

# CS 589 Project

## 1. Model-Based Testing

- idle

The incoming transitions of state “idle” are: T1, T2, T3, T4, T6, T8, T9, T10, T11, T12, T13, T14, T15

The outgoing transitions of state “idle” are: T2, T3, T4, T5, T6, T7

Table 1 Transition Pairs of "idle"

idel					
transition pairs	test cases	transition pairs	test cases	transition pairs	test cases
T1,T2	4	T1,T3	6	T1,T4	1
T2,T2	4	T2,T3	4	T2,T4	6
T3,T2	6	T3,T3	4	T3,T4	4
T4,T2	1	T4,T3	7	T4,T4	6
T6,T2	7	T6,T3	9	T6,T4	9
T8,T2	11	T8,T3	14	T8,T4	18
T9,T2	14	T9,T3	12	T9,T4	17
T10,T2	8	T10,T3	12	T10,T4	7
T11,T2	17	T11,T3	18	T11,T4	10
T12,T2	15	T12,T3	10	T12,T4	14
T13,T2	17	T13,T3	18	T13,T4	6
T14,T2	16	T14,T3	19	T14,T4	19
T15,T2	16	T15,T3	18	T15,T4	19
transition pairs	test cases	transition pairs	test cases	transition pairs	test cases
T1,T5	5	T1,T6	not executable	T1,T7	8
T2,T5	11	T2,T6	7	T2,T7	1
T3,T5	12	T3,T6	7	T3,T7	7
T4,T5	6	T4,T6	4	T4,T7	6
T6,T5	10	T6,T6	9	T6,T7	4
T8,T5	8	T8,T6	16	T8,T7	11
T9,T5	9	T9,T6	13	T9,T7	12
T10,T5	7	T10,T6	17	T10,T7	11
T11,T5	3	T11,T6	19	T11,T7	12
T12,T5	1	T12,T6	16	T12,T7	18
T13,T5	2	T13,T6	19	T13,T7	15
T14,T5	15	T14,T6	19	T14,T7	18
T15,T5	13	T15,T6	14	T15,T7	18

the transition pair [T1,T6] is not executable because after initialize the vending machine(T1),  $k=0$ ,  $k1=0$ ,  $t=0$ ,  $price=0$ . Then we insert a coin in the vending machine(T6) so  $t+25 = 25 > price$  which is T7. Thus [T1,T6] is not executable.

- Coins inserted

The incoming transitions of state "coins inserted" are: T7, T19, T20, T21, T23

The outgoing transitions of state "coins inserted" are: T10, T11, T12, T19, T20, T21, T22, T24, T25

Table 2 Transition Pairs of "coins inserted"

coins inserted									
transition pairs	test cases	transition pairs	test cases	transition pairs	test cases	transition pairs	test cases	transition pairs	test cases
T7,T10	7	T19,T10	11	T20,T10	13	T21,T10	7	T23,T10	15
T7,T11	not executable	T19,T11,	not executable	T20,T11	12	T21,T11	3	T23,T11,	17
T7,T12	not executable	T19,T12	1	T20,T12	14	T21,T12	not executable	T23,T12	15
T7,T19	1	T19,T19	10	T20,T19	7	T21,T19	4	T23,T19	6
T7,T20	2	T19,T20	7	T20,T20	7	T21,T20	12	T23,T20	6
T7,T21	4	T19,T21	7	T20,T21	3	T21,T21	7	T23,T21	6
T7,T22	6	T19,T22	4	T20,T22	2	T21,T22	6	T23,T22	6
T7,T24	not executable	T19,T24	8	T20,T24	14	T21,T24	not executable	T23,T24	14
T7,T25	not executable	T19,T25	not executable	T20,T25	17	T21,T25	9	T23,T25	14

The transition pair [T7,T11], [T7,T12], [T7,T24], [T7,T25], [T19,T11], [T19,T25], [T21,T12], [T21,T24] are not executable. Because after T7,  $s=0$ ,  $t=0$ . However T11 and T25 happened when  $s==2$  and T12 and T24 happened when  $s==1$ . In addition, after T19,  $s=1$  but T11 and T25 happened when  $s==2$ . After T21,  $s=2$ , but T12 and T24 happened when  $s==1$ . Thus these transition pairs are not executable.

