

# TESTING OF VENDING MACHINE

## PROJECT REPORT

FALL 2016 | Software Testing and Analysis | 11/30/16

BY: KOMAL SURENDRA PARCHAND (A20373541)

## The Abstract

The purpose of software testing and analysis is to provide a software to the end user with a quality product. To achieve or maintain the quality of software there are various testing techniques incorporated for example: Model -Based testing, Multiple- condition testing, Branch testing, Object oriented testing etc. With the help of these testing methodologies testers can determine the quality of code which in turn helps to determine the quality of software. In this project Report we are about to observe the testing of simple vending machine through various methods.

## Table of Content

1. Introduction.....	04
2. Testing Methodology .....	05
3. Observations	
a. Model Based Testing.....	06
b. Ghost Transition.....	09
c. Multiple Condition.....	10
4. Result .....	11
5. Test Suite and Execution .....	23
6. Conclusion.....	24
7. Source code.....	25

## Introduction

The goal of this project is to test the “Vending Machine “class exhibiting state transition behavior which has been described through EFSM diagram. The class of vending machine is defined on basis of states which include: idle, coin inserted, sugar, no\_large\_cups and no\_small\_cups. The Purpose of the vending machine is to dispose cup of tea (large or small) with sugar or without sugar based on the conditions. To test this project a testing environment developed in C++ language was used containing class drivers using following methods of testing:

- a. Model – Based Testing
- b. Ghost Transition
- c. Multiple condition testing

# Testing Methodology

## 1. Model Based Testing

Model-based testing is a software testing technique in which the test cases are derived from a model that describes the functional aspects of the system under test. Generally, it makes use of a model to generate tests that includes both offline and online testing. Some advantages of model-based testing are:

- Higher level of Automation is achieved.
- Exhaustive testing is possible.
- Changes to the model can be easily tested.

## 2. Ghost Transition Testing

These testing for part of model based testing wherein the only difference that lies between ghost transition testing and all pair testing is that the transitions which are not defined/ visible for a state (called as ghost transition) when triggered will remain in same state as the original. Such type of transitions when triggered does not affect the change in state

## 3. Multiple Condition Testing

Development of tests by white box testing method in which test cases are designed to execute combinations of single condition outcomes and each of the conditions or branches are evaluated.

## Observations

### Model Based Testing

Below are the observations when carrying out Model based testing (all -pair transitions)

Transaction ID	Description
T1	Vendingmachine / $k=0, k1=0, t=0, price=0$
T2	Insert_large_cups(n) [ $n>0$ ] / $k=k+n$
T3	Insert_small_cups(n) [ $n>0$ ] / $k1=k1+n$
T4	Set_price(p)[ $p>0$ ] / price = p
T5	Dispose()
T6	Coin[ $t+25<price$ ]/ $t=t+25$
T7	Coin[ $(t+25\geq price)\ \&\&\ (price>0)$ ]/ $s=0;t=0$
T8	Insert_large_cups(n)[ $n>0$ ]/ $k=n$
T9	Insert_small_cups(n)[ $n>0$ ]/ $k1=n$
T10	Cancel/ return coins
T11	Tea[ $(k1>1)\ \&\&\ (s==2)$ ]/dispose small cup of tea; $k1=k1-1$
T12	Tea[ $(k>1)\ \&\&\ (s==1)$ ] / dispose large cup of tea; $k= k-1$
T13	Tea[ $(k>1)\ \&\&\ (s==1)$ ] / dispose large cup of tea with sugar; $k= k-1$
T14	Cancel / return coins
T15	Tea[ $(k1>1)\ \&\&\ (s==2)$ ]/dispose small cup of tea with sugar; $k1=k1-1$
T16	Coin / return coin

T17	Small_cup/ s=2
T18	Large_cup/ s=1
T19	Large_cup/ s=1
T20	Coin/ return coin
T21	Small_cup/ s=2
T22	Sugar()
T23	Sugar()
T24	Tea[(k==1) && (s==1)]/ dispose large cup of tea; k=k-1
T25	Tea[(k==1) && (s==1)]/ dispose large cup of tea with sugar; k= k-1
T26	Tea[(k1==1)&&(s==2)]/ dispose small cup of tea ; k1= k1-1
T27	Tea[(k1==1) &&(s==2)]/ dispose small cup of tea with sugar; k1=k1-1
T28	Coin/ return coin
T29	Coin/ return coin

All pair transitions for each state are mentioned below:

Pair for state “no small cups”

(T25, T28) (T25, T9)
(T27, T28) (T27, T9)
(T28, T28) (T28, T9)

Pair for state “no large cups”

(T24, T8) (T24, T29)
(T26, T8) (T26, T29)
(T29, T8) (T29, T29)

Pair for state “Sugar”

