



# STATE TABLE AND DECISION TABLE BASED TESTING

**Dr. Sangharatna Godbole,**  
**Assistant Professor,**  
**Dept. of CSE, NIT Warangal**

# STATE TABLE – BASED TESTING

- Tables are useful tools for representing and documenting many types of information relating to test case design.
- These are beneficial for applications which can be described using state transition diagrams and state tables.



# **Basic terms related to State Table**

## **I. Finite State Machine (FSM)**

- An FSM is a behavioral model whose outcome depends upon both the previous and current inputs.
- This model can be prepared for software structure or software behavior.
- It can be used as a tool for functional testing.

## 2. State Transition Diagrams or State Graph

- A system or its components may have a number of states depending on its input and time.
- States are represented by nodes.
- *With the help of nodes and transition links between nodes, a STD or SG can be prepared.*

# State Graph

- A state graph is the pictorial representation of an FSM.
- Its purpose is to depict the states that a system or its components can assume.
- It shows the events or circumstances that cause or result from a change from one state to another.

# State Graph cont ...

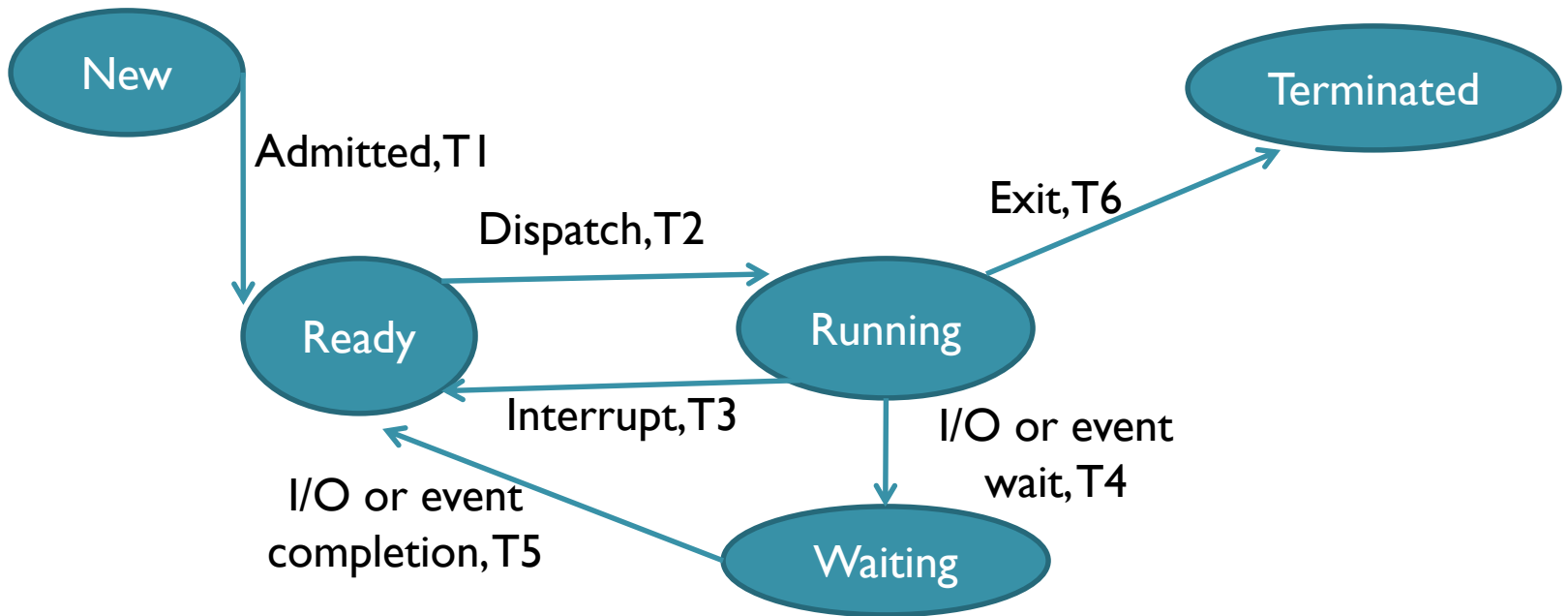
- Whatever is being modeled is subjected to inputs.
- As a result of these inputs, when one state is changed to another is called a transition.
- Transitions are represented by links that join the nodes.