- Sugar

The incoming transitions of state "sugar" are: T16, T17, T18, T22

The outgoing transitions of state "sugar" are: T13, T14, T15, T16, T17, T18, T23, T26, T27

Table 3 Transition Pairs of "sugar"

sugar							
transition pairs	test cases	transition pairs	test cases	transition pairs	test cases	transition pairs	test cases
T16,T13	15	T17,T13	not executable	T18,T13	2	T22,T13	6
T16,T14	16	T17,T14	20	T18,T14	20	T22,T14	15
T16,T15	16	T17,T15	14	T18,T15	not executable	T22,T15	13
T16,T16	15	T17,T16	14	T18,T16	11	T22,T16	11
T16,T17	14	T17,T17	14	T18,T17	15	T22,T17	4
T16,T18	11	T17,T18	4	T18,T18	15	T22,T18	2
T16,T23	15	T17,T23	14	T18,T23	15	T22,T23	6
T16,T26	18	T17,T26	not executable	T18,T26	11	T22,T26	20
T16,T27	20	T17,T27	12	T18,T27	not executable	T22,T27	13

The transition pairs [T17,T13], [T17,T26], [T18,T25], [T18,T27] are not executable. Because after T17, s=2 but T13 and T26 happened when s==1 and after T18, s=1 but T15 and T27 happened when s==2. Thus these transition pairs are not executable.

- No large\_cups

The incoming transitions of state "no large\_cups" are: T24, T26, T29

The outgoing transitions of state "no large\_cups" are: T8, T29

Table 4 Transition Pairs of "no large\_cups"

no large_cups	
transition pairs	test cases
T24,T8	11
T24,T29	8
T26,T8	11
T26,T29	11
T29,T8	8
T29,T29	8

- No small\_cups

The incoming transitions of state "no small\_cups" are: T25, T27, T28

The outgoing transitions of state "no small\_cups" are: T9, T28

Table 5 Transition Pairs of "no small\_cups"

no small_cups	
transition pairs	test cases
T25,T9	12
T25,T28	9
T27,T9	13
T27,T28	12
T28,T9	9
T28,T28	9

## 2. Default(ghost) Transition Testing

The outgoing transitions of "idle" are: "insert\_large\_cups(n)[n>0]", "insert\_small\_cups(n)[n>0]", "set\_price(p)[p>0]", "coin[t+25<price]", "coin[(t+25>=price)&&(price>0)]", "dispose"

Table 6 Default/Ghost Transition of "idle"

idle	
default transition	test case
small_cup()	21
large_cup()	21
sugar()	21
tea()	21
insert_large_cups(n)[n<=0]	21
insert_small_cups(n)[n<=0]	21
set_price(p)[p<=0]	21
cancel()	21

The outgoing transitions of "coins inserted" are: "cancel", "tea[(k1>1)&&(s==2)]", "tea[(k>1)&&(s==1)]", "large\_cup", "coin", "small\_cup", "sugar", "tea[(k1=1)&&(s==2)]", "tea[(k==1)&&(s==1)]"

Table 7 Default/Ghost Transition of "coin inserted"

coins inserted	
default transition	test case
insert_large_cups(n)	22
insert_small_cups(n)	22
set_price(p)	22
tea()[(k<1)&&(k1<1)&&(s!=1)&&(s!=2)]	22
dispose()	22

The outgoing transitions of "sugar" are: "tea[(k>1)&&(s==1)]", "cancel", "tea[(k1>1)&&(s==2)]", "coin", "small\_cup", "large\_cup", "tea[(k1=1)&&(s==2)]", "tea[(k==1)&&(s==1)]", "sugar"

Table 8 Default/Ghost Transition of "sugar"

sugar	
default transition	test case
insert_large_cups(n)	23
insert_small_cups(n)	23
set_price(p)	23
tea()[(k<1)&&(k1<1)&&(s!=1)&&(s!=2)]	23
dispose()	23

The outcoming transitions of "no large\_cups" are: "coin", "insert\_large\_cups(n)[n>0]"

Table 9 Default/Ghost Transition of "no large\_cups"

no large_cups	
default transition	test case
small_cup()	24
large_cup()	24
sugar()	24
tea()	24
insert_large_cups(n)[n<=0]	24
insert_small_cups(n)	24
set_price(p)	24
cancel()	24
dispose()	24

The outcoming transitions of "no small\_cups" are: "coin", "insert\_small\_cups(n)[n>0]"

Table 10 Default/Ghost Transition of "no small\_cups"

no small_cups	
default transition	test case
small_cup()	25
large_cup()	25
sugar()	25
tea()	25
insert_large_cups(n)	25
insert_small_cups(n)[n<=0]	25
set_price(p)	25
cancel()	25
dispose()	25

### 3. Multiple-Condition Testing

The multiple-condition of method "coin()" are:

```
if (x == 1)
if ((t + 25 >= price) && (price > 0))
if (t + 25 < price)
if ((x > 1) && (x < 6))
```