(T16, T13) (T16, T14) (T16, T15) (T16, T16) (T16, T17) (T16, T18) (T16, T23) (T16, T26) (T16, T27)
(T17, T13) (T17, T14) (T17, T15) (T17, T16) (T17, T17) (T17, T18) (T17, T23) (T17, T26) (T17, T27)
(T18, T13) (T18, T14) (T18, T15) (T18, T16) (T18, T17) (T18, T18) (T18, T23) (T18, T26) (T18, T27)
(T22, T13) (T22, T14) (T22, T15) (T22, T16) (T22, T17) (T22, T18) (T22, T23) (T22, T26) (T22, T27)

Pair for state “Coins Inserted”

(T7, T10) (T7, T11) (T7, T12) (T7, T19) (T7, T20) (T7, T21) (T7, T22) (T7, T24) (T7, T25)
(T19, T10) (T19, T11) (T19, T12) (T19, T19) (T19, T20) (T19, T21) (T19, T22) (T19, T24) (T19, T25)
(T20, T10) (T20, T11) (T20, T12) (T20, T19) (T20, T20) (T20, T21) (T20, T22) (T20, T24) (T20, T25)
(T21, T10) (T21, T11) (T21, T12) (T21, T19) (T21, T20) (T21, T21) (T21, T22) (T21, T24) (T21, T25)
(T23, T10) (T23, T11) (T23, T12) (T23, T19) (T23, T20) (T23, T21) (T23, T22) (T23, T24) (T23, T25)

Pairs for state “Idle”

(T1, T2) (T1, T3) (T1, T4) (T1, T5) (T1, T6) (T1, T7)
(T2, T2) (T2, T3) (T2, T4) (T2, T5) (T2, T6) (T2, T7)
(T3, T2) (T3, T3) (T3, T4) (T3, T5) (T3, T6) (T3, T7)
(T4, T2) (T4, T3) (T4, T4) (T4, T5) (T4, T6) (T4, T7)
(T6, T2) (T6, T3) (T6, T4) (T6, T5) (T6, T6) (T6, T7)



(T8, T2) (T8, T3) (T8, T4) (T8, T5) (T8, T6) (T8, T7)
(T9, T2) (T9, T3) (T9, T4) (T9, T5) (T9, T6) (T9, T7)
(T10, T2) (T10, T3) (T10, T4) (T10, T5) (T10, T6) (T10, T7)
(T11, T2) (T11, T3) (T11, T4) (T11, T5) (T11, T6) (T11, T7)
(T12, T2) (T12, T3) (T12, T4) (T12, T5) (T12, T6) (T12, T7)
(T13, T2) (T13, T3) (T13, T4) (T13, T5) (T13, T6) (T13, T7)
(T14, T2) (T14, T3) (T14, T4) (T14, T5) (T14, T6) (T14, T7)
(T15, T2) (T15, T3) (T15, T4) (T15, T5) (T15, T6) (T15, T7)

### Ghost Transition testing

For “IDLE” state:

insert_small_cups	sugar
insert_large_cups	tea
set_price	small_cup
cancel	large cup
coin	

For “Coin Inserted” state:

dispose	insert_large_cups
insert_small_cups	set price

For “Sugar” state:

insert_small_cups	set_price
insert_large_cups	dispose

For “No Large cups” state:

small_cup	set_price
large_cup	cancel

sugar

dispose

tea

insert\_small\_cups

For “no Small cups:” state:

small\_cup

sugar

large\_cup

tea

set\_price

dispose

cancel

insert\_large\_cups

### Multiple Condition testing

- 1) ( t+25>=price )&&(price>0)
- 2) (t+25<price)
- 3) (x>1)&&(x<6)
- 4) (x==2)||(x==3) – Small cup
- 5) (x==2)||(x==3) – Large Cup
- 6) ((x==2)||(x==3)) - Sugar
- 7) (x==2) && (k1>1) && (s==2) – Small Cup of tea
- 8) (x==2)&&(k>1)&&(s==1) – Large Cup of tea
- 9) (x==2)&&(k==1)&&(s==1) – large cup of tea
- 10) (x==2)&&(k1==1)&&(s==2) – small cup of tea
- 11) (x==3)&&(k1==1)&&(s==2) – small cup with sugar
- 12) (x==3)&&(k==1)&&(s==1) – large cup with sugar
- 13) (x==3) &&(k1>1)&&(s==2) – small cup with sugar
- 14) (x==3) &&(k>1) &&(s==1) – large cup with sugar
- 15) (x==1)&&(n>0) – Large cup
- 16) (x==5)&&(n>0) – Large Cup
- 17) (x==1)&&(n>0) – Small Cup
- 18) (x==4)&&(n>0)- Small Cup
- 19) (x==1)&&(p>0)
- 20) ((x==1)) – Dispose
- 21) (x==1) – Coin
- 22) (x==2) ||(x==3) – cancel

