CS 589 Project

Jian Zhang

A20327380

jzhan150@hawk.iit.edu

1: Transition-Pairing Testing

Firstly, identify all the transition pairs for each state in the EFSM.

1. For the state S0, the incoming transitions are T1, T3, T5, T13, T14, T19,the outgoing transitions are T2, T6, T20:

Table1.1: Transition Pairs of S0

Transition Pair	Test Case
(T1, T2)	Test#1
(T1, T6)	Test#2
(T1, T20)	Test#6
(T3, T2)	Test#7
(T3, T6)	Test#1
(T3, T20)	Test#3
(T5, T2)	Test#3
(T5, T6)	Test#2
(T5, T20)	Test#7
(T13, T2)	Test#3
(T13, T6)	Test#8
(T13, T20)	Test#5
(T14, T2)	Test#9
(T14, T6)	Test#2
(T14, T20)	Test#1
(T19, T2)	Test#3
(T19, T6)	Test#3
(T19, T20)	Test#2

2. For the state S1, the incoming transitions are T17, T18, the outgoing transitions are T16, T17:

Table1.2: Transition Pairs of S1

Transition Pair	Test Case
(T17, T16)	Test#11
(T17, T17)	Test#11
(T18, T16)	Test#10
(T18, T17)	Test#11

3. For the state S2, the incoming transitions are T2, the outgoing transitions are T3, T4:

Table1.3: Transition Pairs of S2

Transition Pair	Test Case
(T2, T3)	Test#1
(T2, T4)	Test#3

4. For the state S3, the incoming transitions are T4, T6, the outgoing transitions areT5, T7, T8,T15:

Table1.4: Transition Pairs of S3

Transition Pair	Test Case
(T4, T5)	Test#7
(T4, T7)	Test#8
(T4, T8)	Test#3
(T4, T15)	Test#3
(T6, T5)	Test#2
(T6, T7)	Test#2
(T6, T8)	Test#2
(T6, T15)	Test#1

5. For the state S4, the incoming transitions are T7, T8,T15, the outgoing transitions are T9, T18, T19:

Table1.5: Transition Pairs of S4

Transition Pair	Test Case
(T7, T9)	Test#3
(T7, T18)	Test#10
(T7, T19)	Test#2
(T8, T9)	Test#2
(T8, T18)	Test#11
(T8, T19)	Test#3
(T15, T9)	Test#1
(T15, T18)	Test#12
(T15, T19)	Test#3

6. For the state S5, the incoming transitions are T9, T10, the outgoing transitions are T10, T11, T12:

Table1.6: Transition Pairs of S5

Transition Pair	Test Case
(T9, T10)	Test#1
(T9, T11)	Test#2
(T9, T12)	Test#3
(T10, T10)	Test#1
(T10, T11)	Test#1
(T10, T12)	Test#4

7. For the state S6, the incoming transitions are T11, T12, T16, the outgoing transitions are T13, T14:

Table1.7: Transition Pairs of S6

Transition Pair	Test Case
(T11, T13)	Test#5
(T11, T14)	Test#1
(T12, T13)	Test#3
(T12, T14)	Test#4
(T16, T13)	Test#10
(T16, T14)	Test#11

2: Default/Ghost Transition Testing

Firstly, identify all the default/ghost transitions for each state in the EFSM.

1. For the starting state, the outgoing transition is T1:

Table2.1: Default/Ghost Transitions of Starting State

Default/Ghost Transition	Test Case
Active(a, b, c) [a<=0||b<=0||d<=0]	Test#13
PayCredit()	Test#13
Reject()	Test#13
Approved()	Test#13
Cancel()	Test#13
PayCash(c)	Test#13
Regular()	Test#13
Super()	Test#13
Diesel()	Test#13
StartPump()	Test#13
PumpLiter()	Test#13
StopPump()	Test#13
NoReceipt()	Test#13
Receipt()	Test#13
TurnOff()	Test#13

