Lab1: Glass-box (2023)

Mimmi Lindgren, Katrina Liang

January 2023

1 Control Flow Coverage

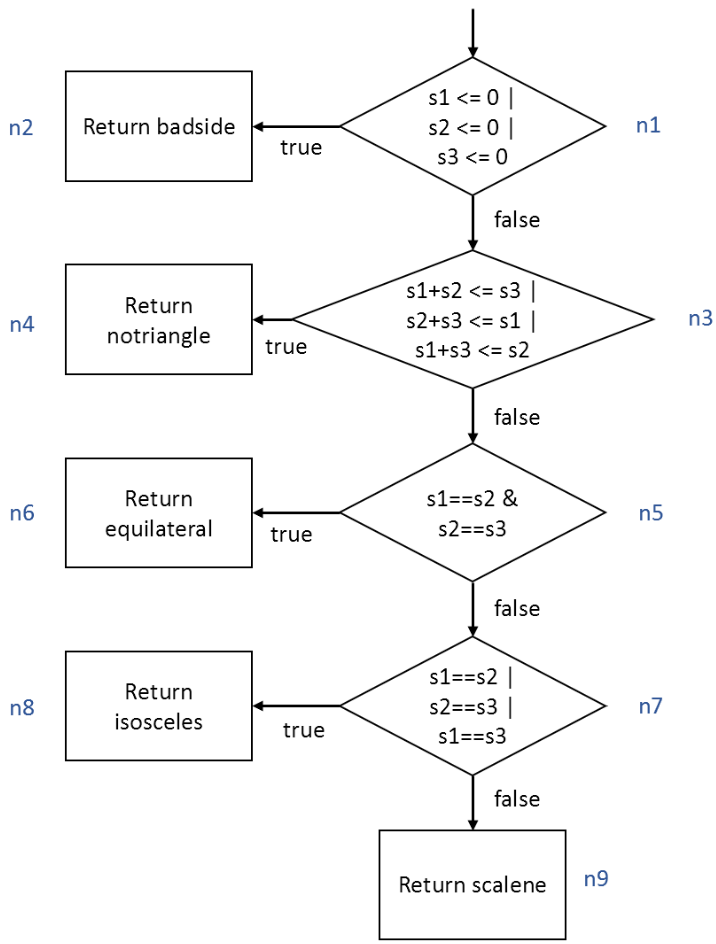

Figure 1: A condensation graph for the Triangle Test algorithm.

1.1

(a) $p_1 = \{n_1, n_2\}$
$p_2 = \{n_1, n_3, n_4\}$
$p_3 = \{n_1, n_3, n_5, n_6\}$
$p_4 = \{n_1, n_3, n_5, n_7, n_8\}$
$p_5 = \{n_1, n_3, n_5, n_7, n_9\}$
NC $TR = \{p_1, p_2, p_3, p_4, p_5\}$

Alternativ:
$p_1 = n_2$
$p_2 = n_4$
$p_3 = n_6$
$p_4 = n_8$
$p_5 = n_9$

(b) NC $TC_1 : s_1 = -1, s_2 = -1, s_3 = -1$
NC $TC_2 : s_1 = 1, s_2 = 1, s_3 = 2$
NC $TC_3 : s_1 = 1, s_2 = 1, s_3 = 1$
NC $TC_4 : s_1 = 34, s_2 = 34, s_3 = 30$
NC $TC_5 : s_1 = 3, s_2 = 4, s_3 = 5$

1.2

(a) $p_1 = \{n_1, n_2\}$
$p_2 = \{n_1, n_3, n_4\}$
$p_3 = \{n_1, n_3, n_5, n_6\}$
$p_4 = \{n_1, n_3, n_5, n_7, n_8\}$
$p_5 = \{n_1, n_3, n_5, n_7, n_9\}$
EC $TR = \{p_1, p_2, p_3, p_4, p_5\}$

Minimum redundant TR:
$p_1 = (n_1, n_2)$ $p_4 = (n_7, n_8)$
$p_2 = (n_3, n_4)$ $p_5 = (n_7, n_9)$
$p_3 = (n_5, n_6)$

(b) EC $TC_1 : s_1 = -1, s_2 = -1, s_3 = -1$
EC $TC_2 : s_1 = 1, s_2 = 1, s_3 = 2$
EC $TC_3 : s_1 = 1, s_2 = 1, s_3 = 1$
EC $TC_4 : s_1 = 34, s_2 = 34, s_3 = 30$
EC $TC_5 : s_1 = 3, s_2 = 4, s_3 = 5$

(c) Every node have only one edge going to it. So we are going over all the paths when we reach NC.