## Result

The detailed analysis of test suite, test driver and the proof of execution are present in 3 separate documents in the zip folder attached with title of documents Ts.txt, VendingMachine.exe and, Test Execution folder respectively. Based on the testing carried out for vending machine state based- code below are some of the key results that should be noted for each of the testing methodology

### Model Based Testing:

- Ghost Transition testing

Test Cases Covered	States
Test Case 37	idle
Test Case 38	coin inserted
Test Case 39	Sugar
Test Case 40	No_large_cups
Test Case 41	No_small_cups

- All - Pair testing

Test Case No	Transition covered	Transition Pair
Test Case 1	1,2,3,4,7,21,20,11,7,10, 5	(1,2) (2,3) (3,4) (4,7) (7,21) (21,20) (20,11) (11,7) (7,10) (10,5)
Test Case 2	1,2,2,3,3,4,7,19,12,7,19,20,24,29,8,5	(1,2) (2,2) (2,3) (3,3) (3,4) (4,7) (7,19) (19,12) (12,7) (7,19) (19,20) (20,24) (24,29) (29,8) (8,7) (8,5)
Test Case 3	1,2,4,2,6,2,3,6,5	(1,2) (2,4) (4,2) (2,6) (6,2) (2,3) (3,6) (6,5)
Test Case 4	1,2,3,4,6,7,19,22,18,26,29,8,5	(1,2) (2,3) (3,4) (4,6) (6,7) (7,19) (19,22) (22,18) (18,26) (26,29) (29,8) (8,5)
Test Case 5	1,2,4,6,7,24,8,7,10,5	(1,2) (2,4) (4,6) (6,7) (7,24) (7,8) (7,19) (19,24) (24,8) (8,6) (6,7) (7,10) (10,5)
Test Case 6	1,2,4,6,7,19,20,22,18,16,18,26,29,29,8,5	(1,2) (2,4) (4,6) (6,7) (7,19) (19,20) (20,22) (22,18) (18,16) (16,18) (18,26) (26,29) (29,29) (29,8) (8,5)
Test Case 7	1,2,4,6,7,12,5	(1,2) (2,4) (4,6) (6,7) (7,12) (7,5) (12,5)
Test Case 8	1,3,4,6,6,7,11,5	(1,3) (3,4) (4,6) (6,6) (6,7) (7,11) (7,21) (21,11) (11,5)

Test Case 9	1,3,4,6,7,10,7,21,25,9,3,7,22,23,22,17,16,27,9,6,5	(1,3) (3,4)(4,6)(6,7)(7,10) (10,7)(10,6)(6,7) (7,21) (21,25)(25,9)(9,3)(3,7)(3,6)(6,7)(7,22) (22,23)(23,22)(22,17)(17,16)(16,27)(27,9)(9,6)(6,5)
Test Case 10	1,3,4,6,7,21,25,28,9,4,5	(1,3) (3,4) (4,6) (6,7) (7,21) (21,25) (25,28) (28,9) (9,4) (4,5)
Test Case 11	1,3,2,4,6,7,19,22,26,8, 6,6,4, 7,22,17, 27,28,28,9,3,5	(1,3) (3,2) (2,4) (4,6) ((6,7) (7,19) (19,22) (22,26) (26,8) (8,6) (6,6) (6,4) (4,7) (7,22) (22,17) (17,27) (27,28) (28,28) (28,9) (9,3) (3,5)
Test Case 12	1,2,3,4,6,7,19,21,21,22,16,17,18,27,9,2,5	(1,2) (2,3) (3,4) (4,6) (6,7) (7,19) (19,21) (21,21) (21,22) (22,16) (16,17) (17,18) (18,27) (18,17) (17,27) (27,8) (27,9) (9,2) (2,5)
Test Case 13	1,4,6,2,7,21,19,19,20,19,24,8,3,7,22,15,5	(1,4)(4,6)(6,2)(2,7)(2,3)(3,6)(6,7)(7,21)(21,19) (19,19)(19,20)(20,19)(19,24) (24,8)(8,3)(3,7) (3,6) (6,22)(6, 24) (6,7)(7,22)(22,15) (22,17)(17,15)(15,5)
Test Case 14	1,7,4,6,3,7,20,20,25,9,7,25,9,3,2,4,4,3,7,10,4,5	(1,7) (1,4) (4,6)(6,3)(3,7)(7,20)(20,20)(20,25) (20,21)(21,25)(25,9) (9,7) (9,6) (6,7)(7,25)(7,21) (21,25) (25,9)(9,3)(3,2)(2,4) (4,4) (4,3) (3,7) (3,6)(6,7)(7,10)(10,4)(4,5)
Test Case 15	1,5,1,4,1,7,1,1,20,5	(1,5) (5,1) (1,4) (4,1) (1,7) (7,1) (1,1) (1,20) (20,5)
Test Case 16	1,6,3,6,4,7,21,10,7,21,11,2,7,19,21,12,5	(1,6) (6,3) (3,6) (6,4) (4,7) (7,21) (21,10) (10,7) (10,6) (6,7) (7,21) (21,11) (11,2) (2,7) (7,19) (19,21) (21,12) (12,5)
Test Case 17	1,6,7, 4,2,7,22,14,4,7,22,13,5	(1,6) (1,2) (2,4) (4,6) (6,7) (7,4) (7,10) (10,4) (4,2) (2,7) (2,6) (6,7) (7,22) (22,14) (14,4) (4,7) (4,6) (6,7) (7,22) (22,13) (22,18) (18,13) (13,5)
Test Case 18	1,6,7,2,3,4,6,7,19,21,10,11,12,13,7,19,10,3,4,7,19,11,3,7,19,25,9,7,20,10,2,7,20,12,6,5	(1,6) (1,7)(1,2)(3,2)(2,4)(4,6)(6,7)(7,11) (7,12)(7,13) (7,15) (7,2)(7,3)(7,4)(7,19) (19,21)(21,10)(10,6)(6,7) (7,12)(7,13) (7,19)(19,10)(10,3)(3,4)(4,7) (4,6)(6,7)(7,19)(19,11)(19,21)(21,11)(11,3) (3,7)(3,6)(6,7)(7,19)(19,25) (19,21) (21,25)(25,9)(9,7)(9,6)(6,7)(7,20)(20,10)(10,2)(2,7)(2,6)(6,7)(7,20)(20,21)(21,20)(20,12)(12,6)(6,5)
Test Case 19	1,2,3,7,19,20,21,24,8,7,22,23,25,9,6,7,22,23,24,29,8,5	(1,2)(2,3)(3,7)(3,4)(4,6)(6,7)(7,19)(19,20)(20,21) (21,24) (21,19)(19,24)(24,8)(8,7)(8,6)(6,7)(7,22)(22,23) (23,25)(23,21)(21,25)(25,9)(9,6)((6,7)(7,22) (22,23) (23,24) (23,19)(19,24)(24,29)(29,8)(8,5)