2. For the state S0, the outgoing transition is T2, T6, T20:

Table2.2: Default/Ghost Transitions of S0

Default/Ghost Transition	Test Case
Active(a, b, c)	Test#14
Reject()	Test#14
Approved()	Test#14
Cancel()	Test#14
PayCash(c)[c<=0]	Test#14
Regular()	Test#14
Super()	Test#14
Diesel()	Test#14
StartPump()	Test#14
PumpLiter()	Test#14
StopPump()	Test#14

NoReceipt()	Test#14
Receipt()	Test#14

3. For the state S1, the outgoing transition is T16, T17:

Table2.3: Default/Ghost Transitions of S1

Default/Ghost Transition	Test Case
Active(a, b, c)	Test#15
PayCredit()	Test#15
Reject()	Test#15
Approved()	Test#15
Cancel()	Test#15
PayCash(c)	Test#15
Regular()	Test#15
Super()	Test#15
Diesel()	Test#15
StartPump()	Test#15
NoReceipt()	Test#15
Receipt()	Test#15
TurnOff()	Test#15

4. For the state S2, the outgoing transition is T3, T4:

Table2.4: Default/Ghost Transitions of S2

Default/Ghost Transition	Test Case
Active(a, b, c)	Test#16
PayCredit()	Test#16
PayCash(c)	Test#16
Cancel()	Test#16
Regular()	Test#16
Super()	Test#16
Diesel()	Test#16
StartPump()	Test#16
PumpLiter()	Test#16
StopPump()	Test#16
NoReceipt()	Test#16
Receipt()	Test#16
TurnOff()	Test#16

5. For the state S3, the outgoing transition is T5, T7, T8, T15:

Table2.5: Default/Ghost Transitions of S3

Default/Ghost Transition	Test Case
Active(a, b, c)	Test#17
PayCredit()	Test#17
Reject()	Test#17
Approved()	Test#17
PayCash(c)	Test#17
StartPump()	Test#17
PumpLiter()	Test#17
StopPump()	Test#17
NoReceipt()	Test#17
Receipt()	Test#17
TurnOff()	Test#17

6. For the S4, the outgoing transition is T9,T18,T19:

Table2.6: Default/Ghost Transitions of S4

Default/Ghost Transition	Test Case
Active(a, b, c)	Test#18
PayCredit()	Test#18
Reject()	Test#18
Approved()	Test#18
PayCash(c)	Test#18
Regular()	Test#18
Super()	Test#18
Diesel()	Test#18
PumpLiter()	Test#18
StopPump()	Test#18
NoReceipt()	Test#18
Receipt()	Test#18
TurnOff()	Test#18

7. For the S5, the outgoing transition is T10,T11,T12:

Table2.7: Default/Ghost Transitions of S5

Default/Ghost Transition	Test Case
Active(a, b, c)	Test#19
PayCredit()	Test#19
Reject()	Test#19
Approved()	Test#19
Cancel()	Test#19
PayCash(c)	Test#19
Regular()	Test#19
Super()	Test#19
Diesel()	Test#19
StartPump()	Test#19
NoReceipt()	Test#19
Receipt()	Test#19
TurnOff()	Test#19

8. For the S6, the outgoing transition is T13,T14:

Table2.8: Default/Ghost Transitions of S6

Default/Ghost Transition	Test Case
Active(a, b, c)	Test#20
PayCredit()	Test#20
Reject()	Test#20
Approved()	Test#20
Cancel()	Test#20
PayCash(c)	Test#20
Regular()	Test#20
Super()	Test#20
Diesel()	Test#20
StartPump()	Test#20
PumpLiter()	Test#20
StopPump()	Test#20
TurnOff()	Test#20

9. For the ending state, there is no outgoing transition:

Table2.7: Default/Ghost Transitions of Ending State

Default/Ghost Transition	Test Case
Active(a, b, c)	Test#21
PayCredit()	Test#21
Reject()	Test#21
Approved()	Test#21
Cancel()	Test#21
PayCash(c)	Test#21
Regular()	Test#21
Super()	Test#21
Diesel()	Test#21
StartPump()	Test#21
PumpLiter()	Test#21
StopPump()	Test#21
NoReceipt()	Test#21
Receipt()	Test#21
TurnOff()	Test#21

3: Multiple-Condition Testing

Firstly, identify all the multiple conditions for each method in the source code.