The graph is a tree

2 Logic Coverage

2.1

(a) $p_1 : s_1 <= 0 \mid s_2 <= 0 \mid s_3 <= 0$
$p_2 : s_1 + s_2 <= s_3 \mid s_2 + s_3 <= s_1 \mid s_1 + s_3 <= s_2$
$p_3 : s_1 == s_2 \; \& \; s_2 == s_3$
$p_4 : s_1 == s_2 \mid s_2 == s_3 \mid s_1 == s_3$

PC $TR_1 : p_1 = true$, $\quad TC_1 = (s_1 = -1, s_2 = -1, s_3 = -1)$
PC $TR_2 : p_1 = false$, $\quad TC_2 = (s_1 = 1, s_2 = 1, s_3 = 1)$

PC $TR_3 : p_2 = true,$ $TC_3 = (s_1 = 1, s_2 = 1, s_3 = 2)$
PC $TR_4 : p_2 = false,$ $TC_4 = (s_1 = 1, s_2 = 1, s_3 = 1)$

PC $TR_5 : p_3 = true,$ $TC_5 = (s_1 = 1, s_2 = 1, s_3 = 1)$
PC $TR_6 : p_3 = false,$ $TC_6 = (s_1 = 3, s_2 = 4, s_3 = 5)$

PC $TR_7 : p_4 = true,$ $TC_7 = (s_1 = 34, s_2 = 34, s_3 = 30)$
PC $TR_8 : p_4 = false,$ $TC_8 = (s_1 = 3, s_2 = 4, s_3 = 5)$

Minimized because TC_8 satisfies $TR_2,\ TR_4,\ TR_6, TR_8$
PC $TR_1 : p_1 = true,$ $TC_1 = (s_1 = -1, s_2 = -1, s_3 = -1)$
PC $TR_2 : p_1 = false,$ TC_8

PC $TR_3 : p_2 = true,$ $TC_3 = (s_1 = 1, s_2 = 1, s_3 = 2)$
PC $TR_4 : p_2 = false,$ TC_8

PC $TR_5 : p_3 = true,$ $TC_5 = (s_1 = 1, s_2 = 1, s_3 = 1)$
PC $TR_6 : p_3 = false,$ TC_8

PC $TR_7 : p_4 = true,$ $TC_7 = (s_1 = 34, s_2 = 34, s_3 = 30)$
PC $TR_8 : p_4 = false,$ $TC_8 = (s_1 = 3, s_2 = 4, s_3 = 5)$

(b) It can be observed that the set of test cases for NC and PC are the same for the graph. This is due to all of the Boolean condition nodes have each one leaf node which is an assignment node. If we have an assignment node we can access from two different condition nodes, we only need to access it once for NC, but evaluate both statements in the condition node for PC.

(c) $p_1* = \{n_1, n_2, n_4\}$ (Since we have an assignment node n_4 which we can access from two different condition nodes, we don't need to visit n_1, n_2 and n_1, n_3, n_4 separately.)
$p_3 = \{n_1, n_3, n_5, n_6\}$
$p_4 = \{n_1, n_3, n_5, n_7, n_8\}$
$p_5 = \{n_1, n_3, n_5, n_7, n_9\}$
NC $TR = \{p_1, p_3, p_4, p_5\}$

