Specification Based Testing – Part 1

State Based Testing

Ira A. Fulton Schools of Engineering
Arizona State University

Objective

Objective

Apply state based testing technique

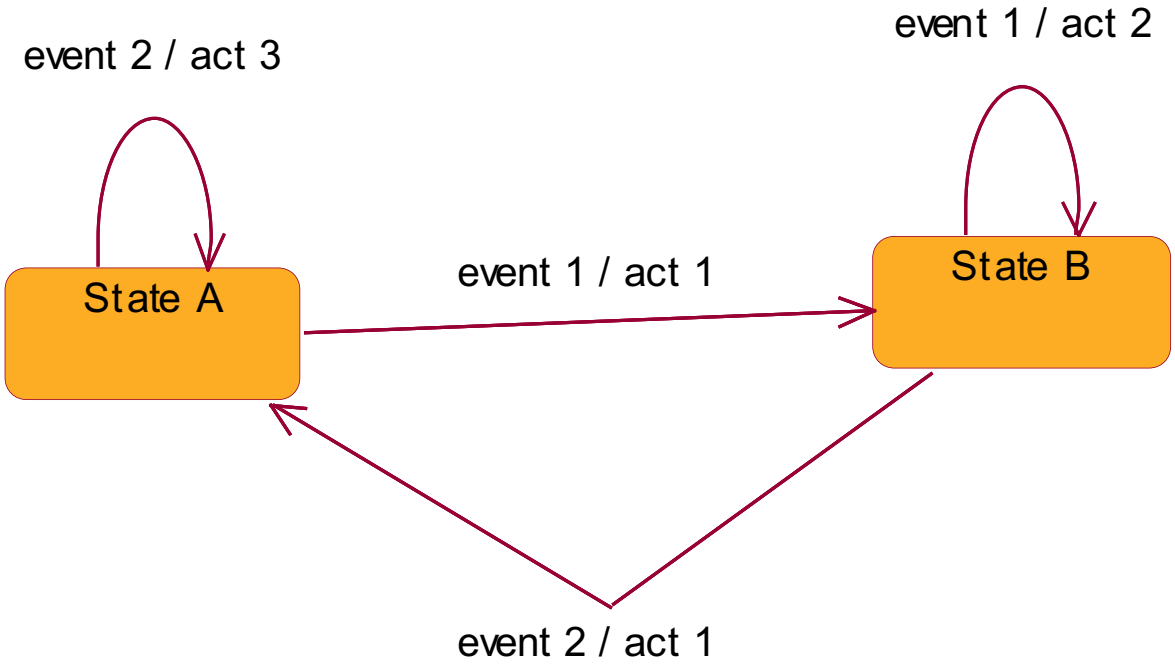
State Machines

- System behavior can sometimes best be captured in the form of a state machine
- State machine consists of a set of states and events
- When an event occurs in a particular state, it triggers a state transition and a response

State Machine Table Representation

	State A	State B
Event 1	State B / act 1	State B / act 2
Event 2	State A / act 3	State A / act 1