1. For the method Activate (float a, float b, float d), there is only one multiple condition line:

   If $((k == -1)\ \&\&\ (a > 0)\ \&\&\ (b > 0)\ \&\&\ (d > 0))$

Table3.1: Multiple-Condition Testing for Active Method

k == -1	a > 0	b > 0	d > 0	Test Case
T	T	T	T	Test#23
T	T	T	F	Test#22
T	T	F	T	Test#22
T	T	F	F	Test#22
T	F	T	T	Test#22
T	F	T	F	Test#22
T	F	F	T	Test#22
T	F	F	F	Test#22
F	T	T	T	Test#23
F	T	T	F	Test#23
F	T	F	T	Test#23
F	T	F	F	Test#23
F	F	T	T	Test#23
F	F	T	F	Test#23
F	F	F	T	Test#23
F	F	F	F	Test#23

2. For the method PayCredit ()there is only one condition line:

   If $(k == 0)$

Table3.2: Multiple-Condition Testing for PayCredit Method

k == 0	Test Case
T	Test#25
F	Test#24

3. For the method Reject ()there is only one condition line:

   If (k == 2)

Table3.3: Multiple-Condition Testing for Reject Method

k == 2	Test Case
T	Test#25
F	Test#24

4. For the method Cancel ()there are two condition lines:

   1. If ((k == 3) || (k == 4))
   2. If (w == 0)

Table3.4.1: Multiple-Condition Testing for Cancel Method

k == 3	k == 4	Test Case
T	T	No Test Case 1
T	F	Test#25
F	T	Test#25
F	F	Test#24

Table3.4.2: Multiple-Condition Testing for Cancel Method

w == 0	Test Case
T	Test#25
F	Test#25

5. For the method Approved ()there is only one condition line:

   If (k == 2)

Table3.5: Multiple-Condition Testing for Approved Method

k == 2	Test Case
T	Test#25
F	Test#24

6. For the method PayCash (float c)there are two condition lines:

$$\text{If } \big((\text{k} == 0) \;\&\&\; (c > 0)\big)$$

Table3.6: Multiple-Condition Testing for PayCash Method

k == 0	c > 0	Test Case
T	T	Test#25
T	F	Test#25
F	T	Test#24
F	F	Test#24

7. For the method Regular ()there is only one condition line:

$$\text{If } (\text{k} == 3)$$

Table3.7: Multiple-Condition Testing for Regular Method

k == 3	Test Case
T	Test#25
F	Test#24

8. For the method Super ()there is only one condition line:

$$\text{If } (\text{k} == 3)$$

Table3.8: Multiple-Condition Testing for Super Method

k == 3	Test Case
T	Test#26
F	Test#24

9. For the method Diesel ()there is only one condition line:

$$\text{If } (\text{k} == 3)$$

Table3.9: Multiple-Condition Testing for Diesel Method

k == 3	Test Case
T	Test#27
F	Test#24

10.For the method Start Pump ()there is only one condition line:

If (k == 4)

Table3.10: Multiple-Condition Testing for StartPump Method

k == 4	Test Case
T	Test#26
F	Test#24

11.For the method PumpLiter ()there are four condition lines:

1. If (k == 5)
2. If $\Big((w == 1) \parallel \big((cash \geq price \times (L+1)\big) \&\& (w == 0)\Big)\Big)$
3. If $\big((w == 0) \&\& (cash < price \times (L+1))\big)$
4. If $\big((w == 0) \&\& (total < cash)\big)$

