Behavioral modeling with UML state diagrams

Professor Hossein Saiedian

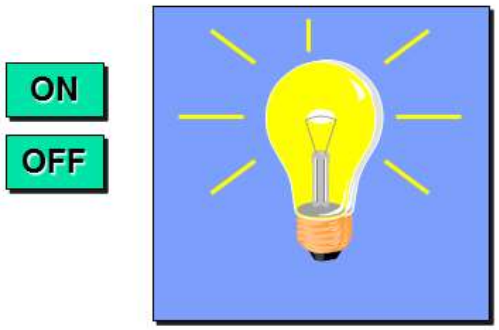
EECS 348: Software Engineering

Spring 2023

THE UNIVERSITY OF KANSAS

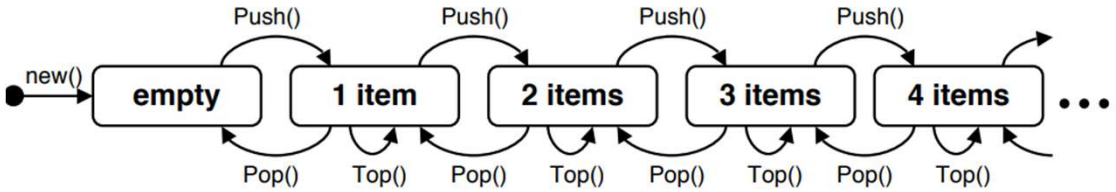
State machines background

- Automata: a machine whose output behavior is not only a direct consequence of the current input, but of some past history of its inputs
  - The internal state represents the past experience
  - Useful for describing *events driven (concrete) behavior*



Terminology

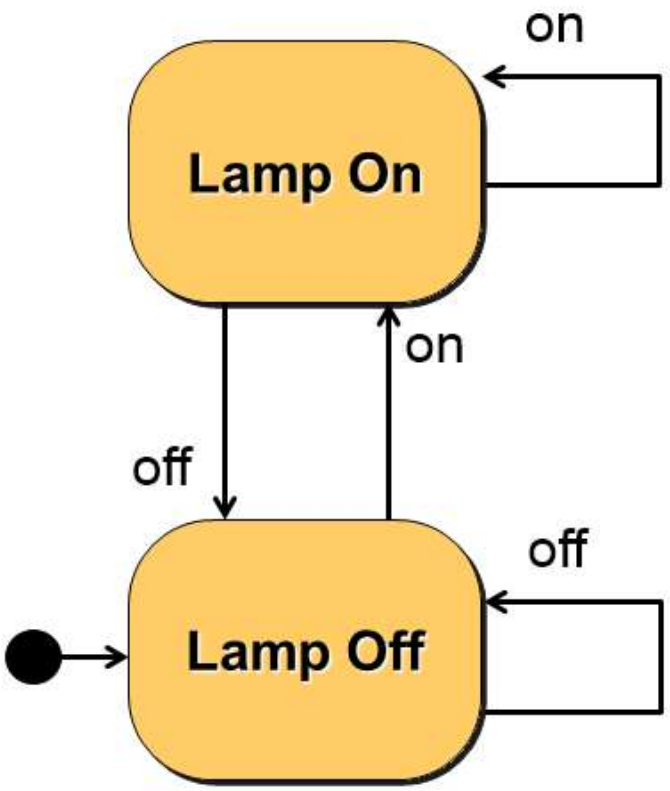
- State
  - All objects have a state
  - Statespace for an object
  - State of stack:



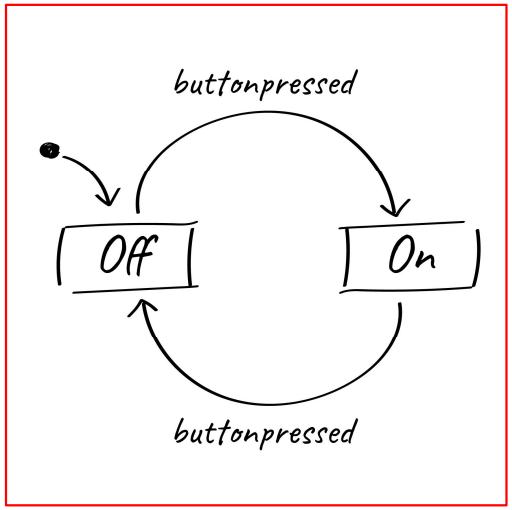
- State transition modeling: modeling the states of an object
  - State modeling or modeling the state
  - Statechart modeling
  - Finite state modeling
  - State machines

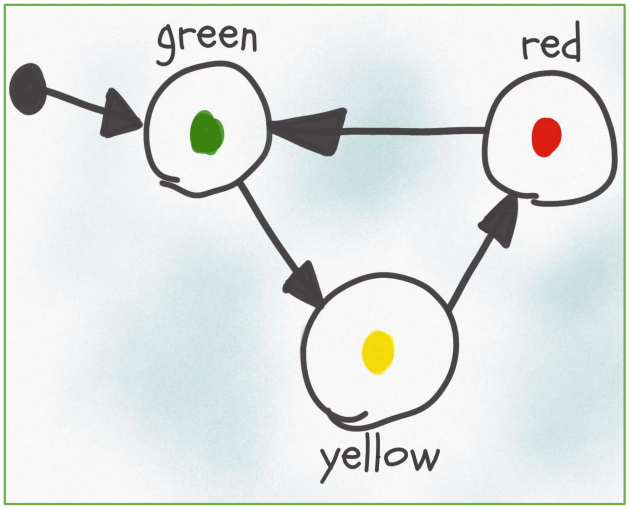
Common state machine diagram

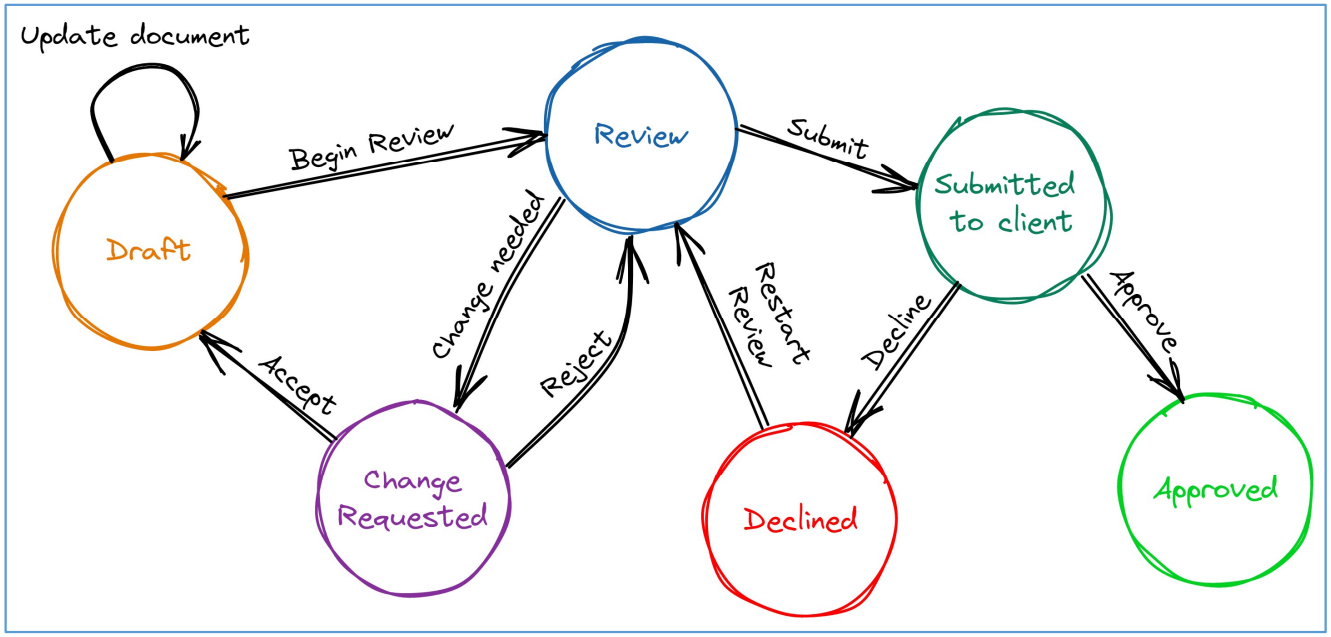
- Graphical representation of the lamp behavior