Test Case 20	1,6,7,3,4,3,7,22,17,18,18,17,17,15,5	(1,6) (6,7) (7,3) (3,7) (7,22) (22,17) (17,18) (18,18) (18,17) (17,17) (17,15) (15,5)
Test Case 21	1,4,2,6,7,21,22,23,10,6,2,7,19,21,24,8,2,5	(1,4) (4,2) (2,6) (6,7) (7,21) (21,22) (22,23) (23,10) (10,6) (6,2) (2,7) (7,19) (19,21) (21,24) (21,19) (19,24) (24,8) (8,2) (2,5)
Test Case 22	1,4,6,2,7,19,20,24,8,4,7,11,4,3,7,11,6,5	(1,4) (4,6) (6,2) (2,7) (7,19) (19,20) (20,24) (24,8) (8,4) (4,7) (7,11) (7,21) (21,11) (11,4) (4,3) (3,7) (7,21) (21,11) (11,6) (6,5)
Test Case 23	1,2,4,6,7,19,22,16,13,3,7,21,20,17,13,7,22,16,16,14,5	(1,2)(2,4)(4,6)(6,7)(7,19)(19,22)(22,16) (16,13)(13,3)(3,7)(7,21)(21,20)(20,17)(17,13) (20,22)(22,17)(17,18)(18,13) (13,7)(7,22)(22,16)(16,16)(16,14)(14,5)
Test Case 24	1,3,4,6,7,21,20,22,23,11,7,12,2,3,7,21,20,22,16,15,3,5	(1,3) (3,4) (4,6) (6,7) (7,21) (21,20) (20,22) (22,23) (23,11) (11,7) (11,6) (6,7) (7,12) (12,2) (2,3) (3,7) (7,21) (21,20) (20,22) (22,16) (16,15) (15,3) (3,5)
Test Case 25	1,2,4,6,7,19,20,22,23,12,3,5	(1,2) (2,3) (3,4) (4,6) (6,7) (7,19) (19,20) (20,22) (22,23) (23,12) (12,3) (3,5)
Test Case 26	1,2,4,6,7,19,20,22,18,16,26,8,5	(1,2) (2,4) (4,6) (6,7) (7,19) (19,20) (20,22) (22,18) (18,16) (16,26) (26,8) (8,5)
Test Case 27	1,2,3,4,6,7,22,23,20,22,16,23,19,22,23,21,22,17,26,8,5	(1,2) (2,3) (3,4) (4,6) (6,7) (7,22) (22,23) (23,20) (20,22) (22,16) (16,23) (23,19) (19,22) (22,23) (23,21) (21,22) (22,17) (17,26) (26,8) (8,5)
Test Case 28	1,3,4,6,7,21,20,22,17,14,7,22,17,23,22,16,17,27,9,5	(1,3) (3,4) (4,6) (6,7) (7,21) (21,20) (20,22) (22,17) (17,14) (14,7) (7,22) (22,17) (17,23) (23,22) (22,16) (16,17) (17,27) (27,9) (9,5)
Test Case 29	1,2,3,4,6,7,21,19,20,22,18,14,3,7,22,18,23,22,18,13,2,5	(1,2) (2,3) (3,4) (4,6) (6,7) (7,21) (21,19) (19,20) (20,22) (22,18) (18,14) (14,3) (3,7) (7,22) (22,18) (18,23) (23,22) (22,18) (18,13) (13,2) (2,5)
Test Case 30	1,2,3,4,6,7,22,17,18,15,7,22,15,2,5	(1,2) (2,3) (3,4) (4,6) (6,7) (7,22) (22,17) (17,18) (18,15) (15,7) (7,22) (22,15) (15,2) (2,5)
Test Case 31	1,4,2,6,7,20,19,22,16,15,4,5	(1,4) (4,2) (2,6) (6,7) (7,20) (20,19) (19,22) (22,16) (16,15) (15,4) (4,5)
Test Case 32	1,4,2,6,7,20,19,22,16,15,6,5	(1,4) (4,2) (2,6) (6,7) (7,20) (20,19) (19,22) (22,16) (16,15) (15,6) (6,5)
Test Case 33	1,4,6,7,20,22,14,2,7,20,19,22,14,6,5	(1,4) (4,6) (6,7) (7,20) (20,22) (22,14) (14,2) (2,7) (7,20) (20,19) (19,22) (22,14) (14,6) (6,5)
Test Case 34	1,2,4,6,7,19,20,22,18,13,4,5	(1,2) (2,4) (4,6) (6,7) (7,19) (19,20) (20,22) (22,18) (18,13) (13,4) (4,5)
Test Case 35	1,2,4,6,7,19,20,22,18,13,6,5	(1,2) (2,4) (4,6) (6,7) (7,19) (19,20) (20,22) (22,18) (18,13) (13,6) (6,5)
Test Case 36	1,2,4,6,7,20,19,12,4,5	(1,2) (2,4) (4,6) (6,7) (7,20) (20,19) (19,12) (12,4) (4,5)