# Example

For example, a task in an O. S. can have its following states:

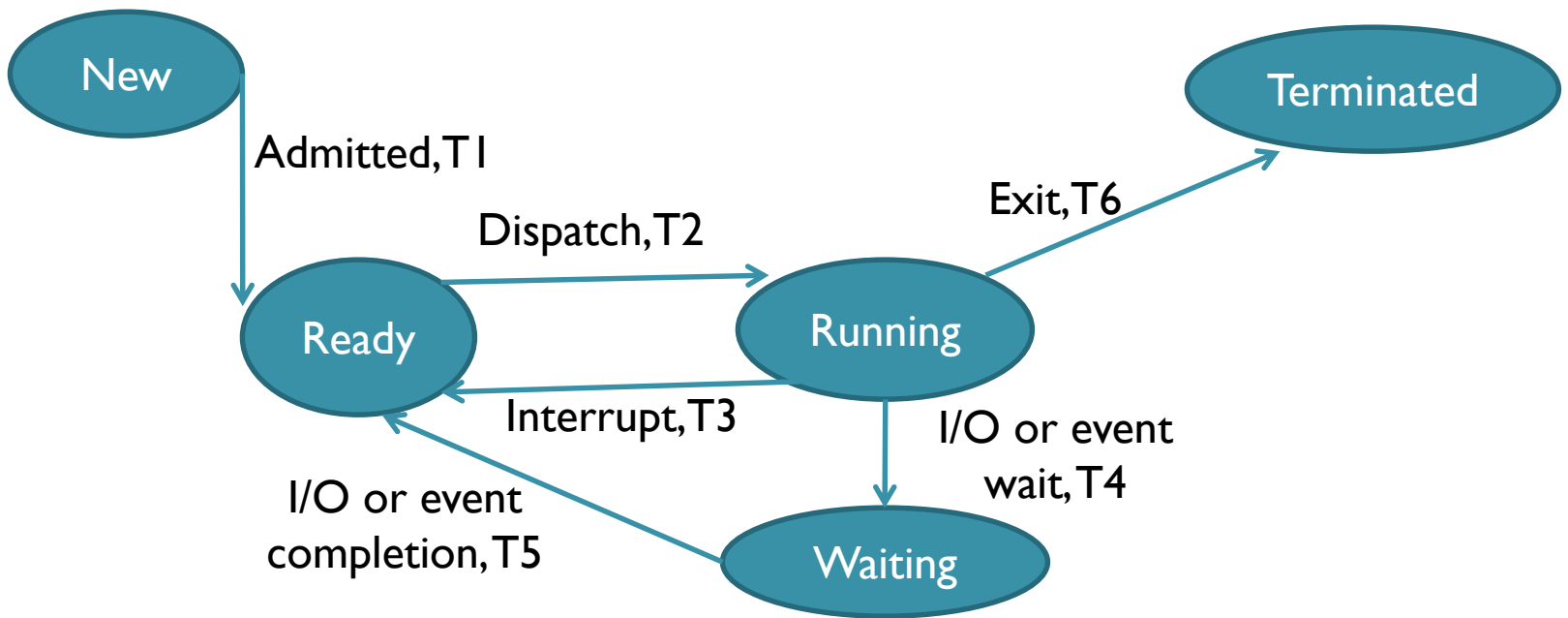
- i. **New State** : When a task is newly created
- ii. **Ready** : When the task is waiting in the ready queue for its turn.
- iii. **Running** : When Instructions of the task are being executed by CPU.
- iv. **Waiting** : When the task is waiting for an I/O event or reception of a signal
- v. **Terminated** : The task has finished execution.

# State Graph



- New State :When a task is newly created
- Ready :When the task is waiting in the ready queue for its turn.
- Running :When Instructions of the task are being executed by CPU.
- Waiting :When the task is waiting for an I/O event or reception of a signal
- Terminated :The task has finished execution.