Informal, FSM notation

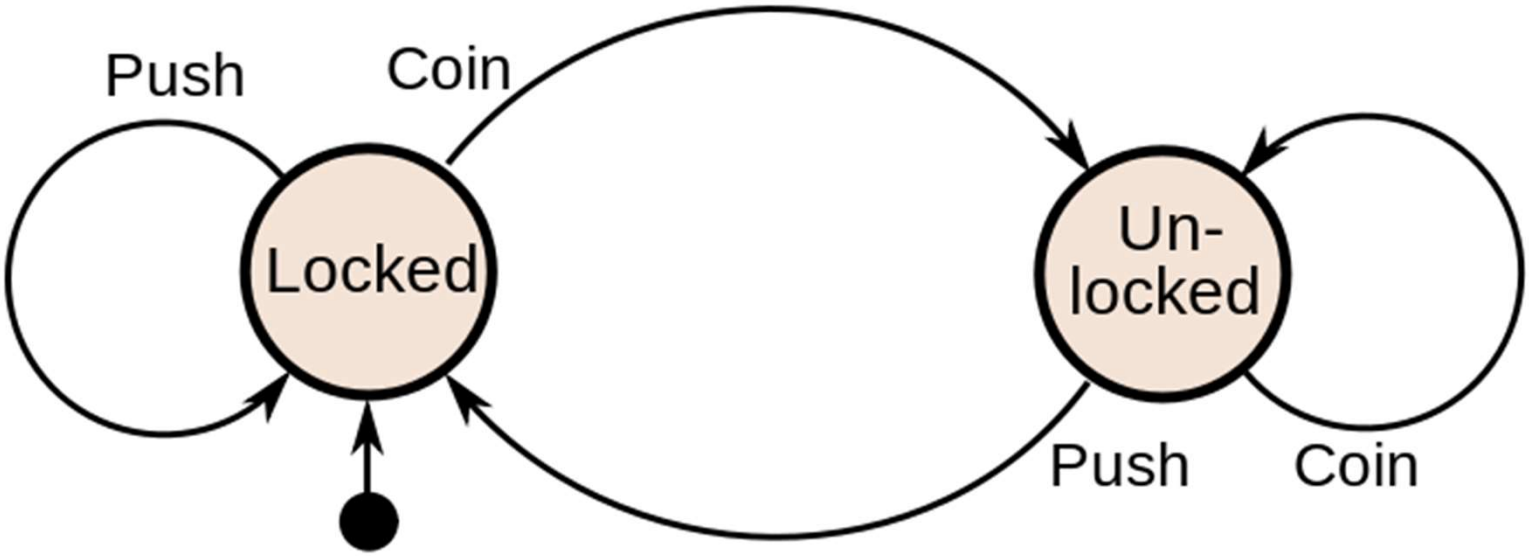




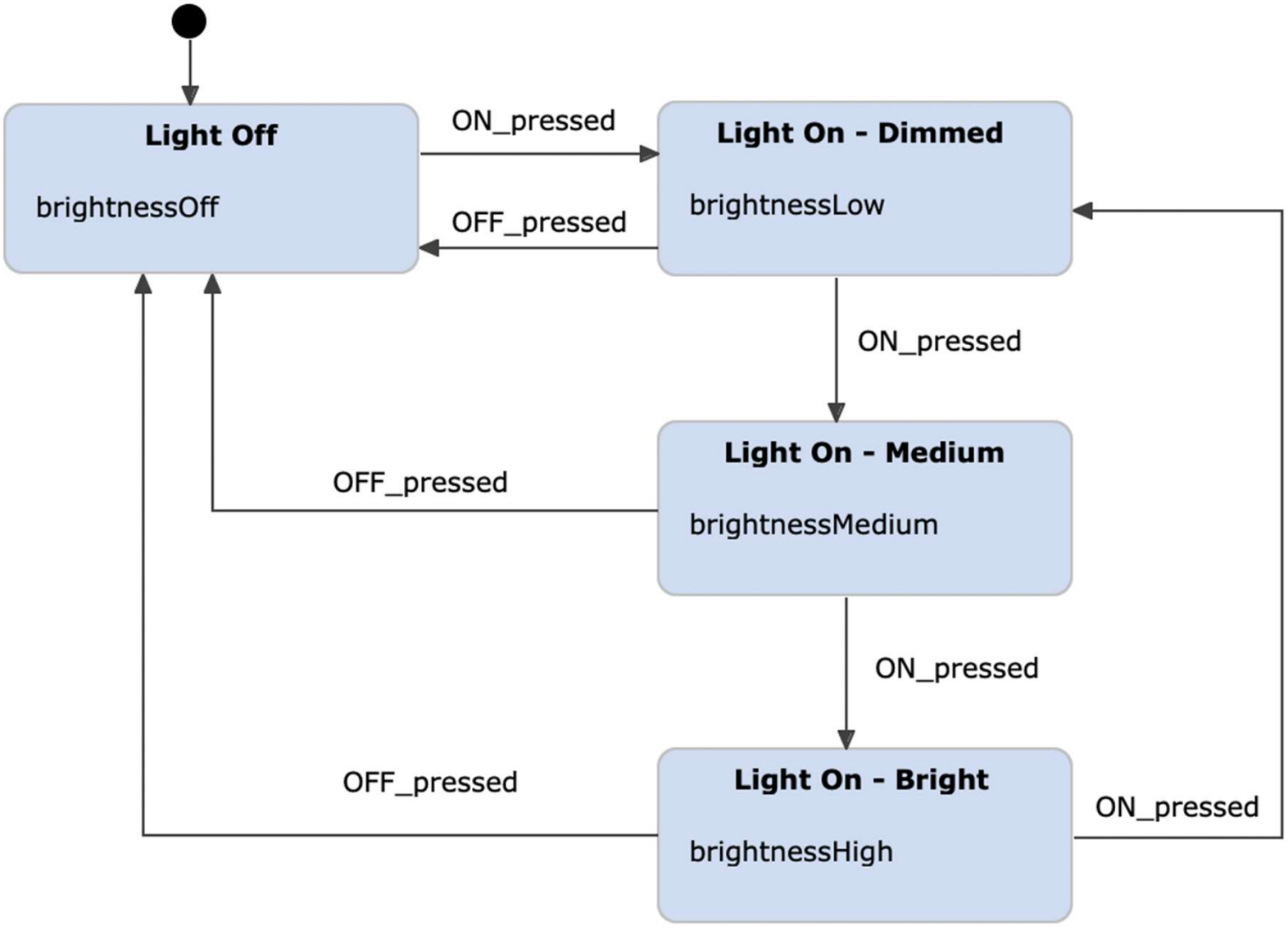


Another FSM example

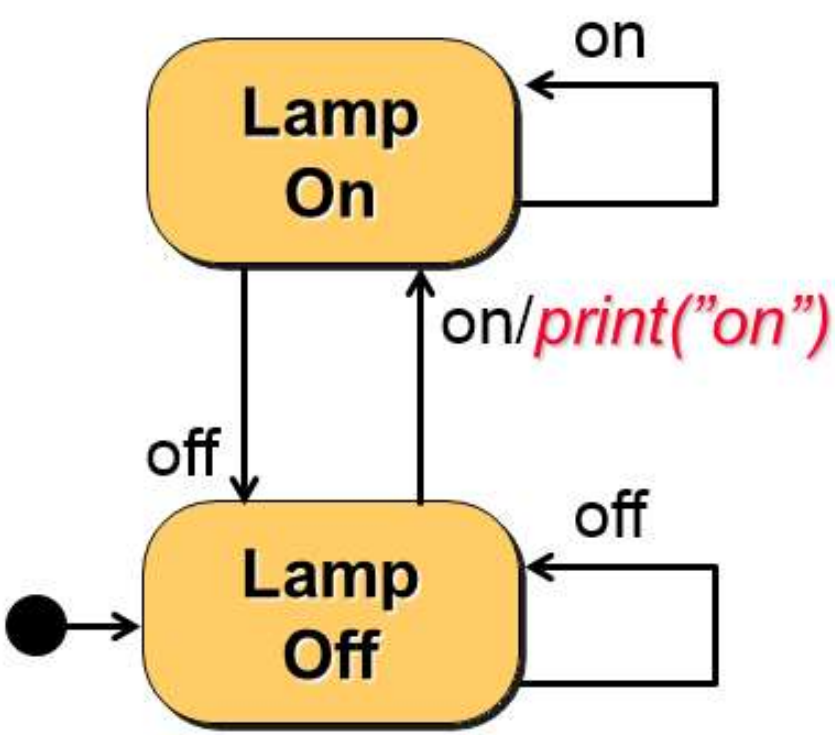
- A turnstile state machine



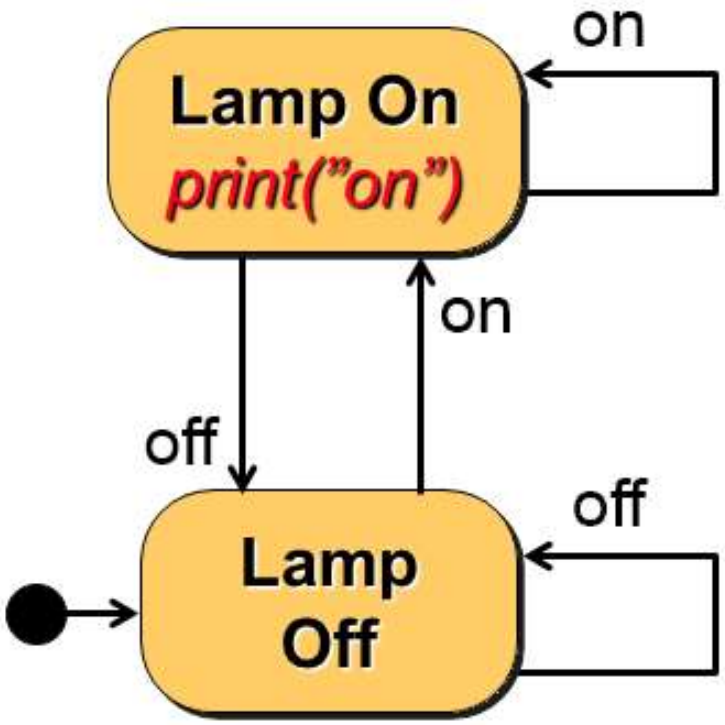
Another example: lamp with more states



State transition can generate output

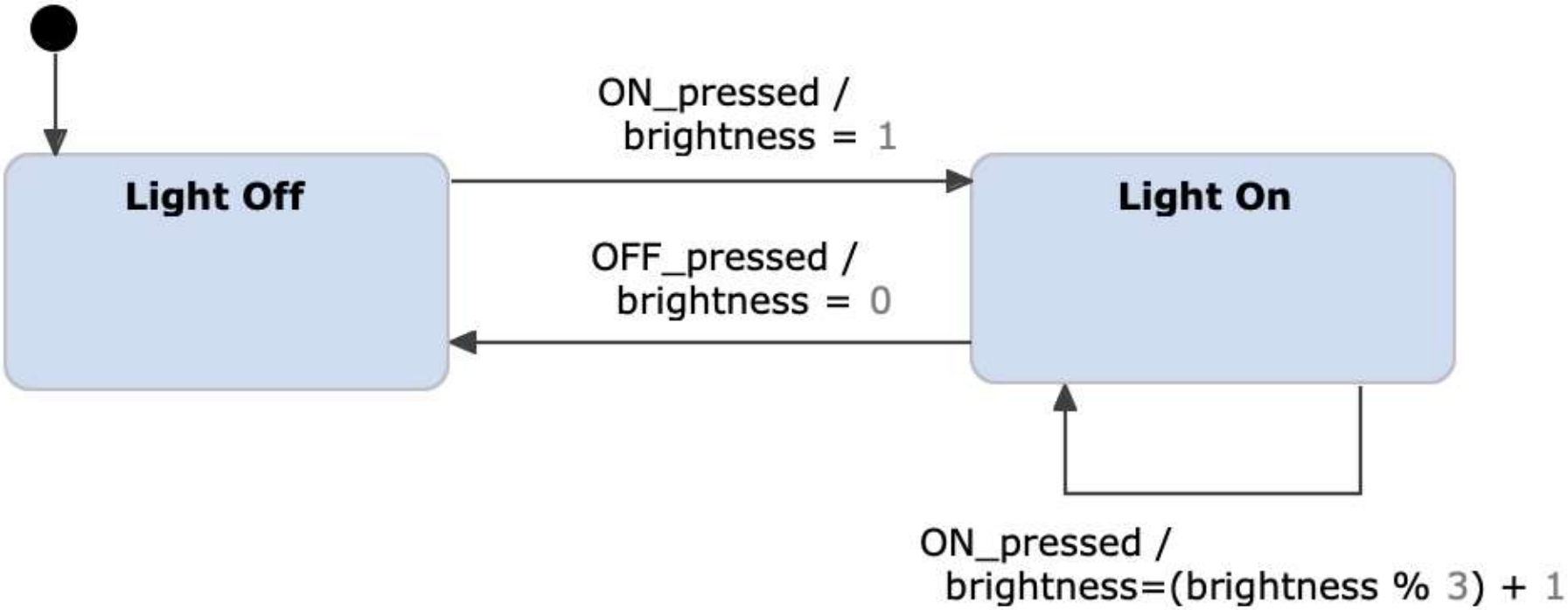


Mealy automaton

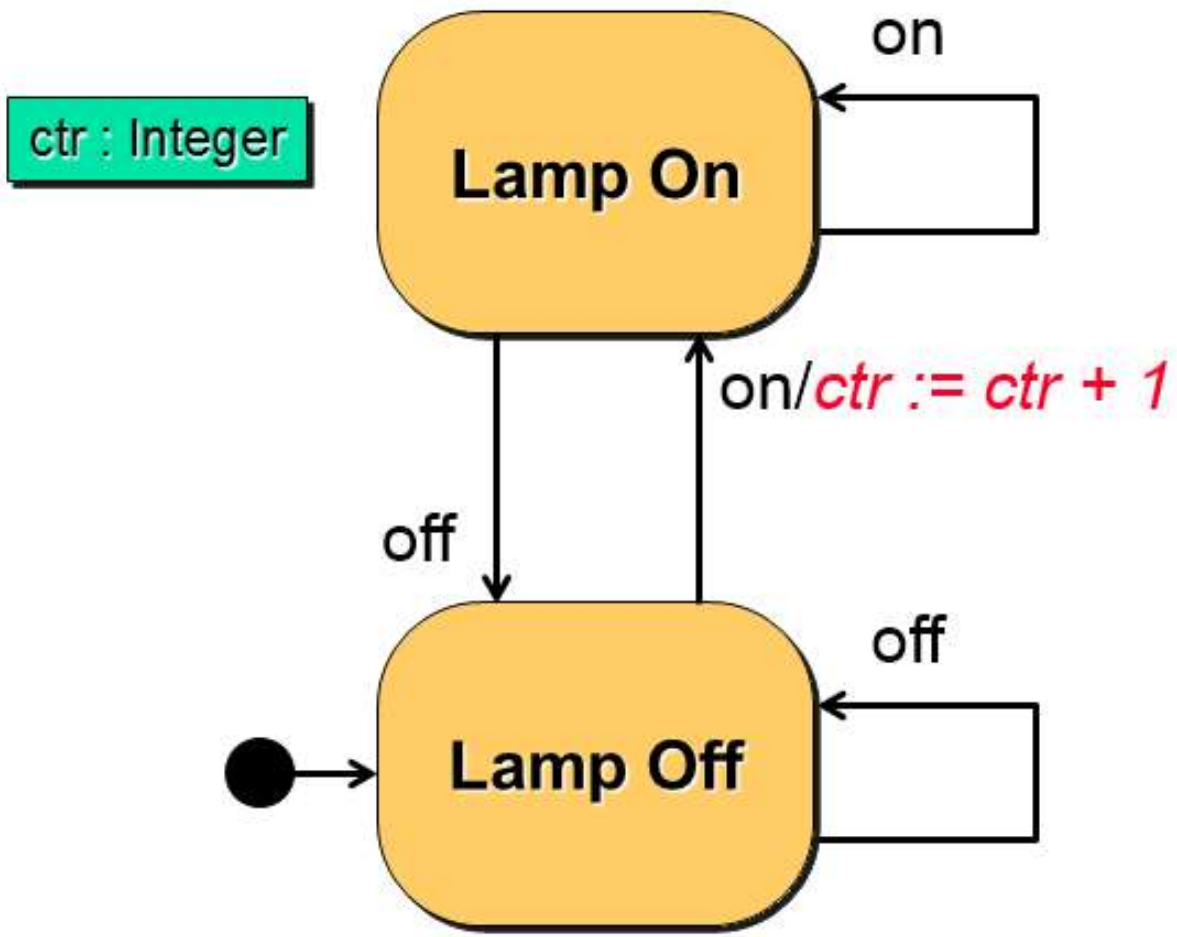


Moore automaton

Another example

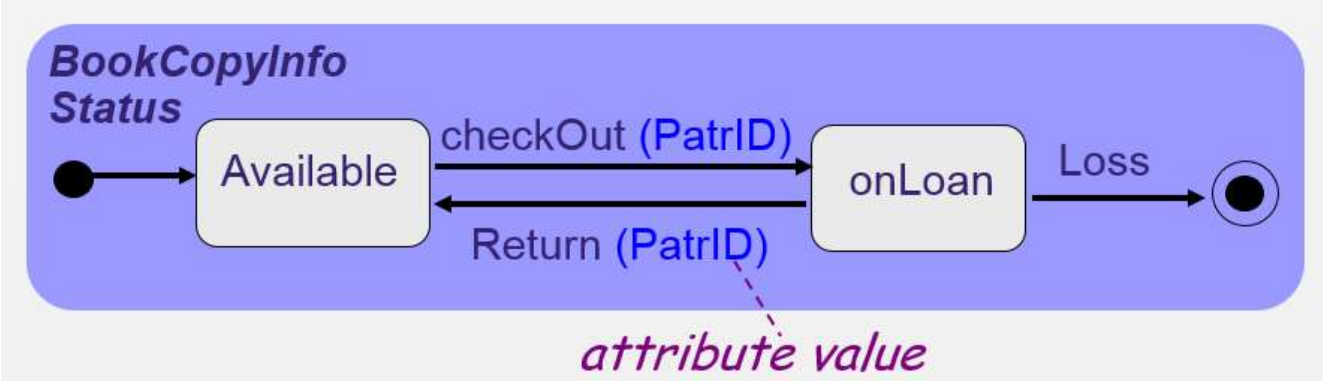


The state can have variables



A variety of states can be modeled

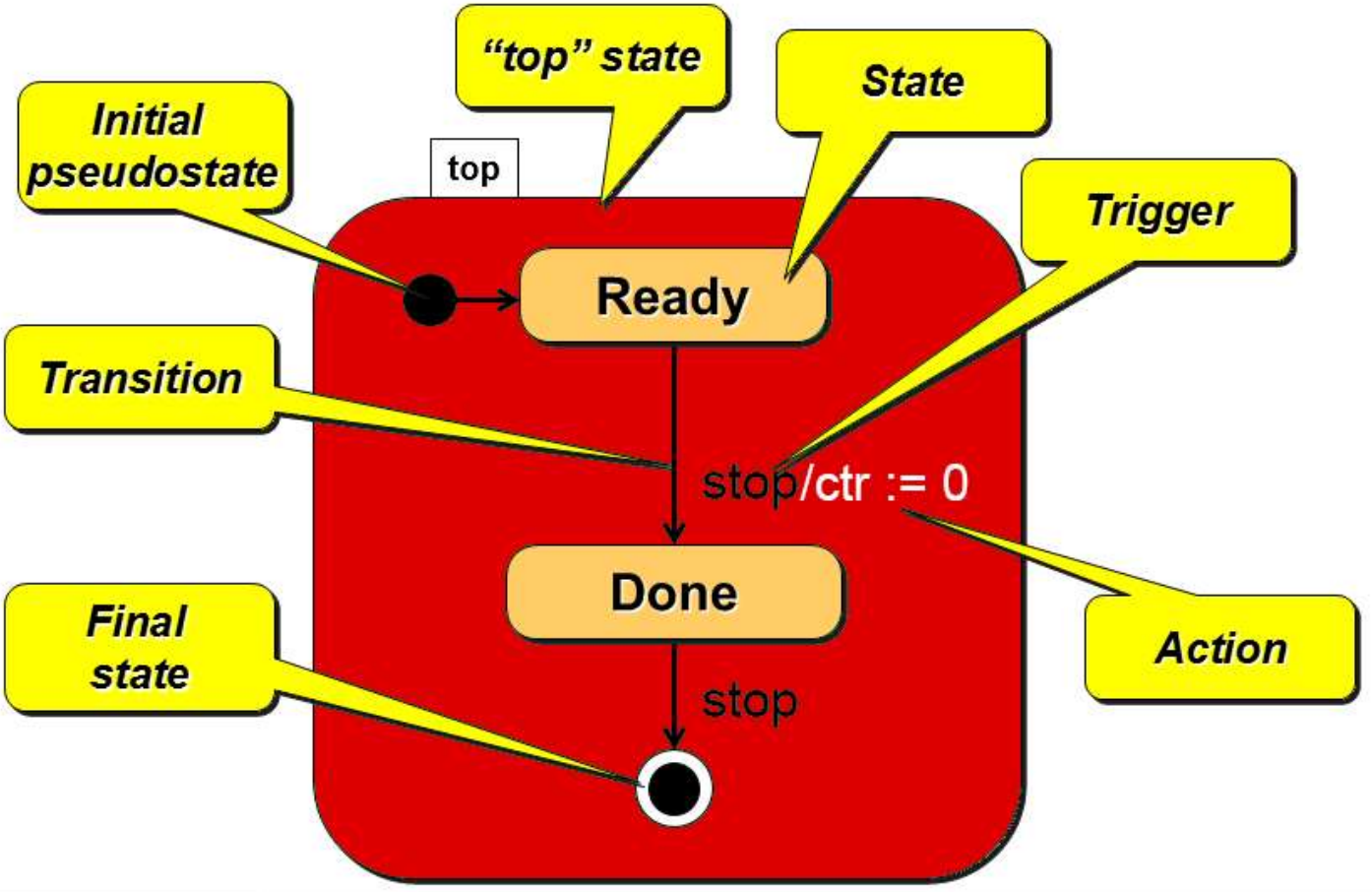
- A digital watch, a microwave system, an engine, a cruise control, and ATM machine
- A software module that does lexical analysis
- Normally used for objects with complex behaviors