Table 11 Multiple-condition of "coin()"

coin									
x == 1	test case	t + 25 >= price	price > 0	test case	t + 25 < price	test case	x > 1	x < 6	test case
T	26	T	T	26	T	34	T	T	26
F	26	T	F	7	F	26	T	F	not executable
		F	T	34			F	T	34
		F	F	not executable			F	F	not executable

(t+25>=price)F and (price>0)F is not executable because t+25>=price is false means t+25<price and price>0 means price<=0. So it means t+25<0 but t>=0. Thus (t+25>=price) F and (price>0)F is not executable.

(x>1)T and (x<6)F means x>=6. However when the vending machine execute coin(), x is 1<=x<=5. Thus (x>1)T and (x<6)F is not executable.

(x>1)F and (x<6)F is not executable because it means (x<=1)&&(x>=6) which is impossible. Thus (x>1)F and (x<6)F is not executable.

The multiple-condition of method "small\_cup()" are:

```
if ((x == 2) || (x == 3))
```

Table 12 Multiple-condition of "small\_cup()"

small_cup()		
x == 2	x == 3	test case
T	T	not executable
T	F	27
F	T	27
F	F	21

(x==2)T and (x==3)T is not executable because x cannot equals to 2 and 3 at the same time.

The multiple-condition of method "large\_cup()" are:

```
If ((x == 2) || (x == 3))
```

Table 13 Multiple\_condition of "large\_cup()"

large_cup()		
x == 2	x == 3	test case
T	T	not executable
T	F	26
F	T	26
F	F	21

(x==2)T and (x==3)T is not executable because x cannot equals to 2 and 3 at the same time.

The multiple-condition of method "sugar()" are:

if ((x == 2) || (x == 3))  
if (x == 2)

Table 14 Multiple-condition of "sugar()"

sugar()				
x == 2	x == 3	test case	x == 2	test case
T	T	not executable	T	26
T	F	26	F	34
F	T	34		
F	F	21		

(x==2)T and (x==3)T is not executable because x cannot equals to 2 and 3 at the same time.

The multiple-condition of method "tea()" are:

if ((x == 2) || (x == 3))

x == 2	x == 3	test case
T	T	not executable
T	F	26
F	T	26
F	F	21

(x==2)T and (x==3)T is not executable because x cannot equals to 2 and 3 at the same time.

if ((x == 2) && (k1 > 1) && (s == 2))



x == 2	k1 > 1	s == 2	test case
T	T	T	29
T	T	F	30
T	F	T	27
T	F	F	26
F	T	T	27
F	T	F	30
F	F	T	29
F	F	F	26

if ((x == 2) && (k > 1) && (s == 1))

x == 2	k > 1	s == 1	test case
T	T	T	28
T	T	F	31
T	F	T	26
T	F	F	27
F	T	T	26
F	T	F	31
F	F	T	28
F	F	F	27

if ((x == 2) && (k == 1) && (s == 1))

x == 2	k == 1	s == 1	test case
T	T	T	26
T	T	F	33
T	F	T	28
T	F	F	27
F	T	T	28
F	T	F	33
F	F	T	26
F	F	F	27

if ((x == 2) && (k1 == 1) && (s == 2))

x == 2	k1 == 1	s == 2	test case
T	T	T	27
T	T	F	32
T	F	T	29
T	F	F	26
F	T	T	29
F	T	F	32
F	F	T	27
F	F	F	26

if ((x == 3) && (k1 == 1) && (s == 2))

x == 3	k1 == 1	s == 2	test case
T	T	T	29
T	T	F	32
T	F	T	27
T	F	F	26
F	T	T	27
F	T	F	32
F	F	T	29
F	F	F	26

if ((x == 3) && (k == 1) && (s == 1))

x == 3	k == 1	s == 1	test case
T	T	T	28
T	T	F	33
T	F	T	26
T	F	F	27
F	T	T	26
F	T	F	33
F	F	T	28
F	F	F	27

if ((x == 3) && (k1 > 1) && (s == 2))

x == 3	k1 > 1	s == 2	test case
T	T	T	27
T	T	F	30
T	F	T	29
T	F	F	26
F	T	T	29
F	T	F	30
F	F	T	27
F	F	F	26

if ((x == 3) && (k > 1) && (s == 1))

x == 3	k > 1	s == 1	test case
T	T	T	26
T	T	F	31
T	F	T	28
T	F	F	27
F	T	T	28
F	T	F	31
F	F	T	26
F	F	F	27