Test Case 42	1,3,4,6,7,22,16,15,17,15,5	(1,3) (3,4) (4,6) (6,7) (7,22) (2,16) (16,15) (15,17) (17,15) (15,5)
Test Case 43	1,3,4,6,7,21,22,15,5	(1,3) (3,4) (4,6) (6,7) (7,21) (21,22) (22,15)
Test Case 44	1,3,4,6,7,21,20,25,9,5	(1,3) (3,4) (4,6) (6,7) (7,21) (21,20) (20,25) (25,9) (9,5)
Test Case 45	1,3,4,6,7,21,22,23,25,9,2,6,7,19,22,23,24,8,5	(1,3) (3,4) (4,6) (6,7) (7,21) (21,22) (22,23) (23,25) (25,9) (9,2) (2,6) (6,7) (7,19) (19,22) (22,23) (23,24) (24,8) (8,5)

Some transition pairs are not executable or impossible to execute because of the precondition that needs to be satisfied below is the list of such transition pair

Sno	Pairs	Reason/description
1	T8, T7	for it to be true value of s and t should be equal to 0 however when the vending machine is reaches idle state from T8 the values are not re-initialized and thus making the condition false
2	T7, T24	for T24 to be true the value of s should be 1 however at stage T7 the value of s =0 and before disposing cup of tea a cup needs to be selected thus it becomes not possible to execute this pair
3	T7, T8	Ghost transition at coin inserted state
4	T7, T12	for T12 to be true the value of s should be 1 however at stage T7 the value of s =0 and before disposing cup of tea a cup needs to be selected thus it becomes not possible to execute this pair
5	T7, T5	Ghost transition at coin inserted state
6	T7, T11	for T11 to be true the value of s should be 2 however at stage T7 the value of s =0 and before disposing cup of tea a cup needs to be selected thus it becomes not possible to execute this pair
7	T10, T7	if the set price is greater than 25 then it is not possible to execute the transition as the condition $(t+25 \geq \text{price})$ fails and (T10, T6) is performed instead of (T10, T7)
8	T3, T7	if price is set to value greater than 25 than this transition is not possible as the value of t will be initialized to 25 first.
9	T6, T6	Not executable
10	T18, T27	for transition T27 to be triggered value of s=2 however T18 will assign the value of 1 to s hence not possible
11	T27, T8	Ghost transition at no_small_cups state
12	T2, T7	if price is set to value greater than 25 than this transition is not possible as the value of t will be initialized to 25 first.
13	T1, T7	Not executable