NC $TC_1 : s_1 = -1, s_2 = -1, s_3 = -1$
NC $TC_3 : s_1 = 1, s_2 = 1, s_3 = 1$
NC $TC_4 : s_1 = 34, s_2 = 34, s_3 = 30$
NC $TC_5 : s_1 = 3, s_2 = 4, s_3 = 5$

p = predicate
$p_1 : s_1 <= 0 \mid s_2 <= 0 \mid s_3 <= 0$
$p_2 : s_1 + s_2 <= s_3 \mid s_2 + s_3 <= s_1 \mid s_1 + s_3 <= s_2$
$p_3 : s_1 == s_2\ \&\ s_2 == s_3$
$p_4 : s_1 == s_2 \mid s_2 == s_3 \mid s_1 == s_3$

PC $TR_1 : p_1 = true,\quad TC_1 = (s_1 = -1, s_2 = -1, s_3 = -1)$
PC $TR_2 : p_1 = false,\quad TC_2 = TC_8$

PC $TR_3 : p_2 = true,\quad TC_3 = (s_1 = 1, s_2 = 1, s_3 = 2)$
PC $TR_4 : p_2 = false,\quad TC_4 = TC_8$

PC $TR_5 : p_3 = true,\quad TC_5 = (s_1 = 1, s_2 = 1, s_3 = 1)$
PC $TR_6 : p_3 = false,\quad TC_6 = TC_8$

PC $TR_7 : p_4 = true,\quad TC_7 = (s_1 = 34, s_2 = 34, s_3 = 30)$
PC $TR_8 : p_4 = false,\quad TC_8 = (s_1 = 3, s_2 = 4, s_3 = 5)$

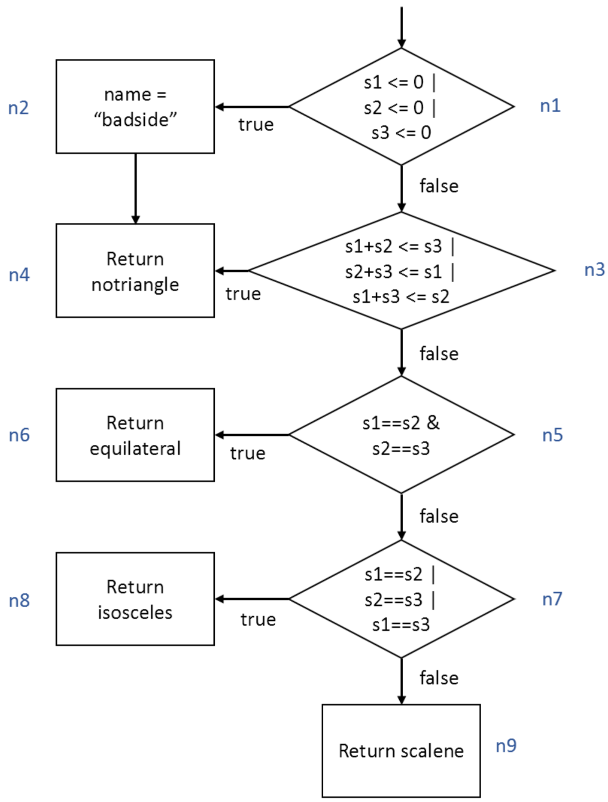

Figure 2: A modified condensation graph.

2.2

(a & b)