The multiple-condition of method “insert\_large\_cups(int n)” are:

if ((x == 1) && (n > 0))

if ((x == 5) && (n > 0))

Table 15 Multiple-condition of "insert\_large\_cups()"

insert_large_cups()					
x == 1	n > 0	test case	x == 5	n > 0	test case
T	T	26	T	T	26
T	F	21	T	F	24
F	T	26	F	T	26
F	F	24	F	F	21

The multiple-condition of method “insert\_small\_cups(int n)” are:

if ((x == 1) && (n > 0))

if ((x == 4) && (n > 0))

Table 16 Multiple-condition of "insert\_small\_cups()"

insert_small_cups()					
x == 1	n > 0	test case	x == 4	n > 0	test case
T	T	27	T	T	27
T	F	21	T	F	25
F	T	27	F	T	27
F	F	25	F	F	21

The multiple-condition of method “set\_price(int p)” are:

if ((x == 1) && (p > 0))

Table 17 Multiple-condition of "set\_price()"

set_price()		
x == 1	p > 0	test case
T	T	26
T	F	21
F	T	22
F	F	23

The multiple-condition of method “cancel()” are:

if ((x == 2) || (x == 3))

Table 18 Multiple-condition of "cancel()"

cancel()		
x == 2	x == 3	test case
T	T	not executable
T	F	34
F	T	34
F	F	21

(x==2)T and (x==3)T is not executable because x cannot equals to 2 and 3 at the same time.

The multiple-condition of method "dispose()" are:  
if (x == 1)

Table 19 Multiple-condition of "dispose()"

dispose()	
x == 1	test case
T	26
F	22

4. Test Suite and the result of its execution

- Test cases for the transition pairs

Test#1: set\_price 25 insert\_large\_cups 5 coin large\_cup tea dispose

Transitions: T1,T4,T2,T7,T19,T12,T5

States: idle, coins inserted, idle, end

Correctness: correct

Test#2: set\_price 25 insert\_large\_cups 5 coin coin sugar large\_cup tea dispose

Transitions: T1,T4,T2,T7,T20,T22,T18,T13,T5

States: idle, coins inserted, sugar, idle, end

Correctness: correct

Test#3: set\_price 25 insert\_small\_cups 5 coin coin small\_cup tea dispose

Transitions: T1,T4,T3,T7,T20,T21,T11,T5

States: idle, coins inserted, idle, end

Correctness: correct

Test#4: insert\_large\_cups 5 insert\_large\_cups 5 insert\_small\_cups 5 insert\_small\_cups 5  
set\_price 50 coin coin small\_cup large\_cup sugar small\_cup large\_cup tea dispose

Transitions: T1,T2,T2,T3,T3,T4,T6,T7,T21,T19,T22,T17,T18,T13,T5

States: start, idle, coins inserted, sugar, idle, end

Correctness: correct

Test#5: dispose

Transitions: T1,T5

States: idle, end

Correctness: correct

Test#6: insert\_small\_cups 5 insert\_large\_cups 5 set\_price 25 set\_price 25 coin sugar sugar  
coin sugar sugar small\_cup sugar sugar large\_cup sugar sugar sugar tea set\_price 25 dispose

Transitions:

T1,T3,T2,T4,T4,T7,T22,T23,T20,T22,T23,T21,T22,T23,T19,T22,T23,T22,T13,T4,T5

States: idle, coins inserted, sugar, coins inserted, sugar, coins inserted, sugar, coins inserted, sugar, coins inserted, sugar, idle, end

Correctness: correct

Test#7: insert\_small\_cups 5 coin cancel set\_price 100 insert\_small\_cups 5 coin  
insert\_large\_cups 5 coin coin large\_cup coin coin large\_cup small\_cup small\_cup cancel  
dispose

Transitions: T1,T3,T7,T10,T4,T3,T6,T2,T6,T7,T19,T20,T20,T19,T21,T21,T10,T5

States: idle, coins inserted, idle, coins inserted, idle, end

Correctness: failed

Actual states: idle, coins inserted, idle, end

Test#8: coin cancel insert\_large\_cups 1 set\_price 25 coin large\_cup tea coin coin insert\_large\_cups 1 dispose

Transitions: T1,T7,T10,T2,T4,T7,T19,T24,T29,T29,T8,T5

States: idle, coins inserted, idle, coins inserted, no large\_cups, idle, end

Correctness: failed

Actual states: idle, coins inserted, no large\_cups, idle, end

Test#9: set\_price 100 coin coin insert\_small\_cups 1 coin set\_price 100 coin small\_cup tea coin coin insert\_small\_cups 1 dispose