## Each arrow link provides two types of Information :

1. Transition events like admitted, dispatch, interrupt, etc.
2. The resulting output from a state like T1, T2, T3 etc.

**T0=Task is in new state and waiting for admission to ready queue.**

**T1= A new task admitted to ready queue**

**T2= A ready task has started running**

**T3= Running task has been interrupted**

**T4= Running task is waiting for I/O or event**

**T5= Wait period of waiting task is over**

**T6= Task has completed execution**

### 3. State Table

| State/input Event | Admit           | Dispatch           | Interrupt         | I/O or event Wait | I/O or event Wait Over | Exit                 |
|-------------------|-----------------|--------------------|-------------------|-------------------|------------------------|----------------------|
| New               | <b>Ready/T1</b> | New / T0           | New / T0          | New / T0          | New / T0               | New / T0             |
| Ready             | Ready / T1      | <b>Running /T2</b> | Ready / T1        | Ready / T1        | Ready / T1             | Ready / T1           |
| Running           | Running /T2     | Running /T2        | <b>Ready / T3</b> | <b>Waiting/T4</b> | Running/T2             | <b>Terminated/T6</b> |
| Waiting           | Waiting/T4      | Waiting/T4         | Waiting/T4        | Waiting/T4        | <b>Ready /T5</b>       | Waiting/T4           |

- Each rows of the table corresponds to a state.
- Each column corresponds to an input condition
- The box at the intersection of a row and a column specifies the next state (transition) and the outputs, if any.

## 4. State Table-Based Testing

- After reviewing the basics, we can start functional testing with state tables.

### Steps:

#### 1. Identify the states

The number of states in a state graph is the number of states we choose to recognize or model.

# Finding the number of states

- First, identify all the component factors of the state.
- Identify all the allowable values for each factor
- The number of states is the product of the number of allowable values of all the factors.

## Steps cont ...

### **2. Prepare state transition diagram after understanding transitions between states**

- After having all the states, identify the inputs on each state and transitions between states and prepare the state graph.
- Every input state combination must have a specified transition.

## Steps cont ...

- 3. Convert the state graph into the state table as discussed earlier**
- 4. Analyze the state table for its completeness.**

## Steps cont ...

**5. Develop the test cases from the state table, in a tabular form, with the following entries.**

- **Test cases ID** : a unique identifier for each test case
- **Test Source** : a trace back to the corresponding cell in the state table.
- **Current state** : the initial condition to run the test
- **Event** : the input triggered by the user
- **Output** : the current value returned
- **Next State** : the new state achieved

# Test Cases for the O.S. Example

| Test case ID | Test Source | Input         |               | Expected results |            |
|--------------|-------------|---------------|---------------|------------------|------------|
|              |             | Current State | Event         | Output           | Next state |
| TC1          | Cell 1      | New           | Admit         | T1               | Ready      |
| TC2          | Cell 2      | New           | Dispatch      | T0               | New        |
| TC3          | Cell 3      | New           | Interrupt     | T0               | New        |
| TC4          | Cell 4      | New           | I/O wait      | T0               | New        |
| TC5          | Cell 5      | New           | I/O wait over | T0               | New        |
| TC6          | Cell 6      | New           | Exit          | T0               | New        |
| TC7          | Cell 7      | Ready         | Admit         | T1               | Ready      |
| TC8          | Cell 8      | Ready         | Dispatch      | T2               | Running    |
| TC9          | Cell 9      | Ready         | Interrupt     | T1               | Ready      |
| TC10         | Cell 10     | Ready         | I/O wait      | T1               | Ready      |
| TC11         | Cell 11     | Ready         | I/O wait over | T1               | Ready      |
| TC12         | Cell 12     | Ready         | Exit          | T1               | Ready      |
| TC13         | Cell 13     | Running       | Admit         | T2               | Running    |
| TC14         | Cell 14     | Running       | Dispatch      | T2               | Running    |
| TC15         | Cell 15     | Running       | Interrupt     | T3               | Ready      |
| TC16         | Cell 16     | Running       | I/O wait      | T4               | Waiting    |
| TC17         | Cell 17     | Running       | I/O wait over | T2               | Running    |
| TC18         | Cell 18     | Running       | Exit          | T6               | Terminated |
| TC19         | Cell 19     | Waiting       | Admit         | T4               | Waiting    |
| TC20         | Cell 20     | Waiting       | Dispatch      | T4               | Waiting    |
| TC21         | Cell 21     | Waiting       | Interrupt     | T4               | Waiting    |
| TC22         | Cell 22     | Waiting       | I/O wait      | T4               | Waiting    |
| TC23         | Cell 23     | Waiting       | I/O wait over | T5               | Ready      |
| TC24         | Cell 24     | Waiting       | Exit          | T4               | Waiting    |



