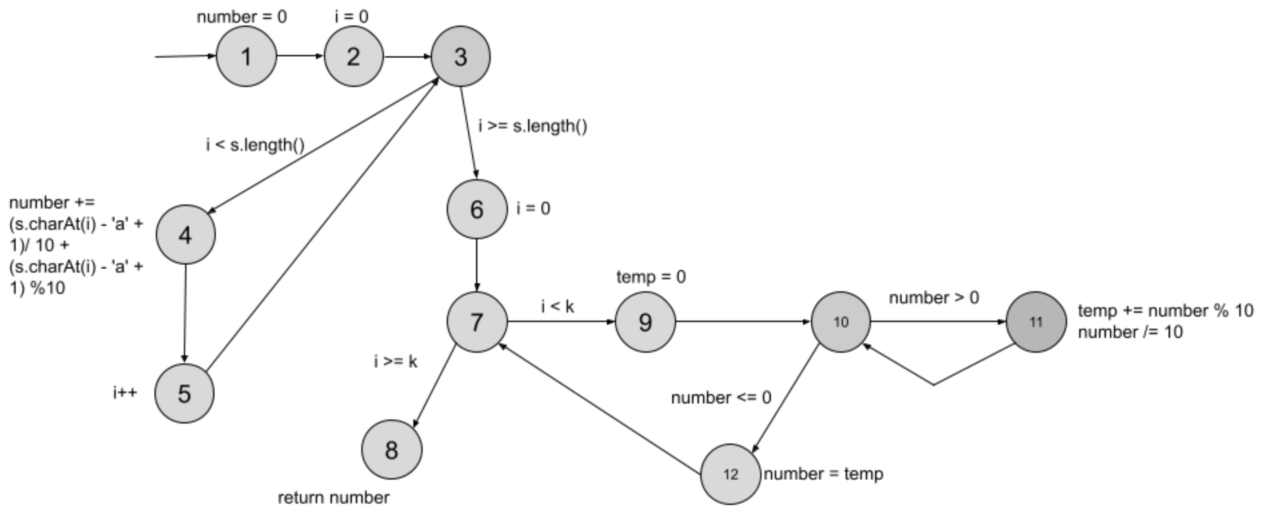
System Test Plan
Graph-Based Coverage

By: Cail Keeling

Table of Contents

1. Control Flow

1.1. getLucky1 Control Flow Graph



2. Test Requirements

2.1. getLucky1 Coverage Criteria – variable “number”

Defs:

def(1) = { number }
def(4) = { number }
def(11) = { number }
def(12) = { number }

Uses:

use(4) = { number }
use(8) = { number }
use(10,11) = { number }
use(11) = { number }
use(10,12) = { number }

du-paths:

1.	du(1,4)	= [1,2,3,4]	****4 5
2.	du(1,8)	= [1,2,3,6,7,8]	****3
3.	du(1, (10,11))	= [1,2,3,6,7,9,10,11]	****1
4.	du(1,11)	= [1,2,3,6,7,9,10,11]	
5.	du(1, (10,12))	= [1,2,3,6,7,9,10,12]	****2
6.	du(4,4)	= [4,5,3,4]	****4
7.	du(4,8)	= [4,5,3,6,7,8]	**** 6
8.	du(4, (10,11))	= [4,5,3,6,7,9,10,11]	****4
9.	du(4,11)	= [4,5,3,6,7,9,10,11]	
10.	du(4, (10,12))	= [4,5,3,6,7,9,10,12]	****5
11.	du(11, (10,11))	= [11,10,11]	****1
12.	du(11,11)	= [11,10,11]	
13.	du(11, (10,12))	= [11,10,12]	****1 2 4
14.	du(12,8)	= [12,7,8]	****1 2 4
15.	du(12,11)	= [12,7,9,10,11]	****2
16.	du(12, (10,11))	= [12,7,9,10,11]	
17.	du(12, (10,12))	= [12,7,9,10,12]	****

Test Path ID	Test Path	Covered TR	Feasible?
1	1,2,3,6,7,9,10,11,10,11,10,12,7,8	3/4,11/12,13,14	No: s.length cannot equal 0 (which is necessary for 1-11 to occur) and have number > 0 at the start of the second for loop
2	1,2,3,6,7,9,10,12,7,9,10,11,10,12,7,9,10,12,7,8	5,15/16,13,17,14	No: s.length cannot equal 0 (which is necessary for 1-12 to occur) and have number > 0 at the start of the second for-loop
3	12,3,6,7,8	2	Yes: s=””, k=0
4	1,2,3,4,5,3,4, 4,5,3,6,7,9,10,11,10,12,7,8	1,6,8/9,13,14	Yes: s.length = 2, k=1
5	1,2,3, 4,5,3,6,7,9,10,12,7,8	5,10,14	No: length of s cannot be 1 (which is necessary to trigger du(4, (10,12)) and have number be <= 0
6	1,2,3,4,5,3,6,7,8	1,7	Yes: s=any 1 letter, k=0

3. Test Results with Traceability

Test ID	Test Path	Input	Observed Output	Result
3	1-8	s=””, k=0	0	Pass
4	1-4,4-4,4-11,11-12, 12-8	s=”aa”, k=1	2	Pass
6	1-4, 4-8	s=”a”, k=0	1	Pass

Effectiveness:
Does your test suite detect this program is faulty? **No**
If you have failing tests, do your tests help you figure out what is wrong? **NULL**
If you have do not have any failing tests, what is the limitation or gap in your du-paths test paths? **My test paths do not cover any k or s size greater than 2, and I did not have the 4 node loop back to itself multiple times and have 11 loop back to itself multiple times in the same test path.**

4. Graph Coverage Effectiveness

4.1. Different Implementation

Test ID	Test Path	Input	Observed Output	Result
3	1-8	s=””, k=0	0	Pass
4	1-4,4-4,4-11,11-12, 12-8	s=”aa”, k=1	2	Pass
6	1-4, 4-8	s=”a”, k=0	1	Pass

Effectiveness:
Does your test suite detect this program is faulty? **No**
If you have failing tests, explain how tests derived from a different control flow graph were still effective? **NULL**
If you have do not have any failing tests, what is the limitation or gap in your du-paths test paths created for one program that does not seem to translate to a different implementation?
My tests for getLucky1 were also fruitless and bear possibly the same faults, so I am unable to speculate what might’ve worked for getLucky1 and not getLucky2.