- But OK to use for simpler concepts (a library)



An informal definition of a state machine

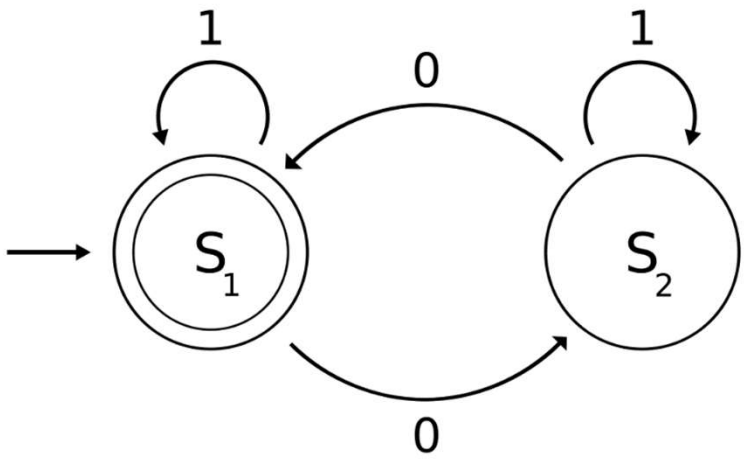
- A set of input signals (input alphabet)
- A set of output signals (output alphabet)
- A set of states
- A set of transitions
  - triggering signal
  - action
- A set of extended state variables
- An initial state designation
- A set of final states (if terminating automaton)

UML's (simplified) statechart notation



For the mathematically intrigued

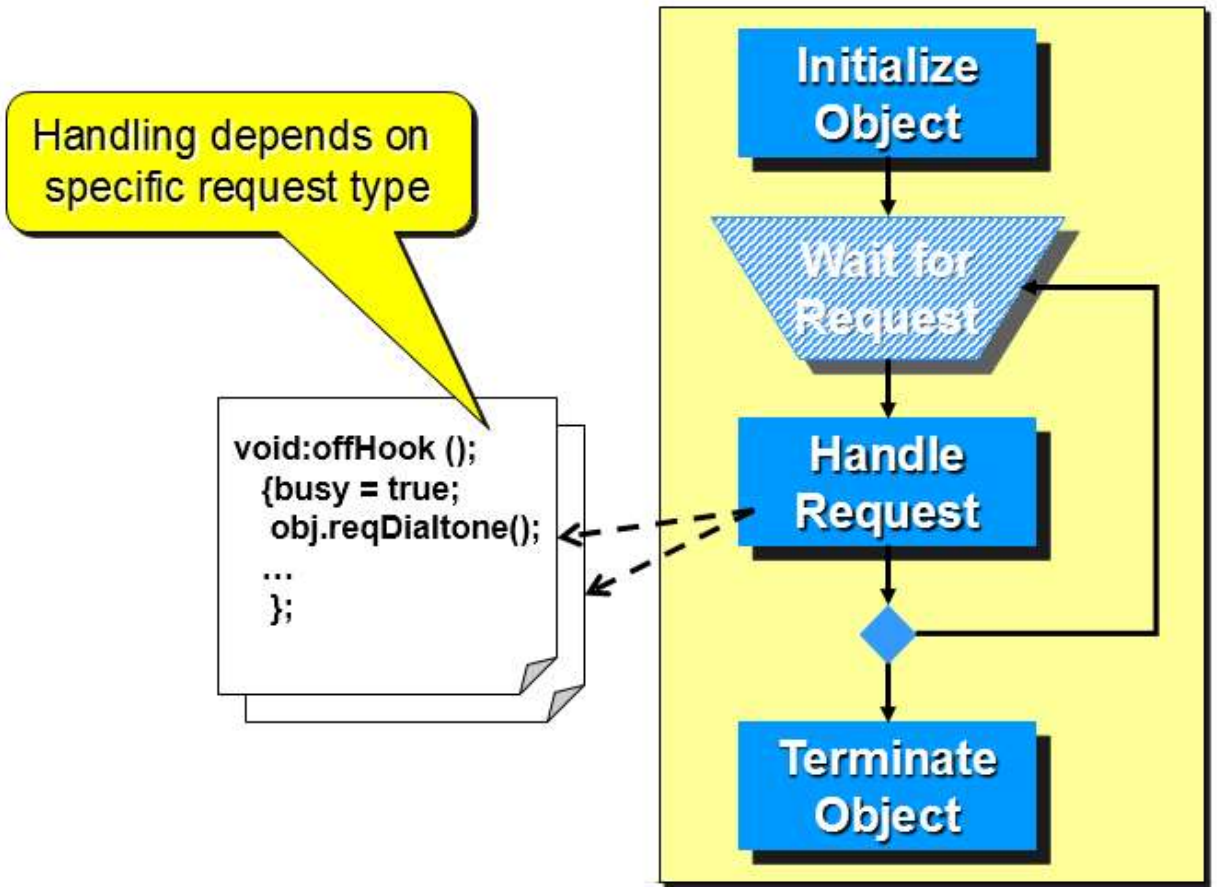
- An automata is a 5-tuple: $\langle S, \Sigma, \delta, S_0 \rangle$
  - S is a set of states
  - Σ is an alphabet; an finite set of symbols
  - δ is the transition function: $\delta : S \times \Sigma \rightarrow S$
  - S_0 is an initial state