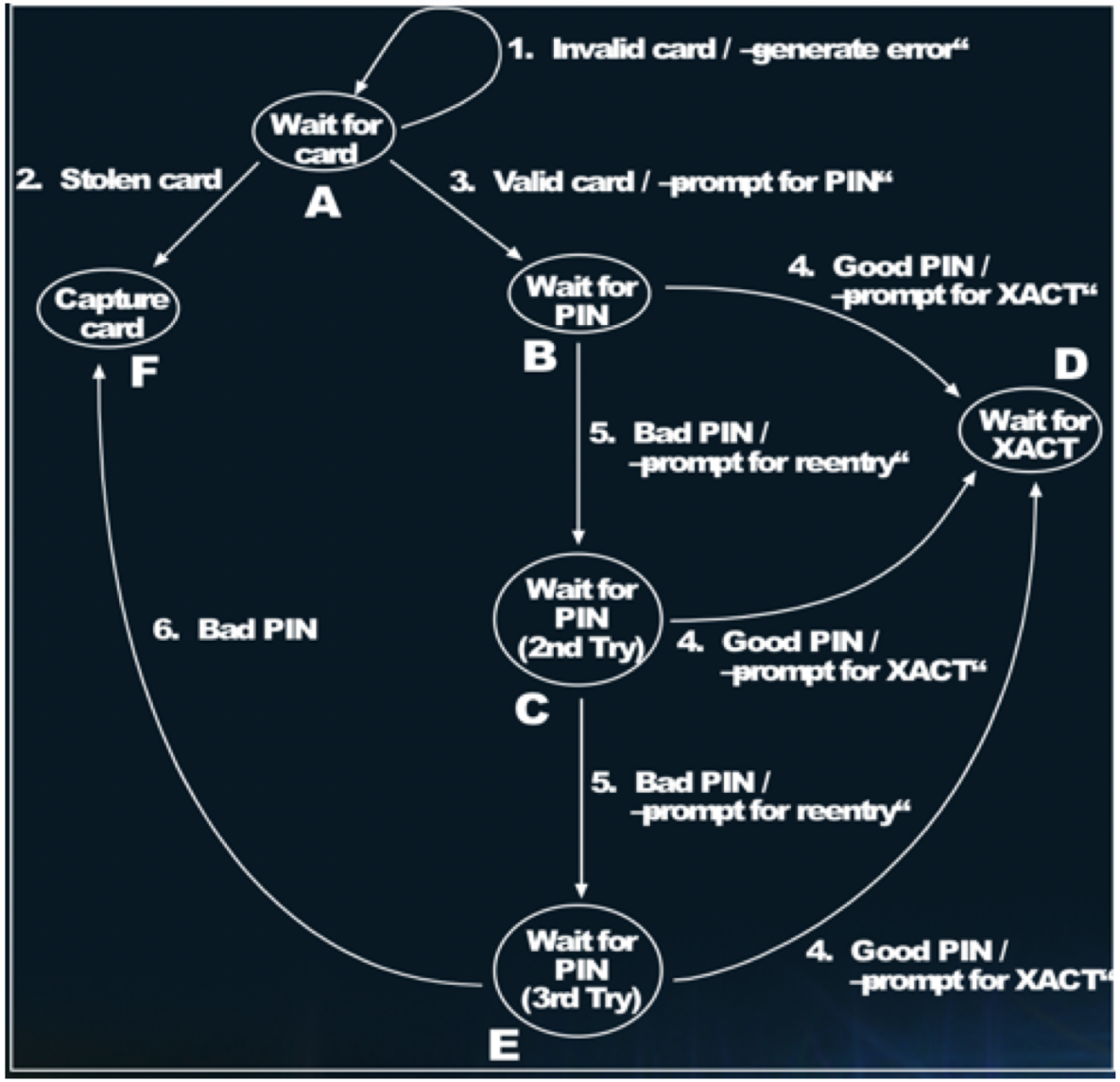
State Machine Diagram



Example State Diagram

- Consider the use-case for validating a user in an ATM system

Example State Machine



State Based Testing

State machines must be inspected for:

Completeness

- every state / event pair must be identified
- verify conditional transitions are correct

Contradiction

- 2 transitions from the same state should not contain the same event
- danger occurs with nested state charts

State Based Testing

State machines must be inspected for:

Unreachable States

- can be complicated with nested state machines and the use of conditional transitions

Dead States

- states that can be entered but not exited

Testing Finite State Diagrams

A testing cover can be developed for a Finite State Diagram by the following steps:

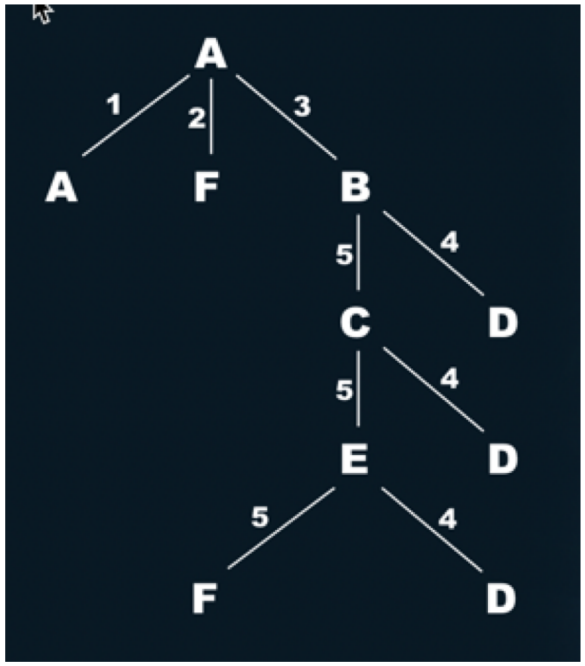
1. Develop a state testing tree
2. Identify test sequences (paths through the tree)
3. Develop tests to contain the sequences starting with the Start state and ending with observable behavior

Developing a State Testing Tree

- Begin with start state as root of tree
- At the first level, identify each of the transitions from the start state and the new states reached
- Continue expanding the tree downward for each state which has not previously occurred in the tree at a higher level

State Testing Tree

Testing Sequences	Tests
1	1
2	2
3, 4	3, 4
3, 5, 4	3, 5, 4
3, 5, 5	3, 5, 5
3, 5, 5, 4	3, 5, 5, 4



Tests map exactly to testing sequences since each sequence generates observable behavior.

Summary