Transitions: T1,T4,T6,T6,T3,T6,T4,T7,T21,T25,T28,T28,T9,T5

States: idle, coins inserted, no small\_cups, idle, end

Correctness: correct

Test#10: set\_price 25 insert\_large\_cups 2 coin large\_cup large\_cup tea insert\_small\_cups 2 coin small\_cup tea set\_price 50 coin dispose

Transitions: T1,T4,T2,T7,T19,T19,T12,T3,T7,T21,T11,T4,T6,T5

States: idle, coins inserted, idle, coins inserted, idle, end

Correctness: correct

Test#11: set\_price 25 insert\_large\_cups 1 coin large\_cup cancel coin large\_cup tea insert\_large\_cups 1 coin sugar coin large\_cup coin large\_cup tea coin insert\_large\_cups 1 coin sugar large\_cup tea insert\_large\_cups 1 insert\_large\_cups 1 dispose

Transitions:

T1,T4,T2,T7,T19,T10,T7,T19,T24,T8,T7,T22,T16,T18,T16,T18,T26,T29,T8,T7,T22,T18,T26,T8,T2,T5

States: idle, coins inserted, idle, coins inserted, no large\_cups, idle, coins inserted, sugar, no large\_cups, idle, coins inserted, sugar, no large\_cups, idle, end

Correctness: correct

Test#12: set\_price 25 insert\_small\_cups 1 coin cancel insert\_small\_cups 1 coin small\_cup coin tea coin small\_cup tea insert\_small\_cups 1 coin sugar small\_cup tea coin insert\_small\_cups 1 insert\_small\_cups 1 dispose

Transitions:

T1,T4,T3,T7,T10,T3,T7,T21,T20,T11,T7,T21,T25,T9,T7,T22,T17,T27,T28,T9,T3,T5

States: idle, coins inserted, idle, coins inserted, idle, coins inserted, no small\_cups, idle, coins inserted, sugar, no small\_cups, idle, end

Correctness: correct

Test#13: set\_price 50 insert\_small\_cups 1 coin coin coin cancel coin small\_cup sugar tea insert\_small\_cups 2 coin coin small\_cup sugar tea dispose

Transitions: T1,T4,T3,T6,T7,T20,T10,T7,T21,T22,T27,T9,T6,T7,T21,T22,T15,T5

States: idle, coins inserted, idle, coins inserted, sugar, no small\_cups, idle, coins inserted, sugar, idle, end

Correctness: correct

Test#14: set\_price 50 insert\_small\_cups 2 coin coin sugar coin small\_cup coin small\_cup small\_cup tea coin coin sugar small\_cup sugar tea insert\_small\_cups 1 insert\_large\_cups 1 coin coin large\_cup coin tea set\_price 25 coin large\_cup coin tea insert\_large\_cups 1 coin large\_cup sugar sugar tea insert\_large\_cups 1 insert\_small\_cups 1 dispose

Transitions:

T1,T4,T3,T6,T7,T22,T16,T17,T16,T17,T17,T15,T6,T7,,T22,T17,T23,T25,T9,T2,T6,T7,T19,T20,T12,T4,T7,T19,T20,T24,T8,T7,T19,T22,T23,T12,T2,T3,T5

States: idle, coins inserted, sugar, idle, coin inserted, sugar, coins inserted, no small\_cups, idle, coins inserted, idle, coins inserted, no large\_cups, idle, coins inserted, sugar, coins inserted, no large\_cups, idle, end

Correctness: correct

Test#15: set\_price 25 coin sugar sugar cancel insert\_large\_cups 2 coin sugar large\_cup small\_cup large\_cup coin coin large\_cup large\_cup sugar tea insert\_large\_cups 1 coin sugar large\_cup coin tea coin large\_cup sugar coin sugar sugar cancel dispose

Transitions:

T1,T4,T7,T22,T23,T10,T2,T7,T22,T18,T17,T18,T16,T16,T18,T18,T23,T12,T2,T7,T22,T18,T16,T13,T7,T19,T22,T16,T23,T22,T14,T5

States: idle, coins inserted, sugar, coins inserted, idle, coins inserted, sugar, coins inserted, idle, coins inserted, sugar, idle, coins inserted, sugar, coins inserted, sugar, idle, end

Correctness: correct

Test#16: set\_price 50 insert\_small\_cups 1 coin coin sugar coin cancel insert\_large\_cups 1 coin coin sugar small\_cup coin tea insert\_large\_cups 1 coin coin large\_cup tea insert\_large\_cups 2 coin coin large\_cup tea coin dispose