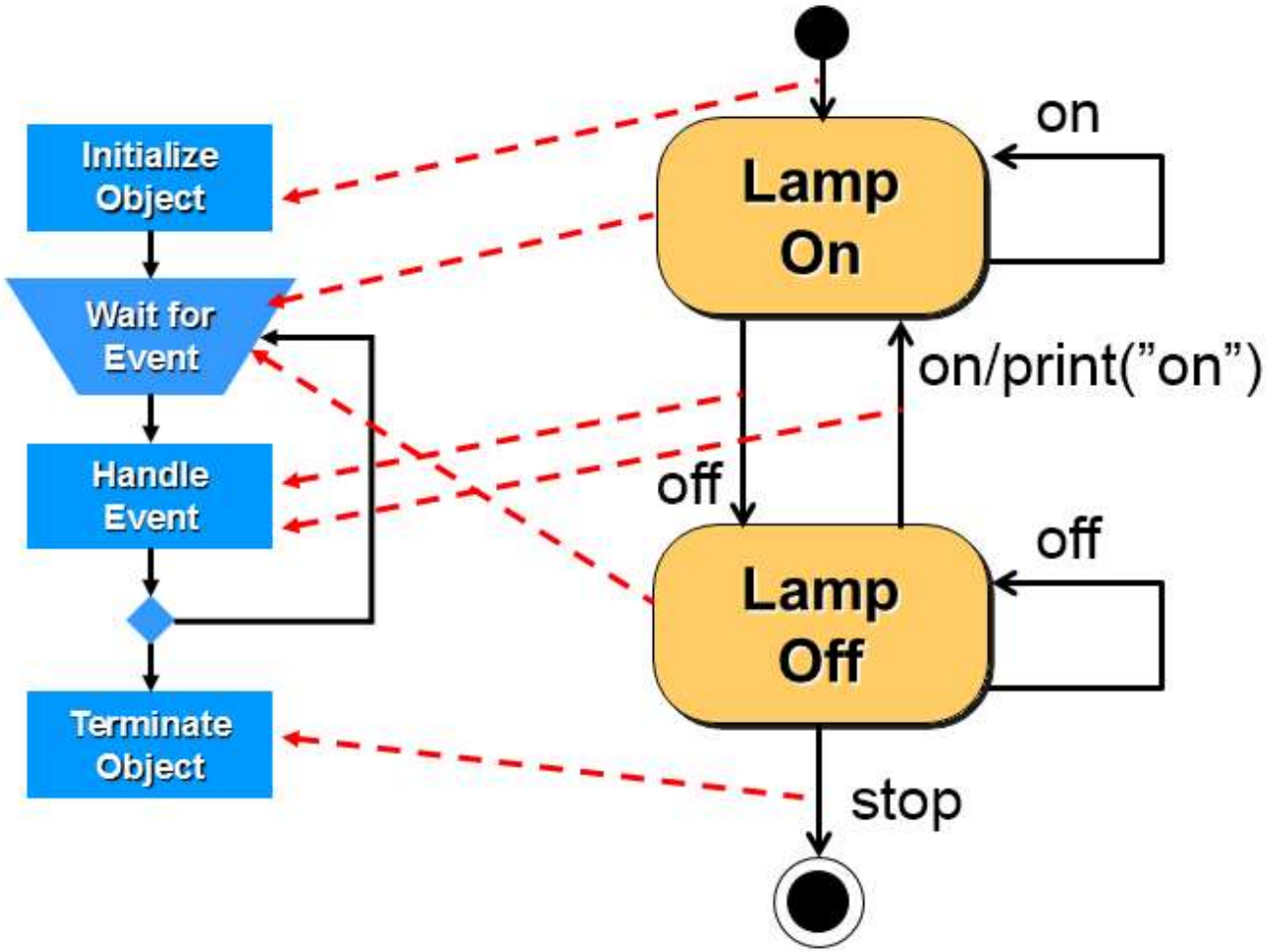
Event-driven behavior

- Even-driven behavior in practice: the behavior of individual objects
- An event: an observable occurrence
  - Object method (call) invocation
  - Asynchronous signal reception
  - Calendar/clock time
  - Change in value of some property

Modeling an object behavior

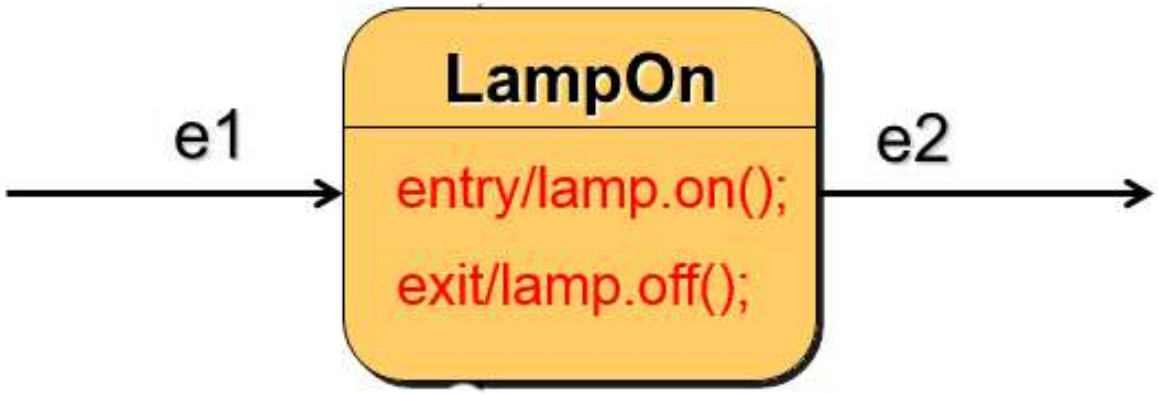


Object behavior as state machine

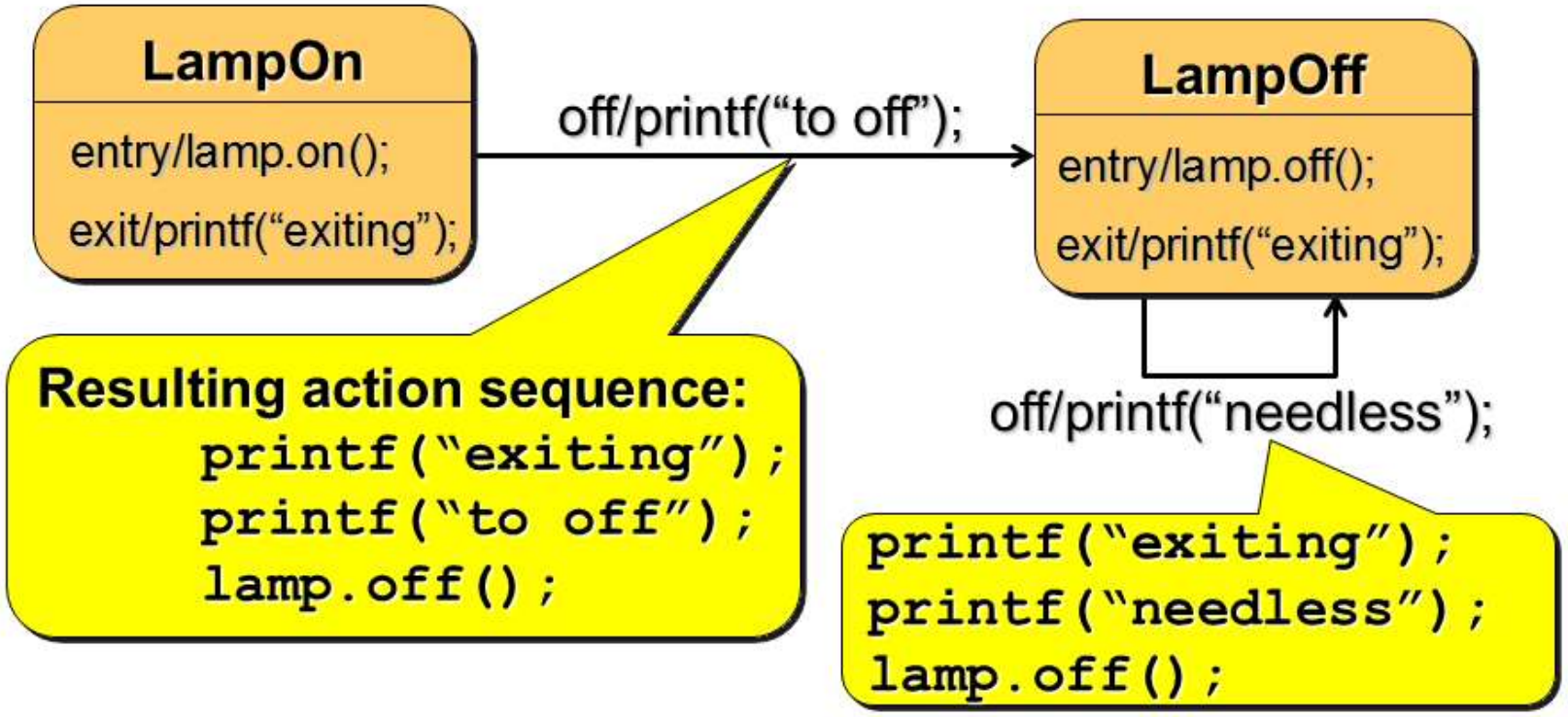


Entry and exist actions

- Entry action postfix transition actions
- Exit actions prefix transition actions

