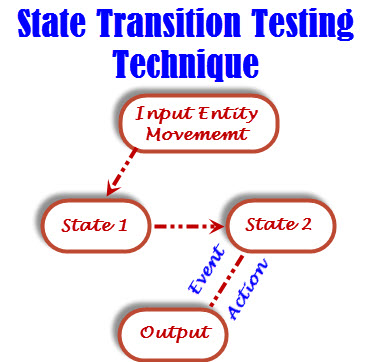
State Transition Testing

State transition testing is a form of **Dynamic Testing Technique** that comes in use when the system explained as a finite number of states and the evolutions between the states is ruled by the rules of the system. Another use of this technique when features of a system are characterized as states that converts to other state, this transition is explained by the method of the software

The diagrammatic explanation is shown below,

[](http://www.softwaretestingclass.com/wp-content/uploads/2014/01/state-transition-testing-technique.jpg)

So, the diagram shows that, the input condition has came the reason of an entity transitions from State 1 to State 2 that guides to an event and results to an action and finally gives the output.

**Four major parts of state transition model:**

* States that the software might get (open/closed or sufficient/insufficient funds)
* The transition from one state to another (with single transitions)
* The events that origin a transition (closing a file or withdrawing money)
* Actions that result from a transition (an error message or being given the cash)

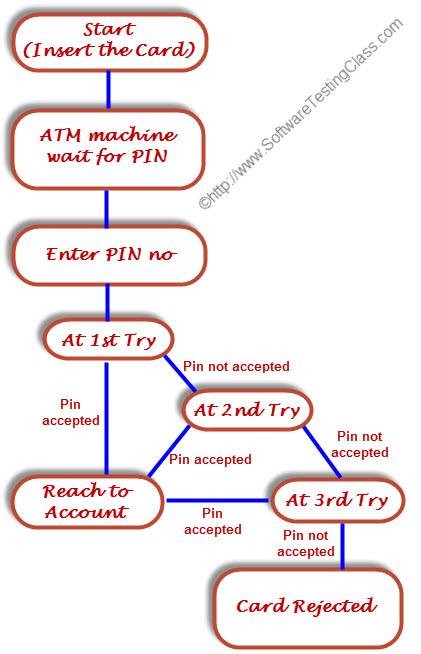
For any given state, one event origins only one action, but that the same event from a different state might origin a different action and a different end state.

**An example to explain** **State transition testing technique:**

Go to ATM machine to withdraw $1000, requires to the ATM machine and get cash. Later on, go to the ATM and give the same request at same amount but this time ATM refuses the amount because of insufficient balance amount.

This refusal happened just because of bank account has been changed from one state (sufficient funds) to another state (insufficient funds). The transaction that caused the account to change its state was maybe the earlier withdrawal.

Below diagram shows that how ATM get process using Personal Identity Number (PIN).

[](http://www.softwaretestingclass.com/wp-content/uploads/2014/01/state-transition-testing-technique-example.jpg)

The above state diagram shows several states and four events: Insert Card, Enter PIN, PIN accepted and PIN not accepted). There would also be a transition from the “Card Rejected” state back to the start state.

There should be a ‘cancel’ option at the time of ‘wait for PIN’ process and at the time of three tries that should also go back to the start state and eject the card.

Even think, still this state diagram is incomplete; it also gives us some more information that should be helpful in designing the test cases and explaining the state transition technique.

Let’s take some scenario to develop test cases,

* The first test case should be very much sensible to enter the correct PIN at the first time
* The second test should be to enter an incorrect PIN each time, so that the system rejects the card
* To test all transition, firstly do the testing where the PIN was incorrect the first time but OK the second time and another test where the PIN was correct on the third try
* But, these tests are basically less important than the first two tests

The major advantages of the state transition technique is that the model can be as detailed or as abstract as you need it to be. Where a part of the system is more significant a larger depth of detail can be modelled. Where the system is less important, the model can use a single state to indicate what would otherwise be a series of different states.

The Coverage of single transition is also called as 0-switch coverage, the coverage of pair’s transition is 1-switch coverage, coverage of triples transition is 2-switch coverage, etc.

Explaining test cases from the state transition model is a black-box approach. Determining how much test has been completed is a white-box perspective. However, state transition testing is regarded as a black-box technique. Determining tests simply from a state graph (also known as a state chart) is wonderful for looking the valid transitions, but we may not simply look the negative tests, where we try to make invalid transitions. In order to see the total number of mixtures of states and transitions, both valid and invalid, a state table is useful.

Table below shows the state table lists all the states down at one side of the table and all the events that origin transitions along the top (or vice versa). Every cell then symbolizes a state-event pair.