14	T9, T7	for it to be true value of s and t should be equal to 0 however when the vending machine is reaches idle state from T9 the values are not re-initialized and thus making the condition false
15	T7, T25	for T25 to be true the value of s should be 2 however at stage T7 the value of s =0 and before disposing cup of tea a cup needs to be selected thus it becomes not possible to execute this pair
16	T1, T6	Not executable
17	T3, T6	if this transition is executed before setting the price for machine the method will return 0
18	T2, T7	if price is set to value greater than 25 than this transition is not possible as the value of t will be initialized to 25 first.
19	T4, T7	if price is set to value greater than 25 than this transition is not possible as the value of t will be initialized to 25 first.
20	T22, T13	if the value of s=0 or 2 and transition T13 is fired the result of the method will be 0
21	T19, T25	for T25 to be true the value of s should be 2 however at stage T19 the value of s =1 thus it becomes not possible to execute this pair
22	T21, T24	for T24 to be true the value of s should be 1 however at stage T21 the value of s =2 thus it becomes not possible to execute this pair
23	T6, T5	The method dispose returns 0 if no of large cups requested are more than 1 as in case 32
24	T20, T17	not possible
25	T17, T13	T17 sets value of s =2 and for T13 to be true value of s=1 which is not true hence not executable
26	T11, T7	since value of large cup is greater than 1 instead of transacting T11, T7 we have T11, T6 triggered since value of s is not equal to 0
27	T18, T15	T18 sets value of s =1 and for T15 to be true value of s=2 which is not true hence not executable
28	T12, T7	since value of large cup is greater than 1 instead of transacting T12, T7 we have T12, T6 triggered since value of s is not equal to 0

### Multiple condition testing:

Some branch predicates are not executable or impossible to execute because of the precondition that needs to be satisfied which can be seen in below result.

1.  $(t+25 \geq \text{price}) \ \&\&(\text{price} > 0)$

$(t+25 \geq \text{price})$	$(\text{price} > 0)$	Test Case Covered
T	T	Test Case #1
T	F	Not Possible
F	T	Test Case #4
F	F	Not possible

2.  $(t+25 < \text{price})$

$(t+25 < \text{price})$	Test Case Covered
T	Test Case #3
F	Test Case #6

3.  $(x > 1) \ \&\&(x < 6)$

$(x > 1)$	$(x < 6)$	Test Case Covered
T	T	Test Case #8
T	F	Not Executable
F	T	Test Case #37
F	F	Impossible



4.  $(x==2) \parallel (x==3)$  – Small cup

$(x==2)$	$(x ==3)$	Test Case Covered
T	T	Impossible
T	F	Test Case #15
F	T	Test Case #9
F	F	Test Case #8

5.  $(x==2) \parallel (x==3)$  – Large Cup

$(x==2)$	$(x ==3)$	Test Case Covered
T	T	Impossible
T	F	Test Case #3
F	T	Test Case #4
F	F	Test Case #7

6.  $((x==2) \parallel (x==3))$  - Sugar

$(x==2)$	$(x ==3)$	Test Case Covered
T	T	Impossible
T	F	Test Case #4
F	T	Test Case #6
F	F	Test Case #1

7.  $(x==2) \&\& (k1>1) \&\& (s==2)$  – Small Cup of tea

$(x==2)$	$(k1>1)$	$(s==2)$	Test Case Covered
T	T	T	Test Case #1
T	T	F	Test Case#8

T	F	T	Test Case#10
T	F	F	Test Case #5
F	T	T	Test Case #20
F	T	F	Test Case #27
F	F	T	Test Case #9
F	F	F	Test Case #6

8.  $(x==2) \ \&\&(k>1) \ \&\&(s==1)$  – Large Cup of tea

$(x==2)$	$(k>1)$	$(s==1)$	Test Case Covered
T	T	T	Test Case #2
T	T	F	Test Case #24
T	F	T	Test Case #2
T	F	F	Test Case#7
F	T	T	Test Case #18
F	T	F	Test Case #29
F	F	T	Test Case #4
F	F	F	Test Case #9

9.  $(x==2) \ \&\&(k==1) \ \&\&(s==1)$  – large cup of tea

$(x==2)$	$(k==1)$	$(s==1)$	Test Case Covered
T	T	T	Test Case #13
T	T	F	Test Case #14
T	F	T	Test Case #7
T	F	F	Test Case #22
F	T	T	Test Case #26

F	T	F	Test case #27
F	F	T	Test Case #17
F	F	F	Test Case #9

10. (x==2) &&(k1==1) &&(s==2) – small cup of tea

(x==2)	(k1==1)	(s==2)	Test Case Covered
T	T	T	Test Case #10
T	T	F	Test Case #2
T	F	T	Test Case #8
T	F	F	Test Case #25
F	T	T	Test Case #12
F	T	F	Test Case #13
F	F	T	Test Case #24
F	F	F	Test Case #17