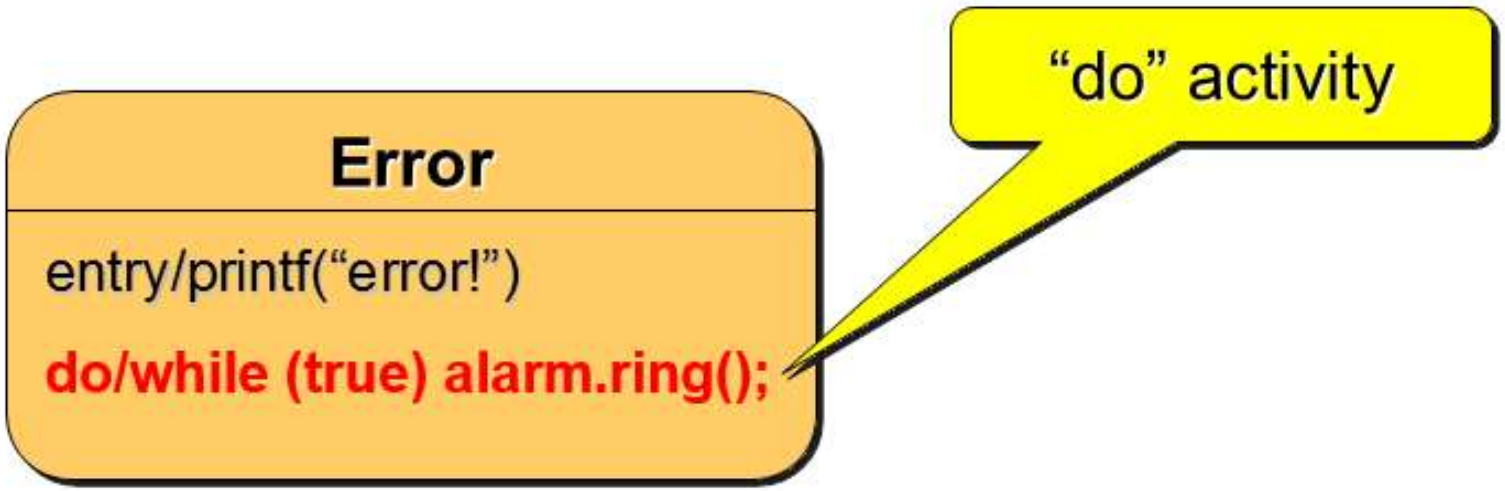


Order of actions

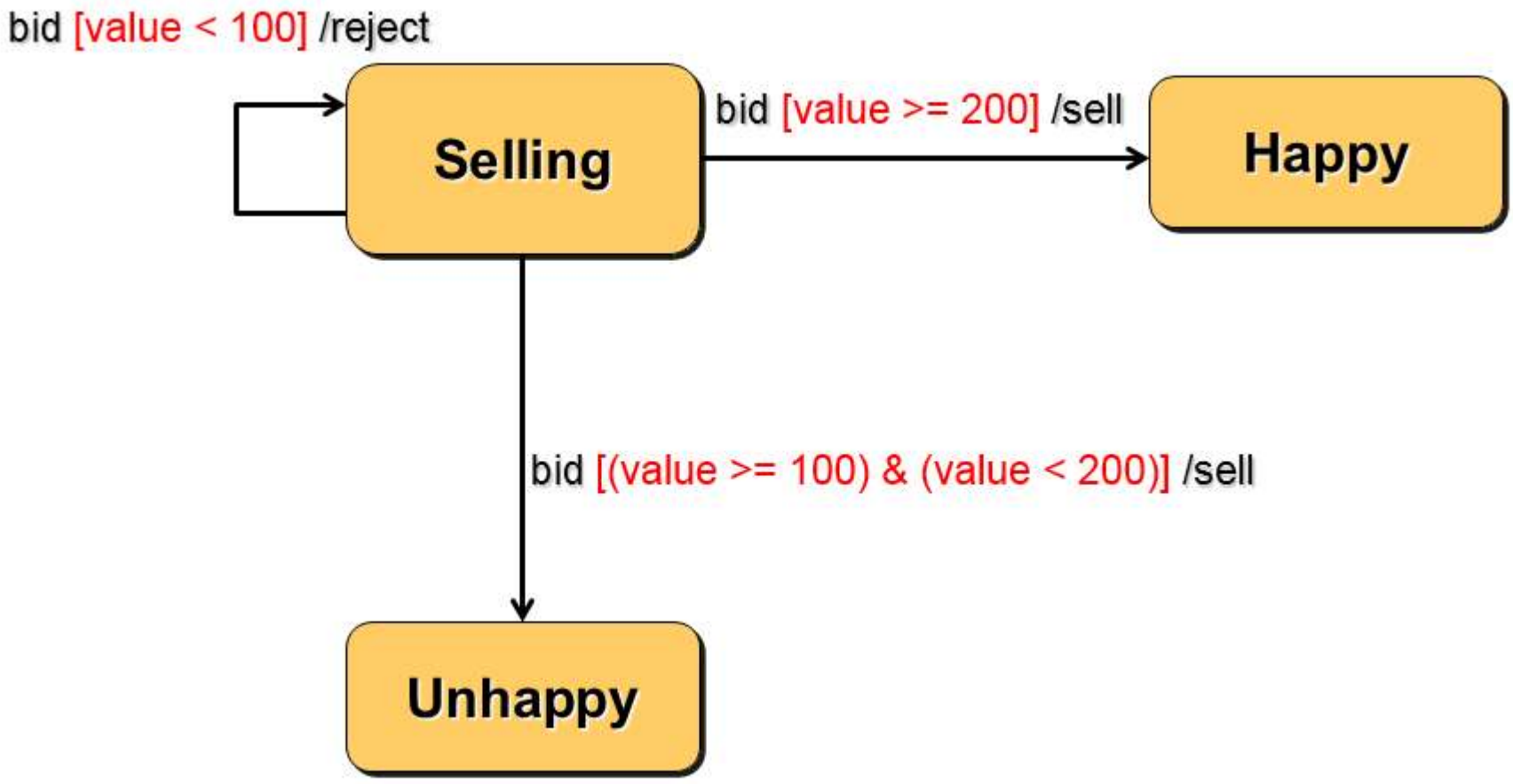


State "do" activities

- Forks a concurrent thread that executes until
  - The action completes
  - The state is exited via an outgoing transition

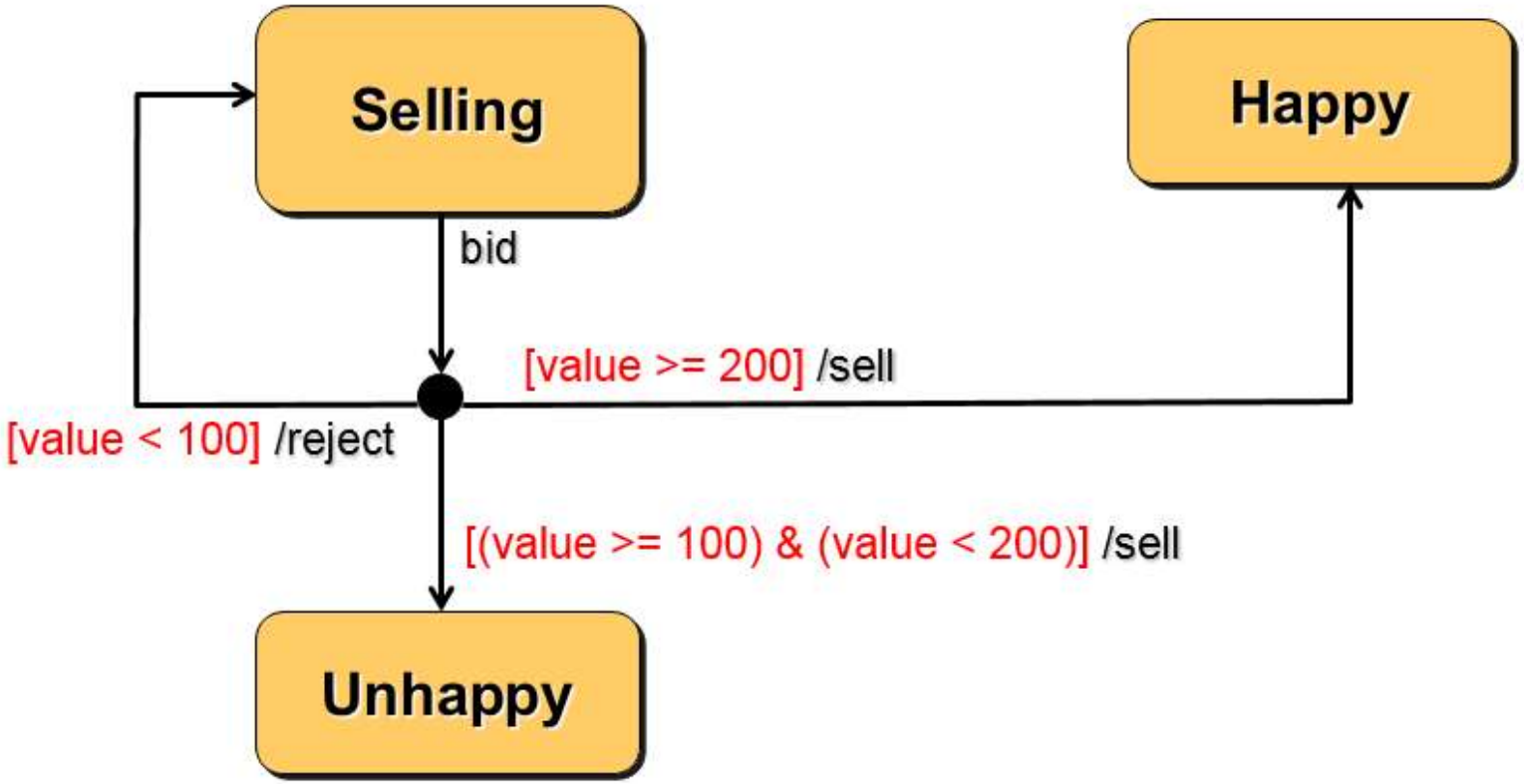


Transitions can have a guard



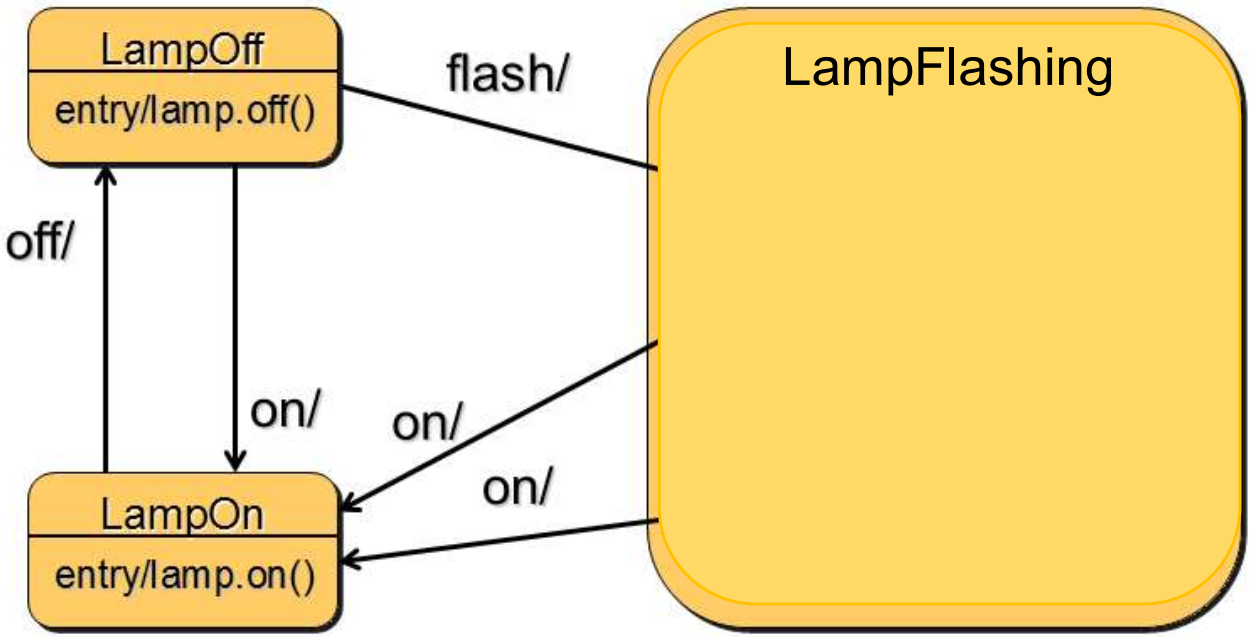
Conditional branching

- A graphical (convenient) shortcut



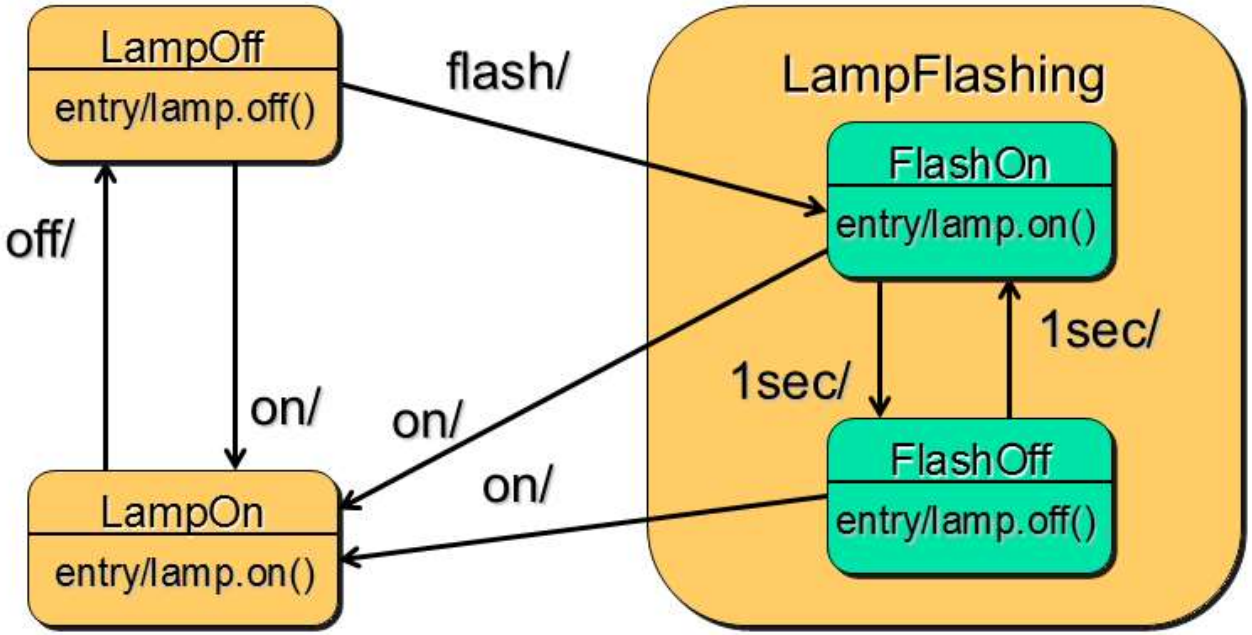
Super (or compound) states

- States composed into state machines (to manage complexity)