Table3.11.1: Multiple-Condition Testing for PumpLiter Method

k == 5	Test Case
T	Test#26
F	Test#24

Table3.11.2: Multiple-Condition Testing for PumpLiter Method

w == 1	cash >= price * (L + 1)	w == 0	Test Case
T	T	T	No Test Case 2
T	T	F	No Test Case 3
T	F	T	No Test Case 4
T	F	F	Test#28
F	T	T	Test#26
F	T	F	No Test Case 5
F	F	T	Test#26
F	F	F	No Test Case 6

Table3.11.3: Multiple-Condition Testing for PumpLiter Method

w == 0	cash < price * (L + 1)	Test Case
T	T	Test#26
T	F	No Test Case 7
F	T	Test#29
F	F	No Test Case 8

Table3.11.4: Multiple-Condition Testing for PumpLiter Method

w == 0	total < cash	Test Case
T	T	Test#26
T	F	No Test Case 9
F	T	No Test Case 10
F	F	No Test Case 11

12.For the method StopPump ()there are four condition lines:

1. If (k == 5)
2. If ((w == 0) && (total < cash))

Table3.12.1: Multiple-Condition Testing for StopPump Method

k == 5	Test Case
T	Test#27
F	Test#24

Table3.12.2: Multiple-Condition Testing for StopPump Method

w == 0	total < cash	Test Case
T	T	Test#27
T	F	No Test Case 12
F	T	Test#29
F	F	Test#28

13.For the method NoReceipt ()there is only one condition line:

If (k == 6)

Table3.13: Multiple-Condition Testing for NoReceipt Method

k == 6	Test Case
T	Test#26
F	Test#24

14.For the method Receipt ()there is only one condition line:

If (k == 6)

Table3.14: Multiple-Condition Testing for Receipt Method

k == 6	Test Case
T	Test#27
F	Test#24

15.For the method TurnOff ()there is only one condition line:

If (k == 0)

Table3.15: Multiple-Condition Testing for TurnOff Method

k == 0	Test Case
T	Test#25
F	Test#24

4: Reasons of Non-Executable Branches

Here is the table that explain why there are some situation has no test case, that is, these are non-executable branches:

Table4: Non-Executable Reason

Number of No Test Case	Reason
1	k cannot equal to both 3 and 4.
2	w cannot equal to both 1 and 0.
3	w = 1 means using credit, so cash = 0, and the price cannot be 0, so cash >= price * (L + 1) cannot be true.
4	w cannot equal to both 1 and 0.
5	w either equals to 1 or 0.
6	w either equals to 1 or 0.
7	When w= 0 and cash < price * (L + 1) is false, this means w = 0 and cash >= price * (L + 1) is true, this satisfied the previous if line, so there is no chance to get into this line.
8	w = 1 satisfy the previous if line, so there is no chance to get into this line.
9	When w = 0, means pay cash, the total cannot greater than cash.
10	This if is contains in the previous if, so when we get here, the w must be 0.
11	This if is contains in the previous if, so when we get here, the w must be 0.
12	When w = 0, means pay cash, the total cannot greater than cash.

5: Test Cases

5.1 Test cases for transition pairing testing

Here is the table of test cases for the transition pairing testing:

Table5.1.1: Test Cases for Transition Pairing Testing

Test#	Test Details	Transitions	Supposed State	Pass
Test#1	Activate 2 3 4 PayCredit Reject PayCash 20 Diesel StartPump PumpLiter PumpLiter StopPump NoReceipt TurnOff	T1 T2 T3 T6 T15 T9 T10 T10 T11 T14 T20	Start S0 S2 S0 S3 S4 S5 S5 S5 S6 S0 End	Yes
Test#2	Activate 2 3 4 PayCash 20 Cancel PayCash 20 Super StartPump StopPump NoReceipt PayCash 20 Regular Cancel TurnOff	T1 T6 T5 T6 T8 T9 T11 T14 T6 T7 T19 T20	Start S0 S3 S0 S3 S4 S5 S6 S0 S3 S4 S0 End	Yes
Test#3	Activate 2 3 4 PayCash 20 Cancel PayCredit Approved Super Cancel PayCredit Approved Diesel Cancel PayCash 1 Regular StartPump PumpLiter Receipt PayCredit Reject TurnOff	T1 T6 T5 T2 T4 T8 T19 T2 T4 T15 T19 T6 T7 T9 T12 T13 T2 T3 T20	Start S0 S3 S0 S2 S3 S4 S0 S2 S3 S4 S0 S3 S4 S5 S6 S0 S2 S0 End	Yes
Test#4	Activate 2 3 4 PayCash 3 Regular StartPump PumpLiter PumpLiter NoReceipt TurnOff	T1 T6 T7 T9 T10 T12 T14 T20	Start S0 S3 S4 S5 S5 S6 S0 End	Yes
Test#5	Activate 2 3 4 PayCash 20 Regular StartPump StopPump Receipt TurnOff	T1 T6 T7 T9 T11 T13 T20	Start S0 S3 S4 S5 S6 S0 End	Yes
Test#6	Activate 2 3 4 TurnOff	T1 T20	Start S0 End	Yes
Test#7	Activate 2 3 4 PayCredit Reject PayCredit Approved Cancel TurnOff	T1 T2 T3 T2 T4 T5 T20	Start S0 S2 S0 S2 S3 S0 End	Yes

Test#8	Activate 2 3 4 PayCredit Approved Regular Cancel PayCash 20 Regular StartPump StopPump Receipt PayCash 20 Cancel TurnOff	T1 T2 T4 T7 T19 T6 T7 T9 T11 T13 T6 T5 T20	Start S0 S2 S3 S4 S0 S3 S4 S5 S6 S0 S3 S0 End	Yes
Test#9	Activate 2 3 4 PayCash 20 Regular StartPump StopPump NoReceipt PayCredit Reject TurnOff	T1 T6 T7 T9 T11 T14 T2 T3 T20	Start S0 S3 S4 S5 S6 S0 S2 S0 End	Yes
Test#10	Activate 2 3 4 PayCredit Approved Regular StartPump StopPump Receipt TurnOff	T1 T2 T4 T7 T18 T16 T13 T20	Start S0 S2 S3 S4 S1 S6 S0 End	No
Test#11	Activate 2 3 4 PayCredit Approved Super StartPump PumpLiter PumpLiter StopPump NoReceipt TurnOff	T1 T2 T4 T8 T18 T17 T17 T16 T14 T20	Start S0 S2 S3 S4 S1 S1 S1 S6 S0 End	No
Test#12	Activate 2 3 4 PayCredit Approved Diesel StartPump StopPump Receipt TurnOff	T1 T2 T4 T15 T18 T16 T13 T20	Start S0 S2 S3 S4 S1 S6 S0 End	No

Here is the detail information table for the failed test cases:

Table5.1.2: Failed Test Cases for Transition Pair Testing

Test#	Test Details	Supposed State	Actual State
Test#10	Activate 2 3 4 PayCredit Approved Regular StartPump StopPump Receipt TurnOff	Start S0 S2 S3 S4 S1 S6 S0 End	Start S0 S2 S3 S4 S5 S6 S0 End
Test#11	Activate 2 3 4 PayCredit Approved Super StartPump PumpLiter PumpLiter StopPump NoReceipt TurnOff	Start S0 S2 S3 S4 S1 S1 S1 S6 S0 End	Start S0 S2 S3 S4 S5 S5 S5 S6 S0 End
Test#12	Activate 2 3 4 PayCredit Approved Diesel StartPump StopPump Receipt TurnOff	Start S0 S2 S3 S4 S1 S6 S0 End	Start S0 S2 S3 S4 S5 S6 S0 End

5.2 Test cases for default/ghost transitions testing

Here is the table of test cases for the default/ghost transitions testing:

Table5.2.1: Test Cases for Default/Ghost Transitions Testing

Test#	Test Details	Supposed States	Pass
Test#13	Activate -2 3 4 PayCredit Reject Approved Cancel PayCash 20 Regular Super Diesel StartPump PumpLiter StopPump NoReceipt Receipt TurnOff	All Starting State	Yes
Test#14	Activate 2 3 4 Activate 2 3 4 Reject Approved Cancel Regular Super Diesel StartPump PumpLiter StopPump NoReceipt Receipt	Starting State, All S0	Yes
Test#15	Activate 2 3 4 PayCredit Approved Regular StartPump Activate 2 3 4 PayCredit Reject Approved Cancel PayCash 20 Regular Super Diesel StartPump NoReceipt Receipt TurnOff	Starting State S0, S2, S3, S4, All S1	No
Test#16	Activate 2 3 4 PayCredit Activate 2 3 4 PayCredit Cancel PayCash 20 Regular Super Diesel StartPump PumpLiter StopPump NoReceipt Receipt TurnOff	Starting State S0, All S2	Yes
Test#17	Activate 2 3 4 PayCredit Approved Activate 2 3 4 PayCredit Reject Approved PayCash 20 StartPump PumpLiter StopPump NoReceipt Receipt TurnOff	Starting State S0, S2, All S3	Yes
Test#18	Activate 2 3 4 PayCredit Approved Regular Activate 2 3 4 PayCredit Reject Approved PayCash 20 Regular Super Diesel PumpLiter StopPump NoReceipt Receipt TurnOff	Starting State S0, S2, S3, All S4	Yes

Test#19	Activate 2 3 4 PayCash 20 Regular StartPump Activate 2 3 4 PayCredit Reject Approved Cancel PayCash 20 Regular Super Diesel StartPump NoReceipt Receipt TurnOff	Starting State S0, S3, S4, All S5	Yes
Test#20	Activate 2 3 4 PayCash 20 Regular StartPump StopPump Activate 2 3 4 PayCredit Reject Approved Cancel PayCash 20 Regular Super Diesel StartPump PumpLiter StopPump TurnOff	Starting State S0, S3, S4, S5 All S6	Yes
Test#21	Activate 2 3 4 TurnOff Activate 2 3 4 PayCredit Reject Approved Cancel PayCash 20 Regular Super Diesel StartPump PumpLiter StopPump NoReceipt Receipt TurnOff	Starting State S0, All Ending State	Yes

Here is the detail information table for the failed test cases:

Table5.2.2: Failed Test Cases for Transition Pair Testing

Test#	Test Details	Supposed State	Actual State
Test#15	Activate 2 3 4 PayCredit Approved Regular StartPump Activate 2 3 4 PayCredit Reject Approved Cancel PayCash 20 Regular Super Diesel StartPump NoReceipt Receipt TurnOff	Starting State S0, S2, S3, S4, All S1	Starting State S0, S2, S3, S4, All S5

5.3 Test cases for multiple-condition testing

Here are the tables of test cases for multiple condition testing, there are some value sequences that listed in the table 5.3.3 to table 5.3.7:

Table5.3.1: Test Cases for Multiple Condition Testing

Test#	Test Details	Values	Pass
Test#22	Activate 2 3 -4 Activate 2 -3 4 Activate 2 -3 -4 Activate -2 3 4 Activate -2 3 -4 Activate -2 -3 4 Activate -2 -3 -4	Only care the value of K: all -1	Yes
Test#23	Activate 2 3 4 Activate 2 3 4 Activate 2 3 -4 Activate 2 -3 4 Activate 2 -3 -4 Activate -2 3 4 Activate -2 3 -4 Activate -2 -3 4 Activate -2 -3 -4	Only care the value of K: -1, all 0	Yes
Test#24	PayCredit Reject Approved Cancel PayCash 20 PayCash -20 Regular Super Diesel StartPump PumpLiter StopPump NoReceipt Receipt TurnOff	Only care the value of K: all -1	Yes
Test#25	Activate 2 3 4 PayCredit Reject PayCredit Approved Cancel PayCash -20 PayCash 20 Regular Cancel TurnOff	Table 5.3.3	Yes
Test#26	Activate 2 3 4 PayCash 5 Super StartPump PumpLiter PumpLiter NoReceipt TurnOff	Table 5.3.4	Yes
Test#27	Activate 2 3 4 PayCash 5 Diesel StartPump StopPump Receipt TurnOff	Table 5.3.5	Yes
Test#28	Activate 2 3 4 PayCredit Approved Regular StartPump PumpLiter StopPump Receipt TurnOff	Table 5.3.6	No
Test#29	Activate 2 3 4 PayCash 20 Cancel PayCredit Approved Regular StartPump PumpLiter StopPump Receipt TurnOff	Table 5.3.7	No