$$
\begin{aligned}
TR\ n_1\ 1_1 : s_1 <= 0\ True, &\qquad TC\ 1 = (s_1 = -1, s_2 = -1, s_3 = -1)\\
TR\ n_1\ 1_2 : s_1 <= 0\ False, &\qquad TC\ 2 = (s_1 = 1, s_2 = 1, s_3 = 1)\\
TR\ n_1\ 1_1 : s_1 <= 0\ True, &\qquad TC\ 1 = (s_1 = -1, s_2 = -1, s_3 = -1)\\
TR\ n_1\ 1_2 : s_1 <= 0\ False, &\qquad TC\ 2 = (s_1 = 1, s_2 = 1, s_3 = 1)\\
TR\ n_1\ 2_1 : s_2 <= 0\ True, &\qquad TC\ 3 = TC\ 1\\
TR\ n_1\ 2_2 : s_2 <= 0\ False, &\qquad TC\ 4 = TC\ 2\\
TR\ n_1\ 3_1 : s_3 <= 0\ True, &\qquad TC\ 5 = TC\ 1\\
TR\ n_1\ 3_2 : s_3 <= 0\ False, &\qquad TC\ 6 = TC\ 2\\
TR\ n_3\ 1_1 : s_1 + s_2 <= s_3\ True, &\qquad TC\ 7 = (s_1 = 1, s_2 = 1, s_3 = 2)\\
TR\ n_3\ 1_1 : s_1 + s_2 <= s_3\ False, &\qquad TC\ 8 = TC_2\\
TR\ n_3\ 2_1 : s_2 + s_3 <= s_1\ True, &\qquad TC\ 9 = (s_1 = 2, s_2 = 1, s_3 = 1)\\
TR\ n_3\ 2_1 : s_2 + s_3 <= s_1\ False, &\qquad TC\ 10 = TC_2\\
TR\ n_3\ 3_1 : s_1 + s_3 <= s_2\ True, &\qquad TC\ 11 = (s_1 = 1, s_2 = 2, s_3 = 1)\\
TR\ n_3\ 3_2 : s_1 + s_3 <= s_2\ False, &\qquad TC\ 12 = TC_2\\
TR\ n_5\ 1_1 : s_1 == s_2\ True, &\qquad TC\ 13 = TC\ 2\\
TR\ n_5\ 1_2 : s_1 == s_2\ False, &\qquad TC\ 14 = (s_1 = 34, s_2 = 30, s_3 = 34)\\
TR\ n_5\ 2_1 : s_2 == s_3\ True, &\qquad TC\ 15 = TC\ 2\\
TR\ n_5\ 2_2 : s_2 == s_3\ False, &\qquad TC\ 16 = TC\ 14\\
TR\ n_7\ 1_1 : s_1 == s_2\ True, &\qquad TC\ 17 = (s_1 = 34, s_2 = 34, s_3 = 30)\\
TR\ n_7\ 1_2 : s_1 == s_2\ False, &\qquad TC\ 18 = (s_1 = 3, s_2 = 4, s_3 = 5)\\
TR\ n_7\ 2_1 : s_2 == s_3\ True, &\qquad TC\ 19 = (s_1 = 30, s_2 = 34, s_3 = 34)\\
TR\ n_7\ 2_2 : s_2 == s_3\ False, &\qquad TC\ 20 = TC\ 18\\
TR\ n_7\ 3_1 : s_1 == s_3\ True, &\qquad TC\ 21 = TC\ 14\\
TR\ n_7\ 3_2 : s_1 == s_3\ False, &\qquad TC\ 22 = TC\ 18
\end{aligned}
$$

2.3

$s_1 <= 0$	$s_2 <= 0$	$s_3 <= 0$	p_1	c determines p
T	T	T	T	
F	T	T	T	
T	F	T	T	
F	F	**T**	T	A1
T	T	F	T	
F	**T**	F	T	B1
T	F	F	T	C1
F	F	F	F	A2, B2, C2

Table 1: Predicate 1 Truth table. When $A1(s_1 <= 0) = A1(s_2 <= 0) = F$, then $(s_3 <= 0)$ determines p_1. When $B1(s_1 <= 0) = B1(s_3 <= 0) = F$, then $(s_2 <= 0)$ determines p_1. When $C1(s_2 <= 0) = C1(s_3 <= 0) = F$, then $(s_1 <= 0)$ determines p_1.

$s_1 + s_2 <= s_3$	$s_2 + s_3 <= s_1$	$s_1 + s_3 <= s_2$	p_2	c determines p
T	T	T	T	
F	T	T	T	
T	F	T	T	
F	F	**T**	T	D1
T	T	F	T	
F	**T**	F	T	E1
T	F	F	T	F1
F	F	F	F	D2, E2, F2

Table 2: Predicate 2 Truth table. When $D1(s_1 + s_2 <= s_3) = D1(s_2 + s_3 <= s_1) = F$, then $(s_1 + s_3 <= s_2)$ determines p_2. When $E1(s_1 + s_2 <= s_3) = E1(s_1 + s_3 <= s_2) = F$, then $(s_2 + s_3 <= s_1)$ determines p_2. When $F1(s_2 + s_3 <= s_1) = F1(s_1 + s_3 <= s_2) = F$, then $(s_1 + s_2 <= s_3)$ determines p_2.