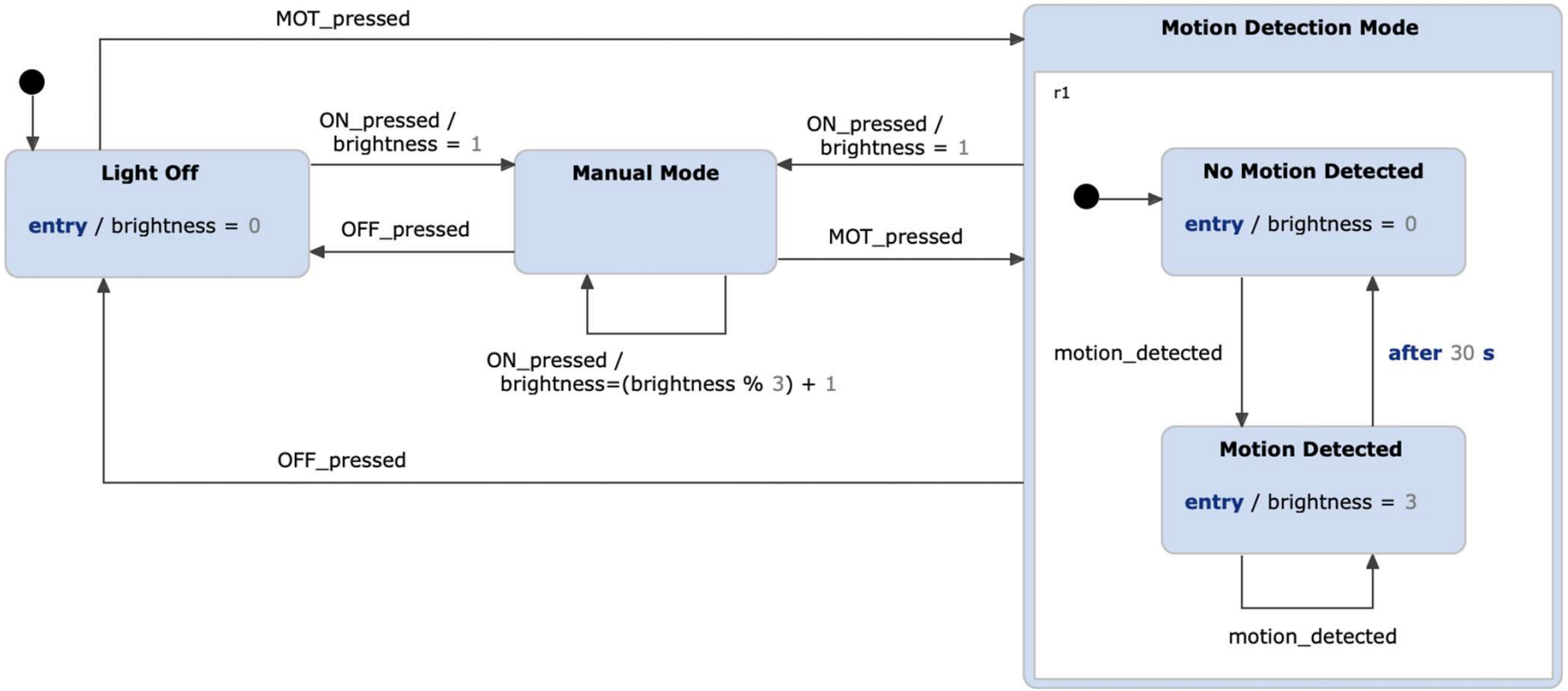
Super (or compound) states

- States composed into state machines (to manage complexity)



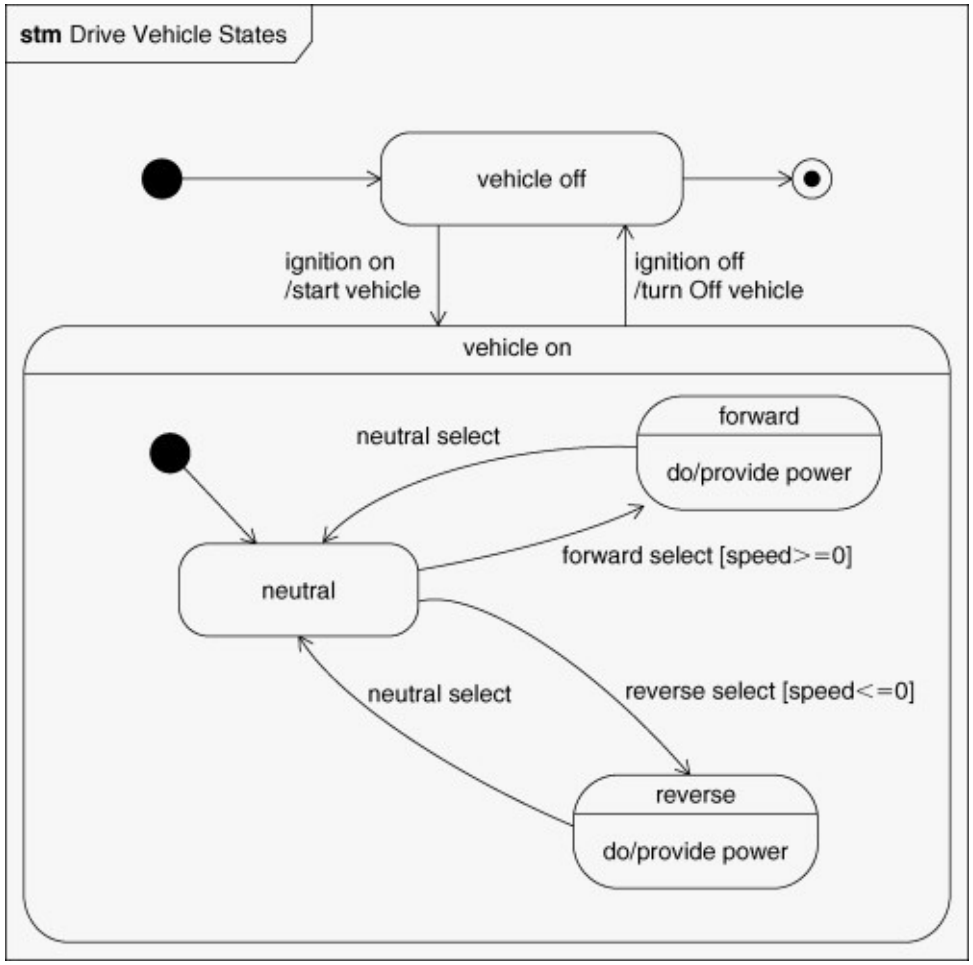
Super states: another example

- States composed into state machines (to manage complexity)