11. (x==3) &&(k1==1) &&(s==2) – small cup with sugar

(x==3)	(k1==1)	(s==2)	Test Case Covered
T	T	T	Test Case #42
T	T	F	Test Case #4
T	F	T	Test Case #9
T	F	F	Test Case #26
F	T	T	Impossible
F	T	F	Test Case #2
F	F	T	Test Case #7
F	F	F	Test Case #2

12.  $(x==3) \ \&\&(k==1) \ \&\&(s==1)$  – large cup with sugar

$(x==3)$	$(k==1)$	$(s==1)$	Test Case Covered
T	T	T	Test Case#4
T	T	F	Test Case #20
T	F	T	Test Case #6
T	F	F	Test Case #9
F	T	T	Impossible
F	T	F	Test Case #16
F	F	T	Test Case #7
F	F	F	Test Case #1

13.  $(x==3) \ \&\&(k1>1) \ \&\&(s==2)$  – small cup with sugar

$(x==3)$	$(k1>1)$	$(s==2)$	Test Case Covered
T	T	T	Test Case #9
T	T	F	Test Case #11
T	F	T	Test Case #24
T	F	F	Test case #7
F	T	T	Impossible
F	T	F	Test Case #25
F	F	T	Test Case #10
F	F	F	Test Case#4

14. (x==3) &&(k>1) &&(s==1) – large cup with sugar

(x==3)	(k>1)	(s==1)	Test Case Covered
T	T	T	Test Case #23
T	T	F	Test Case #20
T	F	T	Test Case #11
T	F	F	Test Case #9
F	T	T	Impossible
F	T	F	Test Case #18
F	F	T	Test Case #2
F	F	F	Test Case #44

15. (x==1) &&(n>0) – Large cup

(x==1)	(n>0)	Test Case Covered
T	T	Test Case #5
T	F	Test case #26
F	T	Test Case#30
F	F	Test Case #21

16. (x==5) &&(n>0) – Large Cup

(x==5)	(n>0)	Test Case Covered
T	T	Test case #6
T	F	Test Case #4
F	T	Test Case #5
F	F	Test Case #2

17. (x==1) &&(n>0) – Small Cup

(x==1)	(n>0)	Test Case Covered
T	T	Test Case #8
T	F	Test Case #37
F	T	Test Case #10
F	F	Test Case #28

18. (x==4) &&(n>0)- Small Cup

(x==4)	(n>0)	Test Case Covered
T	T	Test Case#9
T	F	Test Case #28
F	T	Test Case #14
F	F	Test Case #14

19. (x==1) &&(p>0)

(x==1)	(p>0)	Test Case Covered
T	T	Test Case #1
T	F	Test Case #32
F	T	Test Case #22
F	F	Test Case #31

20. ((x==1)) – Dispose

(x == 1)	Test Case Covered
T	Test Case #7
F	Test Case #3

21. (x==1) – Coin

(x == 1)	Test Case Covered
T	Test Case #1
F	Test Case #8

22. (x==2) || (x==3) – cancel

(x==2)	(x ==3)	Test Case Covered
T	T	Impossible
T	F	Test Case #14
F	T	Test Case #23
F	F	Test Case #11

The source code and the test driver through which the above analysis was drawn out can be referenced through below documents:

- Ts.txt – Test Suit listing all the above-mentioned test cases
- VendingMachine.exe – Test driver to execute the test cases
- VendingMachine.cpp – Source code file containing the driver code as well
- Test Execution folder – Documents outcomes for each of the test cases
- Test suite checker – To check the syntax of test cases (already provided)

## Test Suite and Execution

Test Case No	Output	Status
Test Case 1	Dispose small cup of tea	Pass
Test Case 2	Dispose large cup of tea	Pass
Test Case 3	dispose method returned zero	
Test Case 4	Dispose large cup of tea with sugar	Pass
Test Case 5	DISPOSE LARGE CUP OF TEA	Pass
Test Case 6	Dispose large cup of tea with sugar	Pass
Test Case 7	DISPOSE LARGE CUP OF TEA	Pass
Test Case 8	DISPOSE SMALL CUP OF TEA	Pass
Test Case 9	DISPOSE SMALL CUP OF TEA with sugar	Pass
Test Case 10	dispose small cup of tea	Pass
Test Case 11	dispose large cup of tea with sugar	Pass
Test Case 12	dispose small cup of tea with sugar	Pass
Test Case 13	dispose large cup of tea	Pass
Test Case 14	dispose small cup of tea	Pass
Test Case 15	dispose	Pass
Test Case 16	dispose small cup of tea	Pass
Test Case 17	dispose large cup of tea with sugar	Pass
Test Case 18	Dispose small cup of tea	Pass
Test Case 19	disposes small cup of tea dispose large cup of tea	Pass
Test Case 20	dispose small cup of tea with sugar	Pass
Test Case 21	dispose large cup of tea	Pass
Test Case 22	dispose small cup of tea	Pass
Test Case 23	dispose large cup of tea with sugar	Pass
Test Case 24	dispose small cup of tea with sugar	Pass
Test Case 25	depose large cup of tea	Pass
Test Case 26	dispose large cup with sugar	Pass
Test Case 27	dispose small cup with sugar	Pass
Test Case 28	dispose small cup with sugar	Pass
Test Case 29	dispose large cup with sugar	Pass
Test Case 30	dispose large cup with sugar	Pass
Test Case 31	dispose large cup of tea with sugar	Pass
Test Case 32	dispose large cup of tea with sugar	Pass
Test Case 33	Passed	Pass
Test Case 34	dispose large cup of tea	Pass
Test Case 35	dispose large cup of tea with sugar	Pass
Test Case 36	dispose large cup of tea	Pass
Test Case 42	dispose small cup of tea with sugar	Pass
Test Case 43	dispose small cup of tea with sugar	Pass
Test Case 44	dispose small cup of tea with sugar	Pass
Test Case 45	depose small cu of tea	Pass



