 X1:1 X2:1 Y:0 Y&&Z:0 Y':0 Z':0 0
18 -
19 2125 1 0 -1379 1566 1598 2 -3713 -4336 0 1 2
20 M1:0 A:0 (A&&((B&&C) || D)):0 0
21 -
22 -1949 0 0 -2942 848 1751 1 201 598 0 1 2
23 M1:0 A:0 (A&&((B&&C) || D)):0 0
24 -
25 -4283 0 0 3749 3423 -1640 1 549 3690 1 2 1
26 M1:0 A:0 (A&&((B&&C) || D)):0 0
27 -
28 -2423 0 0 -177 -244 -4093 2 2807 -1985 2 2 0
29 M1:0 A:0 (A&&((B&&C) || D)):0 0
30 -
31 -740 0 1 4858 77 -2311 1 2376 -807 1 1 1
32 M1:0 A:0 (A&&((B&&C) || D)):0 0
33 -
34 74 1 0 -1235 -2347 -4497 1 1812 -2153 2 1 1
35 M1:0 A:0 (A&&((B&&C) || D)):0 0
36 -
37 4639 1 0 -2345 -3315 1861 2 945 -3754 0 1 2
38 M1:0 A:1 B:1 C:0 (B&&C):0 D:0 ((B&&C) || D):0 (A&&((B&&C) || D)):0 0
39 -
```

This is an extract of the generated log file.

5. Test Case Result

- Example sets of the maximum MC/DC coverage

```
0
3175 0 0 -4363 -3282 664 2 1668 -3452 0 1 1
4639 1 0 -2345 -3315 1861 2 945 -3754 0 1 2
1149 1 1 1395 573 -3458 3 4673 1811 0 1 0
1686 1 1 4245 -3997 -2388 1 678 4855 0 2 0
1226 1 0 2755 -3000 4315 3 -964 -1114 1 2 0
```

```
0
M1:1
-
3175 0 0 -4363 -3282 664 2 1668 -3452 0 1 1
M1:0 A:0 (A&&((B&&C) || D)):0 0
-
4639 1 0 -2345 -3315 1861 2 945 -3754 0 1 2
M1:0 A:1 B:1 C:0 (B&&C):0 D:0 ((B&&C) || D):0 (A&&((B&&C) || D)):0 0
-
1149 1 1 1395 573 -3458 3 4673 1811 0 1 0
M1:0 A:1 B:1 C:1 (B&&C):1 ((B&&C) || D):1 (A&&((B&&C) || D)):1 X1:1 X2:1 Y:0 Y&&Z:0 Y':0 Z':0 0
-
1686 1 1 4245 -3997 -2388 1 678 4855 0 2 0
M1:0 A:1 B:0 (B&&C):0 D:1 ((B&&C) || D):1 (A&&((B&&C) || D)):1 X1:0 X2:0 Y:0 Y&&Z:0 Y':0 Z':1 2
-
1226 1 0 2755 -3000 4315 3 -964 -1114 1 2 0
M1:0 A:1 B:0 (B&&C):0 D:1 ((B&&C) || D):1 (A&&((B&&C) || D)):1 X1:1 X2:1 Y:1 Z:0 Y&&Z:0 Y':1 1
-
```

M1 : 1 && 2

(MC/DC Coverage : 100%)

A : 2 && 3

C : 3 && 4

(MC/DC Coverage : 50%)

X1 & X2 : 4 && 5

(MC/DC Coverage : 100%)

Y' : 5 && 6

Z' : 4 && 5

(MC/DC Coverage : 100%)

Minimum test case : 6 Cases