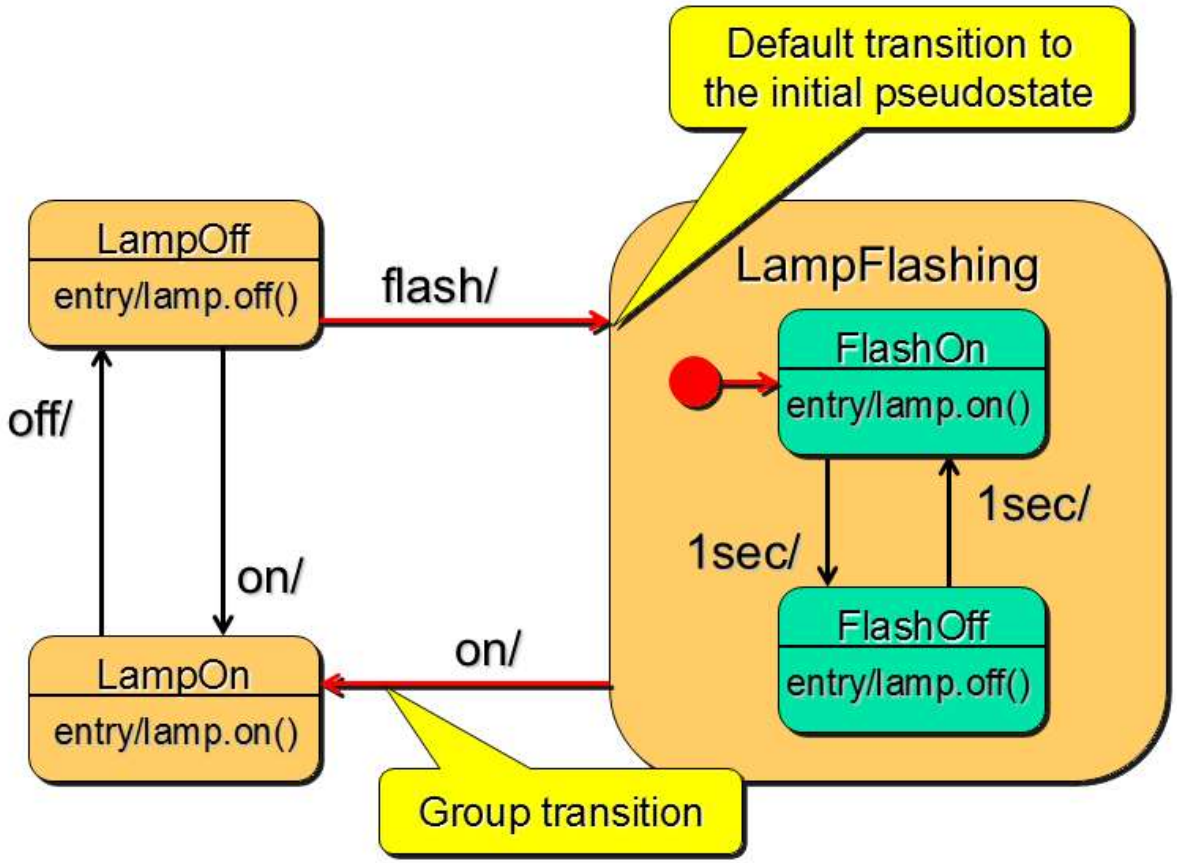


Super states: another example

- A vehicle drive states

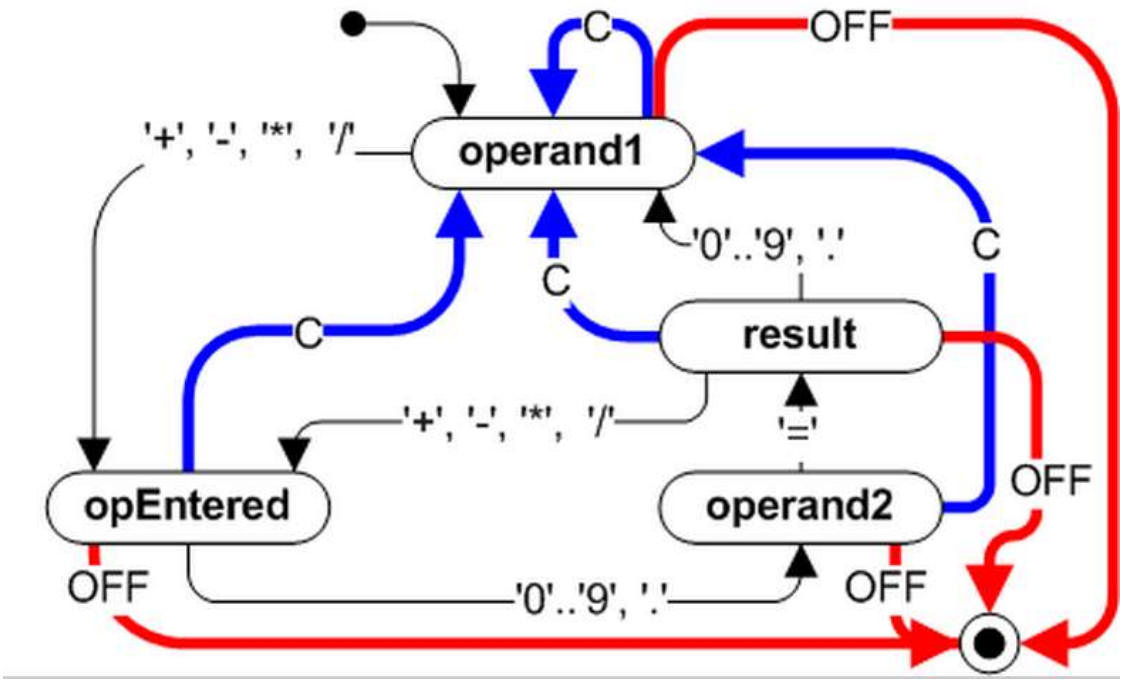


Group transitions are possible



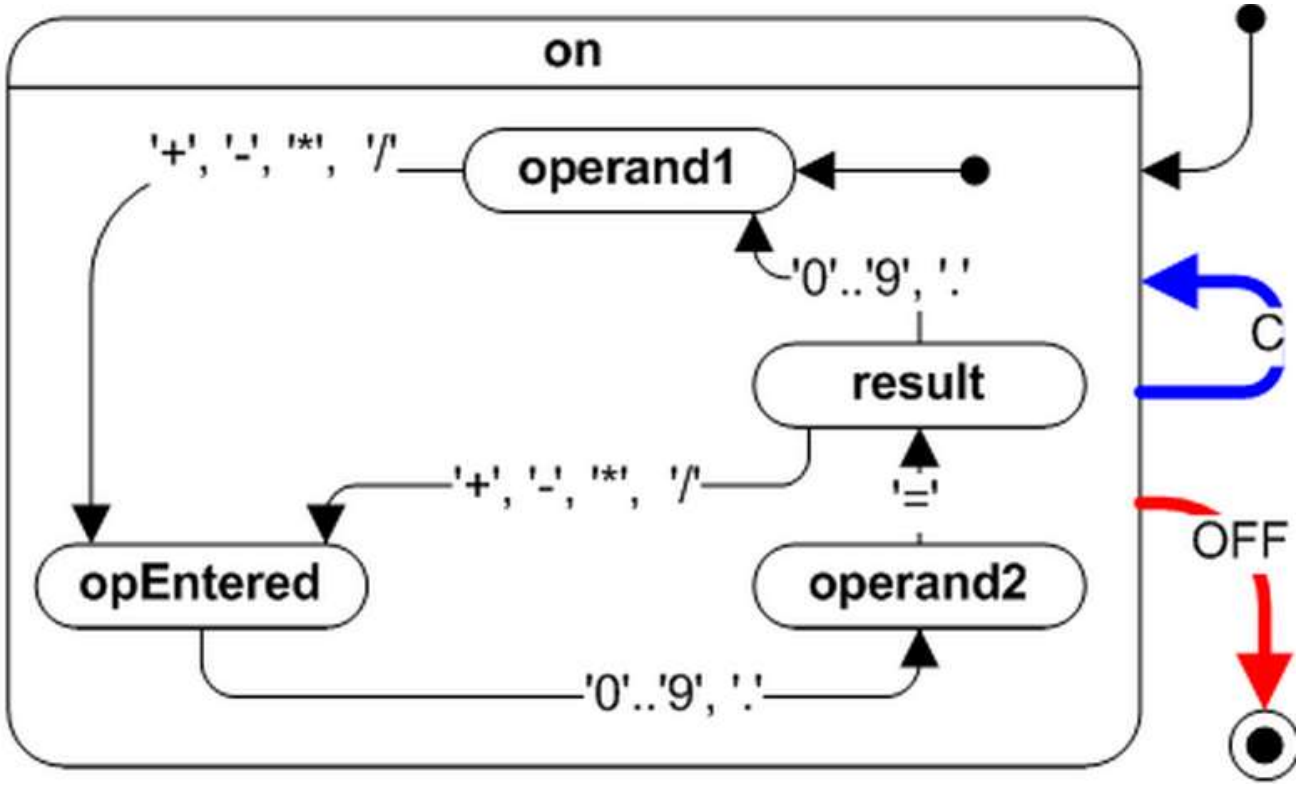
The power of hierarchical states

- Common calculator requests: clear display or turn off



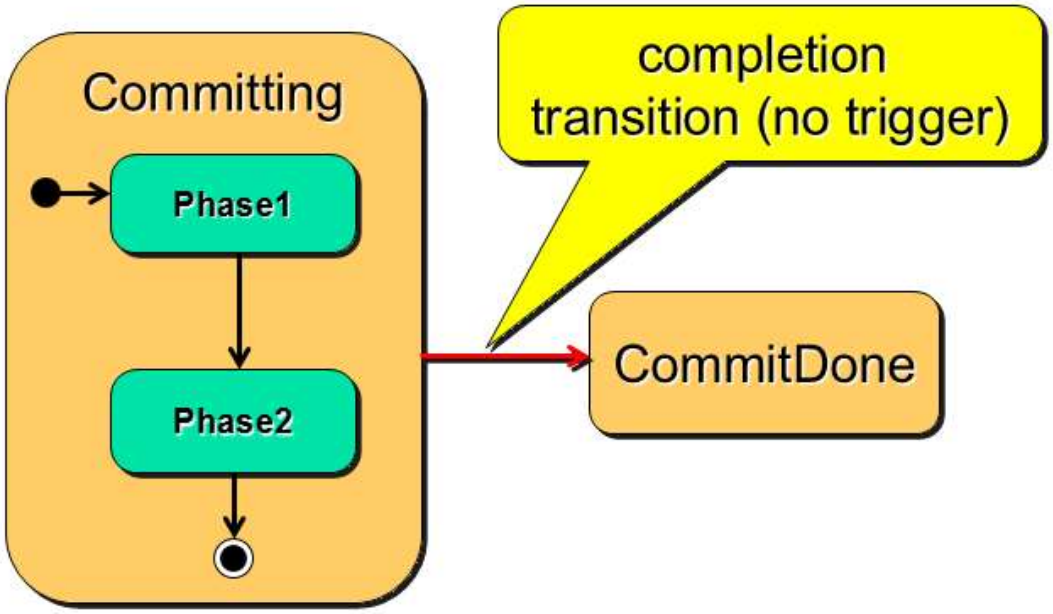
The power of hierarchical states

- Simplify via hierarchical states

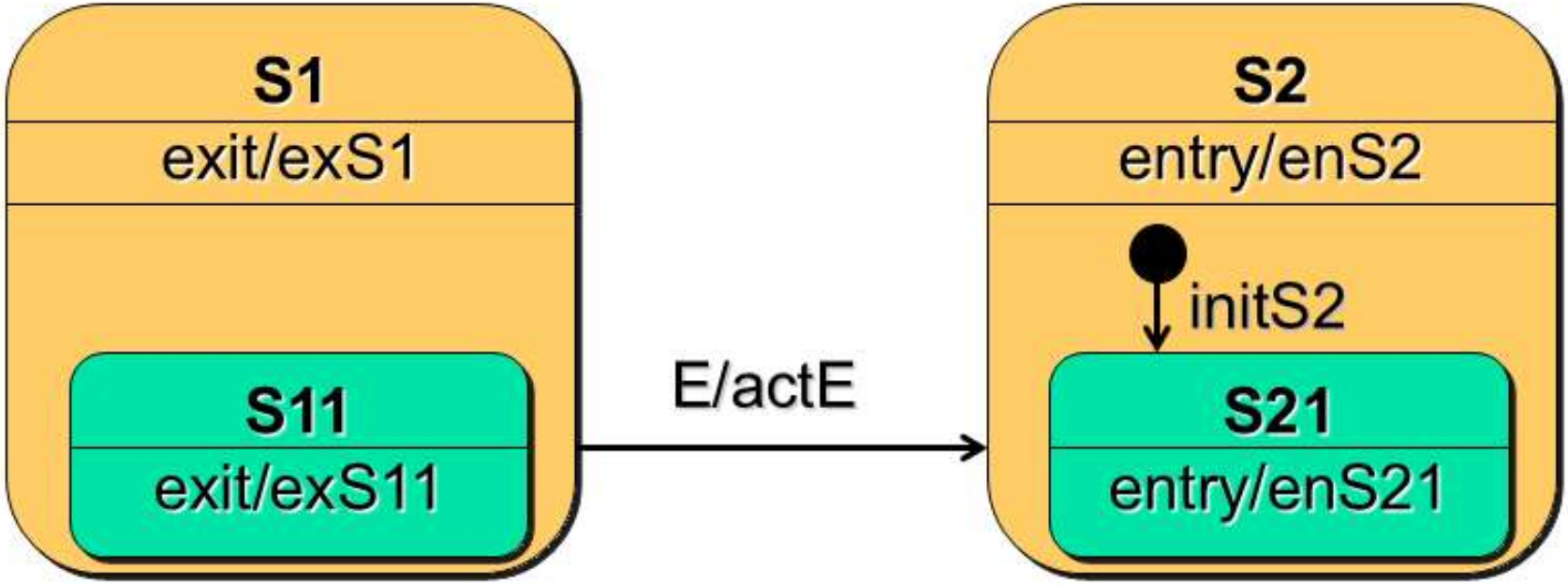


State completion transitions

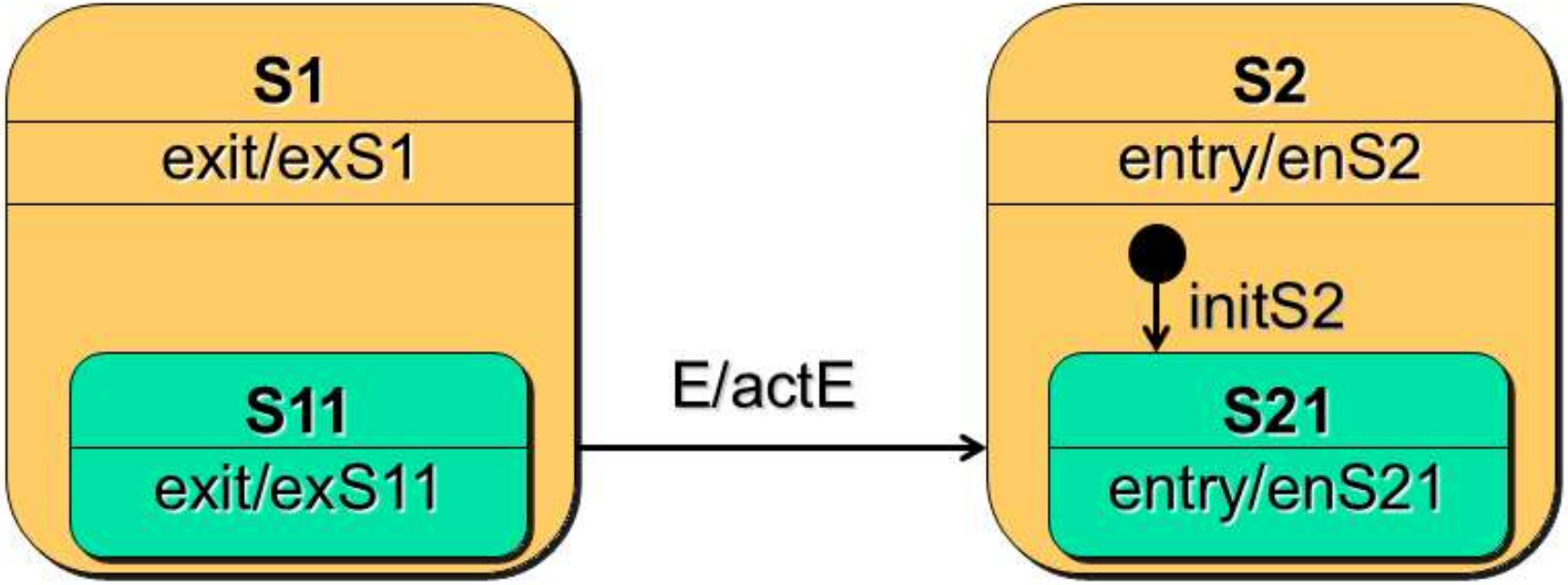
- Triggered by a completion event
  - Generated automatically when a nested state machine terminates