$s_1 == s_2$	$s_2 == s_3$	p_3	c determines p
T	**T**	T	G1, H1
T	**F**	F	G2
F	T	F	H2
F	F	F	

Table 3: Predicate 3 Truth table. $G1(s_1 == s_2) = G1(s_2 == s_3) = T$ then $(s_2 == s_3)$ determines p_3. $H1(s_1 == s_2) = H1(s_2 == s_3) = T$ then $(s_1 == s_2)$ determines p_3.

$s_1 == s_2$	$s_2 == s_3$	$s_1 == s_3$	p_4	c determines p
T	T	T	T	
F	T	T	T	
T	F	T	T	
F	F	**T**	T	I1
T	T	F	T	
F	**T**	F	T	J1
T	F	F	T	K1
F	F	F	F	I2, J2, K2

Table 4: Predicate 4 Truth table. When $I1(s_1 == s_2) = I1(s_2 == s_3) = F$, then $(s_1 == s_3)$ determines p_4. When $J1(s_1 == s_2) = J1(s_1 == s_3) = F$, then $(s_2 == s_3)$ determines p_4. When $K1(s_2 == s_3) = K1(s_1 == s_3) = F$, then $(s_1 == s_2)$ determines p_4.

For RACC of clause $s_3 <= 0$ there are 1 possible test suite ($A1 = (s_1 = 1, s_2 = 1, s_3 = -1)$, $A2 = (s_1 = 3, s_2 = 4, s_3 = 5)$). For clause $s_2 <= 0$ there are 1 possible test suite ($B1 = (s_1 = 1, s_2 = -1, s_3 = 1)$, $B2 = (s_1 = 3, s_2 = 4, s_3 = 5)$). For clause $s_1 <= 0$ there are 1 possible test suite ($C1 = (s_1 = 1, s_2 = -1, s_3 = -1)$, $C2 = (s_1 = 3, s_2 = 4, s_3 = 5)$).

For RACC of clause $s_1 + s_3 <= s_2$ there are 1 possible test suite ($D1 = (s_1 = 1, s_2 = 2, s_3 = 1)$, $D2 = (s_1 = 1, s_2 = 1, s_3 = 1)$). For clause $s_2 + s_3 <= s_1$ there are 1 possible test suite ($E1 = (s_1 = 2, s_2 = 1, s_3 = 1)$, $E2 = (s_1 = 1, s_2 = 1, s_3 = 1)$). For clause $s_1 + s_2 <= s_3$ there are 1 possible test suite ($F1 = (s_1 = 1, s_2 = 1, s_3 = 2)$, $F2 == (s_1 = 1, s_2 = 1, s_3 = 1)$).

For RACC of clause $s_2 == s_3$ there are 1 possible test suite ($G1 = (s_1 = 1, s_2 = 1, s_3 = 1)$, $G2 = (s_1 = 34, s_2 = 34, s_3 = 30)$). For clause $s_1 == s_2$ there are 1 possible test suite ($H1 = (s_1 = 1, s_2 = 1, s_3 = 1)$, $H2 = (s_1 = 30, s_2 = 34, s_3 = 34)$).

For RACC of clause $s_1 == s_3$ there are 1 possible test suite ($I1 = (s_1 = 34, s_2 = 30, s_3 = 34)$, $I2 = (s_1 = 3, s_2 = 4, s_3 = 5)$). For clause $s_2 == s_3$ there are 1 possible test suite ($J1 = (s_1 = 30, s_2 = 34, s_3 = 34)$, $J2 = (s_1 = 3, s_2 = 4, s_3 = 5)$). For clause $s_1 == s_2$ there are 1 possible test suite ($K1 = (s_1 = 34, s_2 = 34, s_3 = 30)$, $K2 == (s_1 = 3, s_2 = 4, s_3 = 5)$).