For Test#28 and Test#29, the states are not as supposed, but the value sequence has test the branches we want. This is the table that analysis the failure of test#28:

Table5.3.2: Failure Details

Test#	Test Details	Supposed State	Actual State
Test#28	Activate 2 3 4 PayCredit Approved Regular StartPump PumpLiter StopPump Receipt TurnOff	Starting State S0, S2, S3, S4, S1, S1, S6, S0, End	Starting State S0, S2, S3, S4, S5, S5, S6, S0, End
Test#29	Activate 2 3 4 PayCash 20 Cancel PayCredit Approved Regular StartPump PumpLiter StopPump Receipt TurnOff	Starting State S0, S3, S0, S2, S3, S4, S1, S1, S6, S0, End	Starting State S0, S3, S0, S2, S3, S4, S5, S5, S6, S0, End

This is the detail value sequence for test#25, R is the price of regular, S is the price of the super, and D is the price of diesel:

Table5.3.3: Value Sequence for Test#25

Operation	R	S	D	w	price	L	total	cash	k
Starting	0	0	0	0	0	0	0	0	-1
Activate 2 3 4	2	3	4	0	0	0	0	0	0
PayCredit	2	3	4	0	0	0	0	0	2
Reject	2	3	4	0	0	0	0	0	0
PayCredit	2	3	4	0	0	0	0	0	2
Approved	2	3	4	1	0	0	0	0	3
Cancel	2	3	4	1	0	0	0	0	0
PayCash -20	2	3	4	1	0	0	0	0	0
PayCash 20	2	3	4	0	0	0	0	20	3
Regular	2	3	4	0	2	0	0	20	4
Cancel	2	3	4	0	2	0	0	20	0
TurnOff	2	3	4	0	2	0	0	20	-2

This is the detail value sequence for test#26, R is the price of regular, S is the price of the super, and D is the price of diesel:

Table5.3.4: Value Sequence for Test#26

Operation	R	S	D	w	price	L	total	cash	k
Starting	0	0	0	0	0	0	0	0	-1
Activate 2 3 4	2	3	4	0	0	0	0	0	0
PayCash 5	2	3	4	0	0	0	0	5	3
Super	2	3	4	0	3	0	0	5	4
StartPump	2	3	4	0	3	0	0	5	5
PumpLiter	2	3	4	0	3	1	3	5	5
PumpLiter	2	3	4	0	3	1	3	5	6
NoReceipt	2	3	4	0	3	1	3	5	0
TurnOff	2	3	4	0	3	1	3	5	3

This is the detail value sequence for test#27, R is the price of regular, S is the price of the super, and D is the price of diesel:

Table5.3.5: Value Sequence for Test#27

Operation	R	S	D	w	price	L	total	cash	k
Starting	0	0	0	0	0	0	0	0	-1
Activate 2 3 4	2	3	4	0	0	0	0	0	0
PayCash 5	2	3	4	0	0	0	0	5	3
Diesel	2	3	4	0	4	0	0	5	4
StartPump	2	3	4	0	4	0	0	5	5
StopPump	2	3	4	0	4	0	0	5	6
Receipt	2	3	4	0	4	0	0	5	0
TurnOff	2	3	4	0	4	0	0	5	-2

This is the detail value sequence for test#28, R is the price of regular, S is the price of the super, and D is the price of diesel:

Table5.3.6: Value Sequence for Test#28

Operation	R	S	D	w	price	L	total	cash	k
Starting	0	0	0	0	0	0	0	0	-1
Activate 2 3 4	2	3	4	0	0	0	0	0	0
PayCredit	2	3	4	0	0	0	0	0	2
Approved	2	3	4	1	0	0	0	0	3

Regular	2	3	4	1	2	0	0	0	4
StartPump	2	3	4	1	2	1	2	0	5
PumpLiter	2	3	4	1	2	1	2	0	5
StopPump	2	3	4	1	2	1	2	0	6
Receipt	2	3	4	1	2	1	2	0	0
TurnOff	2	3	4	1	2	1	2	0	-2

This is the detail value sequence for test#29, R is the price of regular, S is the price of the super, and D is the price of diesel:

Table5.3.7: Value Sequence for Test#29

Operation	**R**	**S**	**D**	**w**	**price**	**L**	**total**	**cash**	**k**
Starting	0	0	0	0	0	0	0	0	-1
Activate 2 3 4	2	3	4	0	0	0	0	0	0
PayCash 20	2	3	4	0	0	0	0	20	3
Cancel	2	3	4	0	0	0	0	20	0
PayCredit	2	3	4	0	0	0	0	20	2
Approved	2	3	4	1	0	0	0	20	3
Regular	2	3	4	1	2	0	0	20	4
StartPump	2	3	4	1	2	0	0	20	5
PumpLiter	2	3	4	1	2	1	2	20	5
StopPump	2	3	4	1	2	1	2	20	6
Receipt	2	3	4	1	2	1	2	20	0
TurnOff	2	3	4	1	2	1	2	20	-2

6: Analysis of Failed Test Cases and Source Code Defects

From the table 5.1.2 and table 5.2.2, we get all the failed test cases and their failure information.

We notice that all the S1 state are replace by the S5 state. They all show that all the test cases about S1 will fail.

Then, we look into the source code and find the reason.

1. In the StartPump method, we notice that the k only set to 5, this means the StartPump can only point to the state S5, but based on the EFSM, we know that StartPump can either point to S5 or S1. So, this method should add predicate line to separate the k's value base on the value of w. If w equals to 0, k will be 5, if w equals to 1, k will be 1.
2. In the PumpLiter method, the first if line only accept the k equals to 5. Based on the EFSM, PumpLiter can appear in S5 or S1, this means if line should accept k equals to 5 or 1.
3. In the StopPump method, the first if line only accept the k equals to 5. Based on the EFSM, PumpLiter can appear in S5 or S1, this means if line should accept k equals to 5 or 1.

Also we notice that after paying by cash, there is no chance to reset the cash value to be zero, so the Cancel method and the Receipt and NoReceipt methods should add the statement to reset the cash value to be zero.

7: Conclusion

In this project, I have designed the testing driver and use this test driver to run the test cases.

When I design the testing driver, I firstly try to generate some testing oriented methods in the source code. At first, I think maybe I can use a string buffer that will save the transitions. But this need to add some if statements into each method. So, I delete this method. Then, I add the methods that can only "watch" the program, and they have no influence to the original methods. So, I use the value of k to present the state number, and add the method that can show all the variables' value.

And I implement the testing environment in Java. In order to make the testing driver can run directly on the PC without Java environment, I export the Java program as a Jar file. This is the new experience for the Java programming.

When I use the testing driver to run test cases, I have to enter each operation and then press "a" to show variables' value and press "b" to show state number. Entering the test operations takes me a lot of time.

When entering the test operations, I concentrate on different points. For the model based testing, I only care about the state numbers, for the code based testing, I only care about the variables' value.

For the automation of class testing, I think firstly, the test driver should be able to read the test case by itself and run the test cases by itself. Also, test driver should show the variable for each operation and the branch.

The class testing is based on the code, we can scan through the code, find the predicate statements and separate the conditions based on the connections. So, we can automatically get all the simple conditions for each predicate statement.