# Decision Table – Based Testing

- Boundary value analysis and equivalence class partitioning methods do not consider combinations of input conditions.
- There may be some critical behavior to be tested when some combinations of input conditions are considered.
- Decision table is another useful method to represent the information in a tabular method.
- Decision table has the specialty to consider complex combinations of input conditions and resulting actions.
- Decision tables obtain their power from logical expressions.
- Each operand or variable in a logical expression takes on the value, TRUE or FALSE

# Formation of Decision Table

| Condition Stub |    | Rule 1 | Rule 2 | Rule 3 | Rule 4 | ... |
|----------------|----|--------|--------|--------|--------|-----|
|                | C1 | True   | True   | False  | I      |     |
|                | C2 | False  | True   | False  | True   |     |
|                | C3 | True   | True   | True   | I      |     |
| Action Stub    | A1 |        |        |        |        |     |
|                | A2 | X      | X      |        | X      |     |
|                |    |        |        | X      |        |     |
|                | A3 |        |        |        |        |     |

- **Condition stub** It is a list a list of input conditions for which the complex combination is made.
- **Action stub** It is a list of resulting action which will be performed if a combination of input condition is satisfied.

## Condition entry

- It is a specific entry in the table corresponding to input conditions mentioned in the condition stub.
- When the condition entry takes only two values – TRUE or FALSE then it is called ***Limited Entry Decision Table.***
- When the condition entry takes several values , then it is called ***Extended Entry Decision Table.***



**Action entry** It is the entry in the table for the resulting action to be performed.

- List all actions that can be associated with a specific procedure.
- List all conditions during execution of the procedure.
- Associate specific sets of conditions with specific actions, eliminating impossible combinations and conditions; alternatively, develop every possible permutation of conditions.
- Define rules by indicating what action occurs for a set of conditions.

# Test Case Design using Decision Table

- Interpret condition stubs as the inputs for the test case.
- Interpret action stubs as the expected output for the test case.
- Rule, which is the combination of input conditions, becomes the test case itself.
- If there are  $k$  rules over  $n$  binary conditions, there are at least  $k$  test cases and at the most  $2^n$  test cases.

# Example - I

A program calculates the total salary of an employee with the conditions that if the working hours are less than or equal to 48, then give normal salary. The hours over 48 normal working days are calculated at the rate of 1.25 of the salary. However, on holidays or Sundays, the hours are calculated at the rate of 2.00 times of the salary. Design test cases using decision table testing.

## Solution

|                       |                         | Entry  |        |        |
|-----------------------|-------------------------|--------|--------|--------|
|                       |                         | Rule 1 | Rule 2 | Rule 3 |
| <b>Condition Stub</b> | C1: Working hours > 48  | I      | F      | T      |
|                       | C2: Holidays or Sundays | T      | F      | F      |
| <b>Action Stub</b>    | A1: Normal Salary       |        | X      |        |
|                       | A2: 1.25 of salary      |        |        | X      |
|                       | A3: 2.00 of salary      | X      |        |        |

# Decision Table

| Test case ID | Working hour | Day     | Expected Result |
|--------------|--------------|---------|-----------------|
| 1            | 48           | Monday  | Normal Salary   |
| 2            | 50           | Tuesday | 1.25 of salary  |
| 3            | 52           | Sunday  | 2.00 of salary  |

# Expanding the Immaterial Cases in Decision Table

- These conditions means that the value of a particular condition in the specific rule does not make a difference whether it is TRUE or FALSE.
- Sometimes expanding the decision table to spell out don't-care conditions can reveal hidden problems.



# Example - I

## Entry (Decision table)

|                       |                         | Rule 1 | Rule 2 | Rule 3 |
|-----------------------|-------------------------|--------|--------|--------|
| <b>Condition Stub</b> | C1: Working hours > 48  | I      | F      | T      |
|                       | C2: Holidays or Sundays | T      | F      | F      |
| <b>Action Stub</b>    | A1: Normal Salary       |        | X      |        |
|                       | A2: 1.25 of salary      |        |        | X      |
|                       | A3: 2.00 of salary      | X      |        |        |

- The immaterial test case in rule I of the above table can be expanded by taking both T and F values of C1.