3 Looping

(a) A condensation graph for the code:

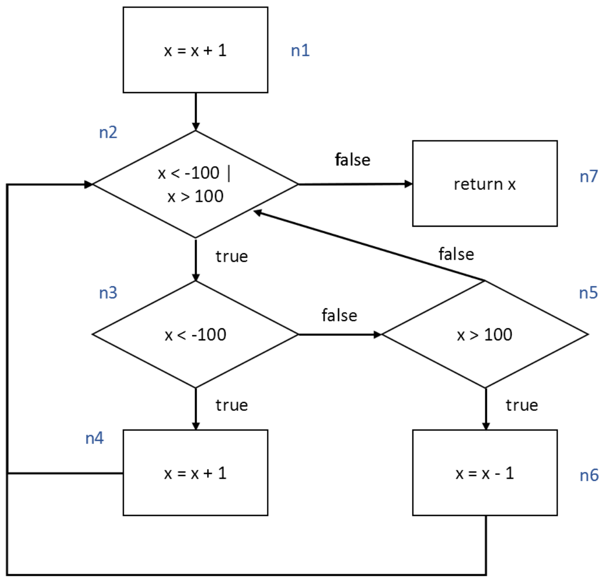

Figure 3: A condensation graph.

(b) There are four possible path without looping the while-loop more than once: $p_1 = \{n_1, n_2, n_7\}$, $p_2 = \{n_1, n_2, n_3, n_4, n_2, n_7\}$, $p_3 = \{n_1, n_2, n_3, n_5, n_6, n_2, n_7\}$, and $p_4 = \{n_1, n_2, n_3, n_5, n_2, n_7\}$. But there is a path that visits all nodes and gives 100% NC, which is $p_5 = \{n_1, n_2, n_3, n_4, n_2, n_3, n_5, n_6, n_2, n_7\}$, and this is the minimized set TR of test requirements $TR = \{p_5\}$.

(c) There is no test case that satisfies the minimized TR. Because when deriving the minimized TR, we did not consider whether all nodes in each of the test requirements can be reached with a single input. However, for the TR p_2 and p_3, we can derive a test suite for it, which is $TC1 : x = -102$ and $TC2 : x = 100$.

(d) There exists a test suite that satisfies the minimized set of test requirements of predicate coverage. The test suite for PC would also cover the edge between n_5 and n_2.

4 Self-Assessment

Question	NC	EC	PC	CC	RACC
Do you have a test case that represents a valid scalene triangle?	1	1	1	1	1
Do you have a test case that represents a valid equilateral triangle?	1	1	1	1	1
Do you have a test case that represents a valid isosceles triangle?	1	1	1	1	1
Do you have at least three test cases that represent valid isosceles triangles such that you have tried all three permutations of two equal sides?	0	0	0	1	1
Do you have a test case in which one side has a zero value?	0	0	0	0	0
Do you have a test case in which one side has a negative value?	1	1	1	1	1
Do you have a test case with three integers such that the sum of two is equal to the third?	1	1	1	1	1
Do you have at least three test cases in category 7 such that you have tried all three permutations where the length of one side is equal to the sum of the lengths of the other two sides?	0	0	0	1	1
Do you have a test case with three integers greater than zero such that the sum of two numbers is less than the third?	1	1	1	1	1
Do you have at least three test cases in category 9 such that you have tried all three permutations?	0	0	0	1	1
Do you have a test case in which all sides are zero?	0	0	0	0	0
Do you have at least one test case specifying non-integer values or does this not make sense?	-	-	-	-	-
Do you have at least one test case specifying the wrong number of values (2 or less, four or more) or does this not make sense?	-	-	-	-	-
For each test case, did you specify the expected output from the program in addition to the input values?	0	0	0	0	0
Sum	6	6	6	9	9

Table 5: Self-assessment

From our table: CC and RACC has the highest coverage sum. In this example NC, EC and PC has the same coverage level where we do not test more than one permutation of inputs for each TC. We can also see that we have not used a single test with a zero in it. Instead we used minus one for all the test that cover a TR with $s <= 0$.