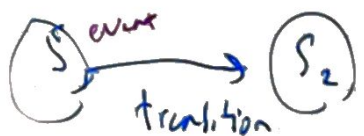


State  
transitions  
events cause transitions

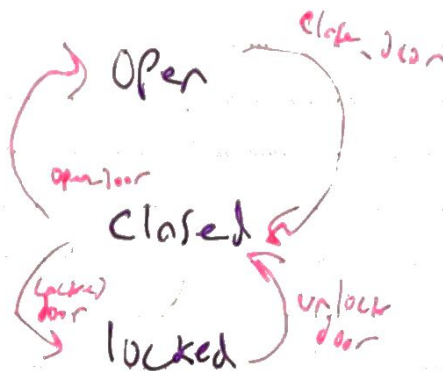
Conditions  
predicate  
consumer

generic, lightweight FSM

functional programming



optional actions  
has failed transition



Open  
locked  
close

Open → close, close door, List of ( () → log, lock door, )  
Closed → lock, lock door

Make  
transition  
table

Start(Open)

fsm. close door → event allows, return state door-closed

fsm. close door → throw illegal state exception

valid transitions are ! - - -

fsm. lock door → event allows, return locked door

fsm.link(source\_event). event → target  
→ action

fsm.linkFrom(source\_event). <sup>State</sup>  
→ ~~event~~  
→ action  
→ target (State)

fsm < Event, State >

fsm.initialState ∈ State

fsm.performEvent(event)