## Conclusion

I am very thankful to Prof. Korel Bogdan for giving us the opportunity to implement the teaching of the class in practical implementation. Through the different phases right from analysis of source code to execution of test suite helped in learning a lot of new challenges which one can face while in field of testing. Vending Machine – a simple tool which a person uses in his day to day life, never thought such complex testing methodology can be so easily used to make the machine error proof. This project helped in making use of traditional testing methods of testing to be used to test the software efficiently.

## Source Code

Rest of the code is unchanged only below code is added to the existing code of Vending Machine

```
void VendingMachine::showvariable()
{
    std::cout<<"x ="<< VendingMachine::x <<"\n" ;
    std::cout<<"k ="<< k <<"\n";
    std::cout<<"k1 ="<< k1<<"\n";
    std::cout<<"t ="<< t <<"\n";
    std::cout<<"price ="<< price <<"\n" ;
    std::cout<<"s ="<< s <<"\n";
}

int main()
{
    int i,n,p;
    char ch,op;
    VendingMachine vm;
    do
    {
        //system("cls");
        std::cout<<" \t  -----VENDING MACHINE MENU ----- " <<"\n";
        std::cout<<"-----" <<"\n";
        std::cout<<"|| \t 1. Coin inserted " <<"\n";
        std::cout<<"|| \t 2. Small_cup()" <<"\n";
        std::cout<<"|| \t 3. Large_cup()" <<"\n";
        std::cout<<"|| \t 4. Sugar() " <<"\n";
        std::cout<<"|| \t 5. Tea()" <<"\n";
        std::cout<<"|| \t 6. Inset_small_cup()" <<"\n";
        std::cout<<"|| \t 7. Insert_large_cup()" <<"\n";
        std::cout<<"|| \t 8. Set_Price" <<"\n";
        std::cout<<"|| \t 9. Dispose" <<"\n";
        std::cout<<"|| \t 0. Cancel()" <<"\n";
        std::cout<<"|| \t s. Show variables()" <<"\n";
        std::cout<<"-----" <<"\n";
        std::cout<<"Please enter your choice: ";
        std::cin>>ch;
```

```

switch(ch)
{ case '1': n= vm.coin();
    std::cout<<" The value returned by method is:"<<n <<"\n";
    break;

    case '2':
        n=vm.small_cup();
        std::cout<<" The value returned by method is:"<<n<<"\n";
        break;

    case '3': n=vm.large_cup();
        std::cout<<" The value returned by method is:"<<n<<"\n";
        break;

    case '4': n=vm.sugar();
        std::cout<<" The value returned by method is:"<<n<<"\n";
        break;

    case '5' : n=vm.tea();
        std::cout<<" The value returned by method is:"<<n<<"\n";
        break;

    case '6':
        std::cout<<" Enter the no of small cups:";
        std::cin>>i;
        n = vm.insert_small_cups(i);
        std::cout<<" The value returned by method is:"<<n<<"\n";
        break;

    case '7': std::cout<<" Enter the no of large cups: ";
        std::cin>>i;
        n=vm.insert_large_cups(i);
        std::cout<<" The value returned by method is:"<<n<<"\n";
        break;

    case '8': std::cout<<"Enter the price to be set: ";
        std::cin>>p;
        n=vm.set_price(p);
        std::cout<<" The value returned by method is:"<<n <<"\n";

```

```

        break;

    case 's': std::cout<<"\n";
        vm.showvariable();
        break;

    case '0': n= vm.cancel();
        std::cout<<" The value returned by method is:"<<n <<"\n";
        break;
    case '9':
        n = vm.dispose();
        std::cout<<" The value returned by method is:"<<n <<"\n";
        break;

    }
    std::cout<<"\n Do you want to continue (y/n):" <<"\t";
    std::cin>>op;
}while(op=='y');

return 0;
}

```