Transitions:

T1,T4,T6,T3,T7,T22,T16,T14,T2,T6,T7,T22,T17,T16,T15,T2,T6,T7,T19,T24,T8,T6,T7,T19,T12,T6,T5

States: idle, coins inserted, sugar, idle, coins inserted, sugar, no large\_cups, idle, coins inserted, no large\_cups, idle, coins inserted, idle, end

Correctness: correct

Test#17: set\_price 25 insert\_small\_cups 2 coin small\_cup sugar sugar tea insert\_large\_cups 2 coin sugar large\_cup tea insert\_large\_cups 1 coin small\_cup coin tea insert\_small\_cups 1 set\_price 50 coin coin cancel coin dispose

Transitions:

T1,T4,T3,T7,T21,T22,T23,T11,T2,T7,T22,T18,T13,T2,T7,T21,T20,T25,9,T4,T6,T7,T10,T6,T5

States: idle, coins inserted, sugar, coins inserted, idle, coins inserted, sugar, idle, coins inserted, no small\_cups, idle, coins inserted, idle, end

Correctness: correct

Test#18: set\_price 25 insert\_large\_cups 2 insert\_small\_cups 2 coin small\_cup tea  
 insert\_small\_cups 1 coin small\_cup sugar tea insert\_small\_cups 1 coin sugar large\_cup tea  
 insert\_small\_cups 1 coin large\_cup tea insert\_large\_cups 2 set\_price 25 coin large\_cup tea  
 coin sugar cancel coin sugar small\_cup tea coin sugar large\_cup coin tea insert\_large\_cups 1  
 dispose

Transitions:

T1,T4,T2,T3,T7,T21,T11,T3,T7,T21,T22,T15,T3,T7,T22,T18,T13,T3,T7,T19,T24,T8,T4,T7,T19,T  
 12,T7,T22,T14,T7,T22,T17,T15,T7,T22,T18,T16,T26,T8,T5

States: idle, coins inserted, idle, coins inserted, sugar, idle, coins inserted, sugar, idle,  
 coins inserted, no large\_cups, idle, coins inserted, idle, coins inserted, sugar, idle, coins  
 inserted, sugar, idle, coins inserted, sugar, no large\_cups, idle, end

Correctness: correct

Test#19: set\_price 25 insert\_large\_cups 2 coin sugar cancel insert\_small\_cups 5 coin sugar  
 cancel set\_price 25 coin sugar small\_cup tea set\_price 50 coin coin small\_cup tea coin coin  
 sugar cancel coin coin sugar large\_cup tea coin dispose

Transitions:

T1,T4,T2,T7,T22,T14,T3,T7,T22,T14,T4,T7,T22,T17,T15,T4,T6,T7,T21,T11,T6,T7,T22,T14,T6,T  
 7,T22,T18,T13,T6,T5

States: idle, coins inserted, sugar, idle, coins inserted, sugar, idle, coins inserted, sugar,  
 idle, coins inserted, idle, coins inserted, sugar, idle, coins inserted, sugar, idle, end

Correctness: correct

Test#20: set\_price 25 insert\_large\_cups 1 insert\_small\_cups 1 coin sugar small\_cup cancel  
 coin sugar large\_cup cancel coin large\_cup sugar tea insert\_large\_cups 1 coin small\_cup  
 sugar coin tea insert\_small\_cups 1 dispose

Transitions:

T1,T4,T2,T3,T7,T22,T17,T14,T7,T22,T18,T14,T14,T7,T19,T22,T26,T8,T7,T21,T22,T16,T27,T9,T  
 5

States: idle, coins inserted, sugar, idle, coins inserted, sugar, idle, coins inserted, sugar,  
 no large\_cups, idle, coins inserted, sugar, no small\_cups, idle, end

Correctness: correct

- Test cases for the Default/Ghost transitions

Test#21: set\_price 0 insert\_small\_cups 0 insert\_large\_cups 0 small\_cup large\_cup sugar tea  
 cancel

States: idle

Correctness: correct

Values:

	x	price	k	K1	t	s
Vendingmachine	1	0	0	0	0	0
Set_price	1	0	0	0	0	0
Insert_small_cups	1	0	0	0	0	0



Insert_large_cups	1	0	0	0	0	0
Small_cup	1	0	0	0	0	0
Large_cup	1	0	0	0	0	0
Sugar	1	0	0	0	0	0
Tea	1	0	0	0	0	0
cancel	1	0	0	0	0	0