The information of each cell shows which state the system will shift to, when the resultant event arises while in the associated state. This will contain possible erroneous events – events that are not accepted to happen in certain states. These are negative test conditions.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Insert Card** | **Valid PIN** | **InValid PIN** |
| Start State | S2 | – | – |
| Wait for PIN | – | S6 | S3 |
| 1st try invalid | – | S6 | S4 |
| 2nd try invalid | – | S6 | S5 |
| 3rd try invalid | – | – | S7 |
| Access Account | – | ? | ? |
| Card not excepted | S1 (for new card) | – | – |

This Table shows different states in the first column and the respective inputs across the top row. For Example:

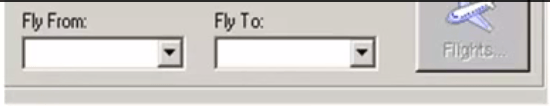
* Suppose the system is in state 1, after inserting the card it will go to state 2.
* If system is in state 2, after entering a valid PIN, it goes to State 6 to access the account. In State 2 if we enter an invalid PIN it goes to State 3.

In the table, a dash (-) sign represents invalid transitions of that state. A question mark in two cells – either a valid or invalid PIN, when we are using the account. Possibly the system will catch our PIN number as the amount of cash to withdraw? Other invalid cells would be actually not possible in this example. Invalid (negative) tests will try to make invalid transitions, transitions that shouldn’t be possible.

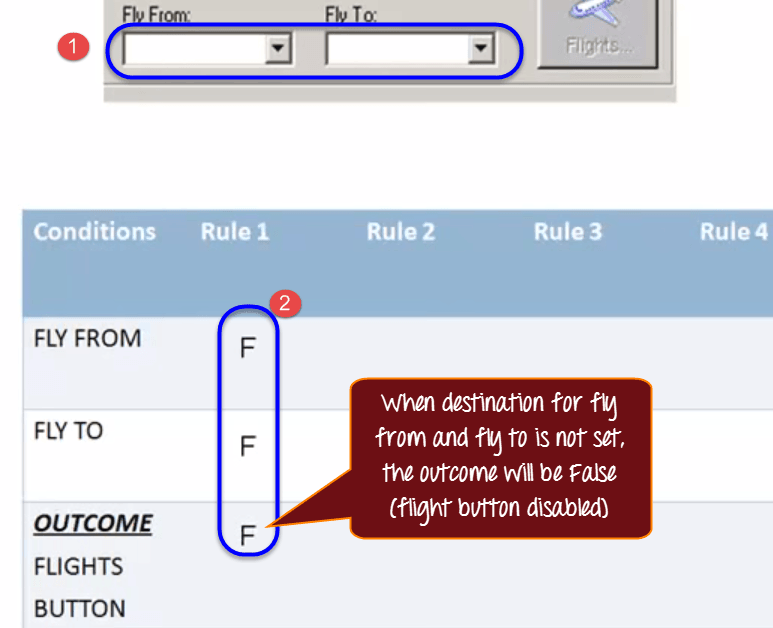
## What is Decision Table Testing?

Decision Table Testing is a good way to deal with a combination of inputs, which produce different results. It helps reduce test effort in verifying each and every combinations of test data, at the same time ensuring complete coverage

Example: To understand the importance of Decision Table Making we will see an example, let's consider the behavior of Flight Button for different combinations of Fly From & Fly To.

[](http://cdn.guru99.com/images/1-2015/012715_0719_DecisionTab1.png)

**Rule 1:**When destination for both Fly From & Fly To are not set the Flight Icon is disabled. In the decision table, we register values False for Fly From & Fly To and the outcome would be False, which is Flights Button will be disabled. Likewise, you can decide different outcome for different situation.

[](http://cdn.guru99.com/images/1-2015/012715_0719_DecisionTab2.png)

**Rule 2**: When Fly From destination is set but Fly to is not set, Flight button is disabled. Correspondingly, you register True for Fly from destination in the decision table, and the rest of the entries are false.

**Rule 3**: When Fly from destination is not set but Fly to destination is set, Flight button is disabled and you make entries in the decision table.

**Rule 4:** only when Fly to and Fly from destinations are set, Flights button is enabled and you make the corresponding entry in the decision table.

[](http://cdn.guru99.com/images/1-2015/012715_0719_DecisionTab3.png)

If you observe the outcomes for Rule 1, 2 & 3 it remains the same. So you can  select any one of them and rule 4 for your testing.

The **significance of this technique becomes immediately clear as the number of inputs increases**.**Number of possible Combinations is given by 2 ^ n , where n is  the number of  Inputs.** **For n = 10,** which is very common in the web based testing, having big input forms, the number of **combinations will be 1024.** Obviously, you cannot test all but **you will choose a rich sub-set of the possible combinations using decision based testing technique**

**Let’s take another example. If you are a new customer and you want to open a credit card account then there are three conditions first you will get a 15% discount on all your purchases today, second if you are an existing customer and you hold a loyalty card, you get a 10% discount and third if you have a coupon, you can get 20% off today (but it can’t be used with the ‘new customer’ discount). Discount amounts are added, if applicable**