# Entry (Expanded decision table)

|                       |   | Rule 1-1  | Rule 1-2  | Rule 2 | Rule 3 |
|-----------------------|---|-----------|-----------|--------|--------|
| <b>Condition Stub</b> | C1: Working hours > 48<br>C2: Holidays or Sundays             | F<br>T    | T<br>T    | F<br>F | T<br>F |
| <b>Action Stub</b>    | A1: Normal Salary<br>A2: 1.25 of salary<br>A3: 2.00 of salary | <br><br>X | <br><br>X | X      | <br>X  |

## Entry (Expanded decision table)

| Test case ID | Working hour | Day     | Expected Result |
|--------------|--------------|---------|-----------------|
| 1            | 48           | Monday  | Normal Salary   |
| 2            | 50           | Tuesday | 1.25 of salary  |
| 3            | 52           | Sunday  | 2.00 of salary  |
| 4            | 30           | Sunday  | 2.00 of salary  |

# Example - 2

A wholesaler has three commodities to sell and has three types of customers. Discount is given as per the following procedure:

- (a) For DGS & D orders, 10% discount is given irrespective of the value of the order.
- (b) For orders of more than Rs 50,000, agents get a discount of 15% and the retailer gets a discount of 10%.
- (c) For orders of Rs 20,000 or more and up to Rs 50,000, agents get 12% and the retailer gets 8% discount.
- (d) For orders of less than Rs 20,000, agents get 8% and the retailer gets 5% discount.

The aforementioned rules do not apply to the furniture items wherein a flat rate of 10% discount is admissible to all customers irrespective of the value of the order.

Design the test cases for this system using decision table testing.

# Decision Table for Example 2

|                |                                    | ENTRY |    |    |    |    |    |    |    |
|----------------|------------------------------------|-------|----|----|----|----|----|----|----|
|                |                                    | R1    | R2 | R3 | R4 | R5 | R6 | R7 | R8 |
| Condition Stub | C1: DGS & D                        | T     | F  | F  | F  | F  | F  | F  | F  |
|                | C2: Agent                          | F     | T  | F  | T  | F  | T  | F  | I  |
|                | C3: Retailer                       | F     | F  | T  | F  | T  | F  | T  | I  |
|                | C4: Order > 50,000                 | I     | T  | T  | F  | F  | F  | F  | I  |
|                | C5: Order $\geq 20000$ to < 50,000 | I     | F  | F  | T  | T  | F  | F  | I  |
|                | C6: Order < 20,000                 | I     | F  | F  | F  | F  | T  | T  | I  |
|                | C7: Furniture                      | F     | F  | F  | F  | F  | F  | F  | T  |
| Action Stub    | A1: Discount of 5%                 |       |    |    |    |    |    | X  |    |
|                | A2: Discount of 8%                 |       |    |    |    | X  | X  |    |    |
|                | A3: Discount of 10%                | X     |    | X  |    |    |    |    | X  |
|                | A4: Discount of 12%                |       |    |    | X  |    |    |    |    |
|                | A5: Discount of 15%                |       | X  |    |    |    |    |    |    |

# Test cases for Example - 2

| <b>Test Case Id</b> | <b>Type of Customer</b> | <b>Product Furniture?</b> | <b>Order value (Rs)</b> | <b>Expected Result</b> |
|---------------------|-------------------------|---------------------------|-------------------------|------------------------|
| 1                   | DGS & D                 | No                        | 51,000                  | 10% Discount           |
| 2                   | Agent                   | No                        | 52,000                  | 15% Discount           |
| 3                   | Retailer                | No                        | 53,000                  | 10% Discount           |
| 4                   | Agent                   | No                        | 23,000                  | 12% Discount           |
| 5                   | Retailer                | No                        | 27,000                  | 8% Discount            |
| 6                   | Agent                   | No                        | 15,000                  | 8% Discount            |
| 7                   | Retailer                | No                        | 18,000                  | 5% Discount            |
| 8                   | Agent                   | Yes                       | 34,000                  | 10% Discount           |



**Thank You**