Test#22: set\_price 25 coin tea insert\_large\_cups 5 insert\_small\_cups 5 set\_price 25 dispose

States: idle, coins inserted

Correctness: correct

Values:

	x	price	k	K1	t	s
Vendingmachine	1	0	0	0	0	0
Set_price	1	25	0	0	0	0
Coin	2	25	0	0	0	0
Tea	2	25	0	0	0	0
Insert_large_cups	2	25	0	0	0	0
Insert_small_cups	2	25	0	0	0	0
Set_price	2	25	0	0	0	0
dispose	2	25	0	0	0	0

Test#23: set\_price 25 coin sugar tea insert\_large\_cups 5 insert\_small\_cups 5 set\_price 0 dispose

States: idle, coins inserted, sugar

Correctness: correct

Value:

	x	price	k	K1	t	s
Vendingmachine	1	0	0	0	0	0
Set_price	1	25	0	0	0	0
Coin	2	25	0	0	0	0
sugar	3	25	0	0	0	0
tea	3	25	0	0	0	0
Insert_large_cups	3	25	0	0	0	0
Insert_small_cups	3	25	0	0	0	0
Set_price	3	25	0	0	0	0
dispose	3	25	0	0	0	0

Test#24: set\_price 25 insert\_large\_cups 1 coin large\_cup tea small\_cup large\_cup sugar tea insert\_large\_cups 0 insert\_small\_cups 5 set\_price 25 cancel dispose

States: idle, coins inserted, no large\_cups

Correctness: correct

Value:

	x	price	k	K1	t	s
--	---	-------	---	----	---	---

Vendingmachine	1	0	0	0	0	0
Set_price	1	25	0	0	0	0
Insert_large_cups	1	25	1	0	0	0
Coin	2	25	1	0	0	0
Large_cup	2	25	1	0	0	1
Tea	5	25	0	0	0	1
Small_cup	5	25	0	0	0	1
Large_cup	5	25	0	0	0	1
Sugar	5	25	0	0	0	1
Tea	5	25	0	0	0	1
Insert_large_cups	5	25	0	0	0	1
Insert_small_cups	5	25	0	0	0	1
Set_price	5	25	0	0	0	1
Cancel	5	25	0	0	0	1
dispose	5	25	0	0	0	1

Test#25: set\_price 25 insert\_small\_cups 1 coin small\_cup tea small\_cup large\_cup sugar tea  
insert\_small\_cups 0 insert\_large\_cups 5 set\_price 25 cancel dispose

States: idle, coins inserted, no small\_cups

Correctness: correct

Value:

	x	price	k	K1	t	s
Vendingmachine	1	0	0	0	0	0
Set_price	1	25	0	0	0	0
Insert_small_cups	1	25	0	1	0	0
Coin	2	25	0	1	0	0
small_cup	2	25	0	1	0	2
Tea	4	25	0	0	0	2
Small_cup	4	25	0	0	0	2
Large_cup	4	25	0	0	0	2
Sugar	4	25	0	0	0	2
Tea	4	25	0	0	0	2
Insert_samll_cups	4	25	0	0	0	2
Insert_large_cups	4	25	0	0	0	2
Set_price	4	25	0	0	0	2
Cancel	4	25	0	0	0	2
dispose	4	25	0	0	0	2

- Test cases for the multiple-condition testing

Test#26: set\_price 25 insert\_large\_cups 1 coin large\_cup coin tea insert\_large\_cups 2 coin  
sugar large\_cup tea dispose

States: idle, coins inserted, no large\_cups, idle, coins inserted, sugar, idle, end

Correctness: correct

Value:

X	1	1	1	2	2	2	5	1	2	3	3	1	6
price	0	25	25	25	25	25	25	25	25	25	25	25	25
K	0	0	1	1	1	1	0	2	2	2	2	1	1
K1	0	0	0	0	0	0	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	1	1	1	1	0	0	1	1	1

Test#27: set\_price 25 insert\_small\_cups 1 coin small\_cup coin tea insert\_small\_cups 2 coin sugar small\_cup tea dispose

States: idle, coins inserted, no small\_cups, idle, coins inserted, sugar, idle, end

Correctness: correct

Value:

X	1	1	1	2	2	2	4	1	2	3	3	1	6
Price	0	25	25	25	25	25	25	25	25	25	25	25	25
K	0	0	0	0	0	0	0	0	0	0	0	0	0
K1	0	0	1	1	1	1	0	2	2	2	2	1	1
T	0	0	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	2	2	2	2	0	0	2	2	2

Test#28: set\_price 25 insert\_large\_cups 2 coin large\_cup tea coin sugar large\_cup tea insert\_large\_cups 1 dispose

States: idle, coins inserted, idle, coins inserted, sugar, no large\_cups, idle, end

Correctness: correct

Value:

X	1	1	1	2	2	1	2	3	3	5	1	6
Price	0	25	25	25	25	25	25	25	25	25	25	25
K	0	0	2	2	2	1	1	1	1	0	1	1
K1	0	0	0	0	0	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	1	1	0	0	1	1	1	1

Test#29: set\_price 25 insert\_small\_cups 2 coin small\_cup tea coin sugar small\_cup tea insert\_small\_cups 1 dispose

States: idle, coins inserted, idle, coins inserted, sugar, no small\_cups, idle, end

Correctness: correct

Value:

X	1	1	1	2	2	1	2	3	3	5	1	6
Price	0	25	25	25	25	25	25	25	25	25	25	25
K	0	0	0	0	0	0	0	0	0	0	0	0
K1	0	0	2	2	2	1	1	1	1	0	1	1
T	0	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	2	2	0	0	2	2	2	2

Test#30: set\_price 25 insert\_large\_cups 3 insert\_small\_cups 3 coin large\_cup tea coin sugar large\_cup tea dispose

States: idle, coins inserted, idle, coins inserted, sugar, idle, end

Correctness: correct

Value:

X	1	1	1	1	2	2	1	2	3	3	1	6
Price	0	25	25	25	25	25	25	25	25	25	25	25
K	0	0	3	3	3	3	2	2	2	2	1	1
K1	0	0	0	3	3	3	3	3	3	3	3	3
T	0	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	1	1	0	0	1	1	1

Test#31: set\_price 25 insert\_large\_cups 3 insert\_small\_cups 3 coin small\_cup tea coin sugar small\_cup tea dispose

States: idle, coins inserted, idle, coins inserted, sugar, idle, end

Correctness: correct

Value:

X	1	1	1	1	2	2	1	2	3	3	1	6
Price	0	25	25	25	25	25	25	25	25	25	25	25
K	0	0	3	3	3	3	3	3	3	3	3	3
K1	0	0	0	3	3	3	2	2	2	2	1	1
T	0	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	2	2	0	0	2	2	2

Test#32: set\_price 25 insert\_large\_cups 1 insert\_small\_cups 1 coin large\_cup tea insert\_large\_cups 1 coin sugar large\_cup tea insert\_large\_cups 1 dispose

States: idle, coins inserted, no large\_cups, idle, coins inserted, sugar, no large\_cups, idle, end

Correctness: correct

Value:

X	1	1	1	1	2	2	5	1	2	3	3	5	1	6
Price	0	25	25	25	25	25	25	25	25	25	25	25	25	25
K	0	0	1	1	1	1	0	1	1	1	1	0	1	1
K1	0	0	0	1	1	1	1	1	1	1	1	1	1	1
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	1	1	1	0	0	1	1	1	1

Test#33: set\_price 25 insert\_large\_cups 1 insert\_small\_cups 1 coin small\_cup tea insert\_small\_cups 1 coin sugar small\_cup tea insert\_small\_cups 1 dispose

States: idle, coins inserted, no small\_cups, idle, coins inserted, sugar, no small\_cups, idle, end

Value:

X	1	1	1	1	2	2	4	1	2	3	3	4	1	6
Price	0	25	25	25	25	25	25	25	25	25	25	25	25	25
K	0	0	1	1	1	1	1	1	1	1	1	1	1	1
K1	0	0	0	1	1	1	0	1	1	1	1	0	1	1
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	2	2	2	0	0	2	2	2	2

Test#34: set\_price 50 insert\_large\_cups 1 coin coin sugar sugar cancel coin coin sugar cancel dispose

States: idle, coins inserted, sugar, coins inserted, idle, coins inserted, sugar, idle, end

Correctness: correct

Value:

[illegible]

## 5. Conclusion

In this project, I design a test drive to test a vending machine with specific source code and EFSM. This vending machine is implemented in Java environment.

In addition, I design two testing oriented method, `show_value()` and `show_state()`. When using vending machine and pressing 'a', the internal values of all the variables would be shown on the screen. And using vending machine and pressing 'b', the current state of the machine would be shown on the screen. By using these two testing oriented method we can design test cases to have class testing and model-based testing.

When having the class based testing, I concentrated on the internal branches and values. So use the `show_value()` method to compare the test result with the expected result. When having the model-based testing, I concentrated on the internal state transitions. So I use the `show_state()` method to compare the test result with the expected result.