Action execution order

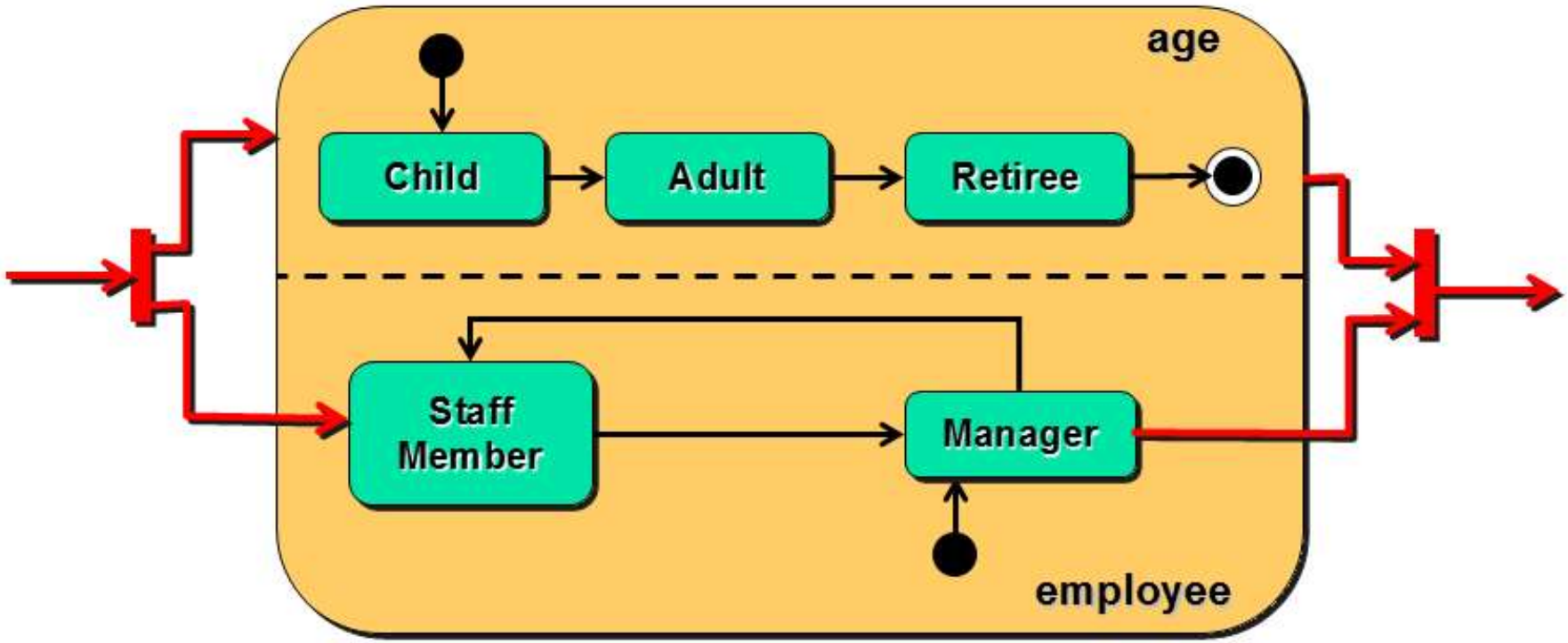


Action execution order



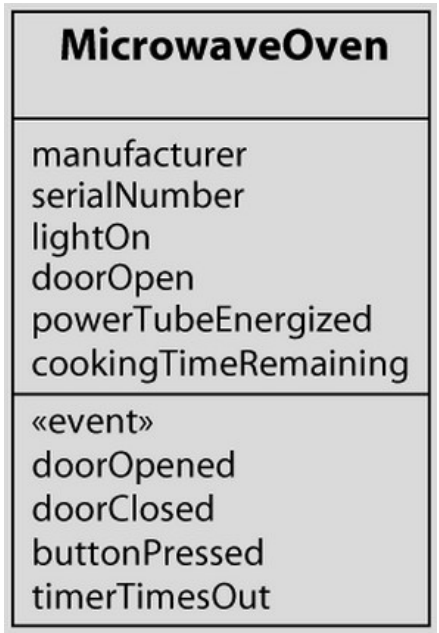
exS11 | exS1 | actE | enS2 | initS2 | enS21

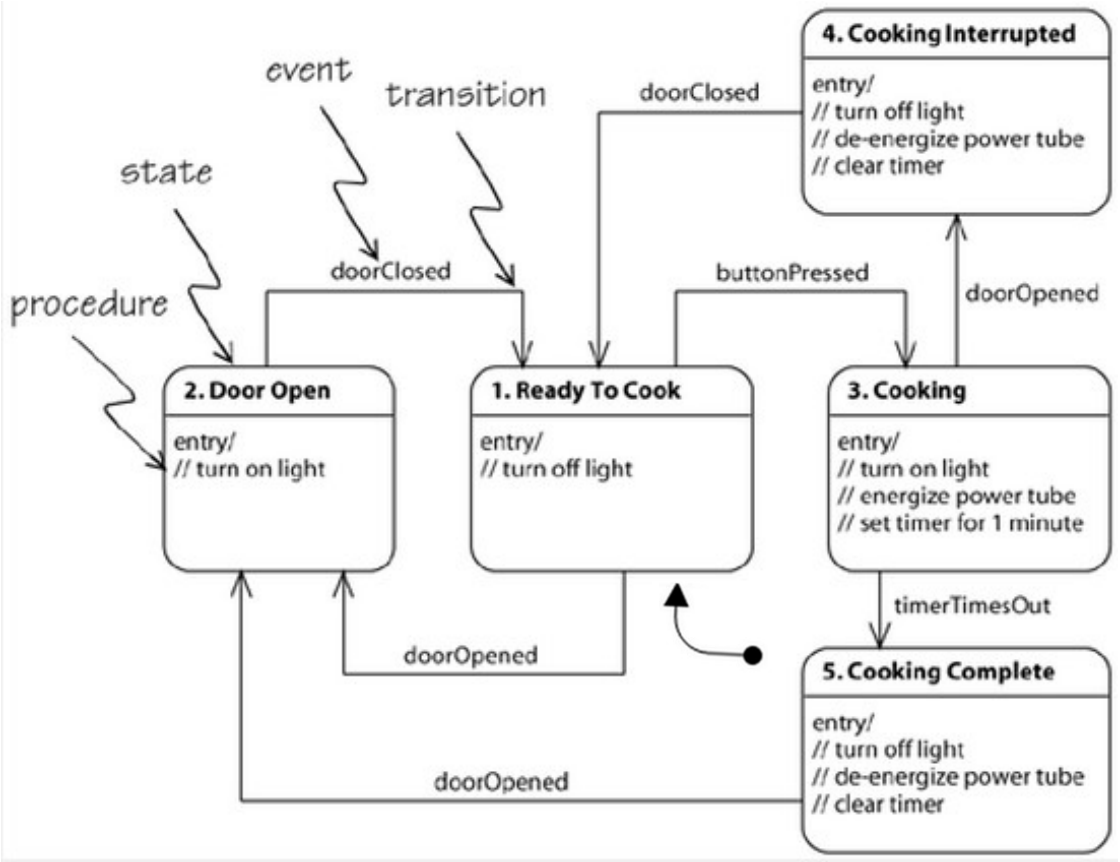
Concurrent transition forks and joins



State machine examples

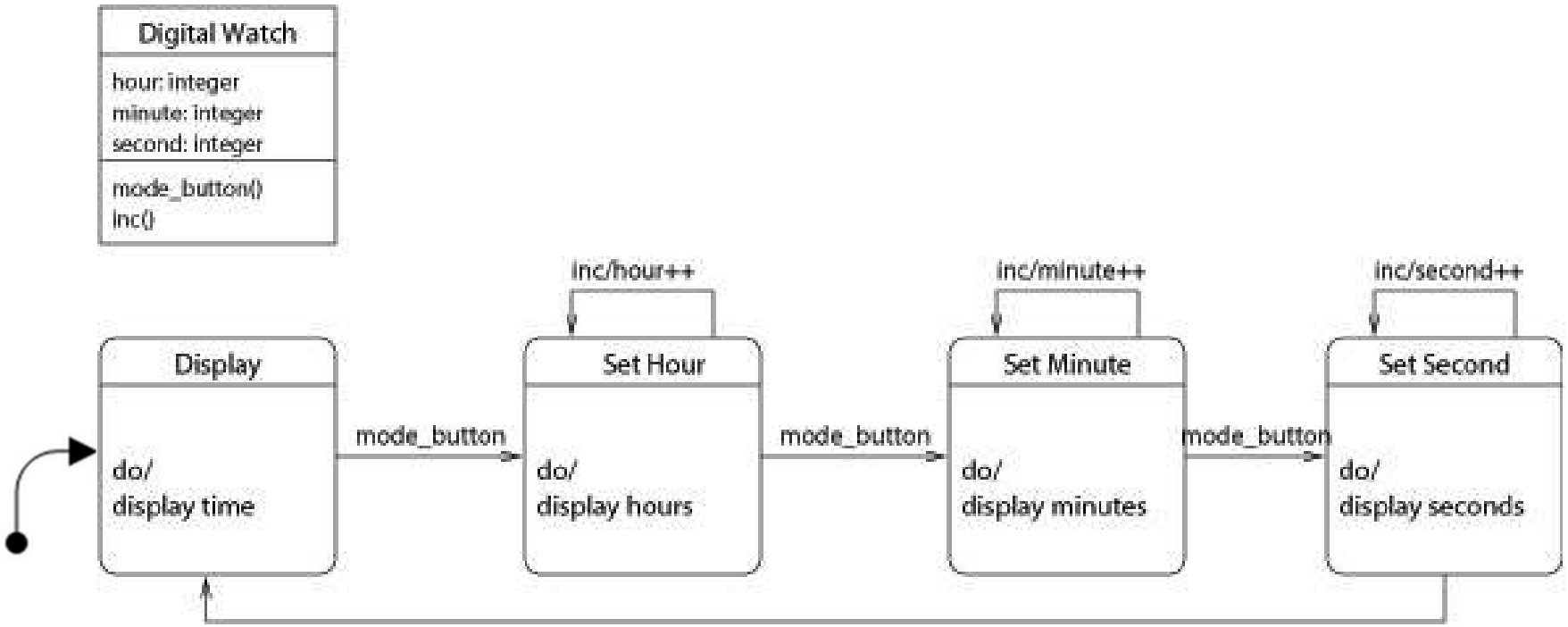
- A simple (1-minute) microwave system





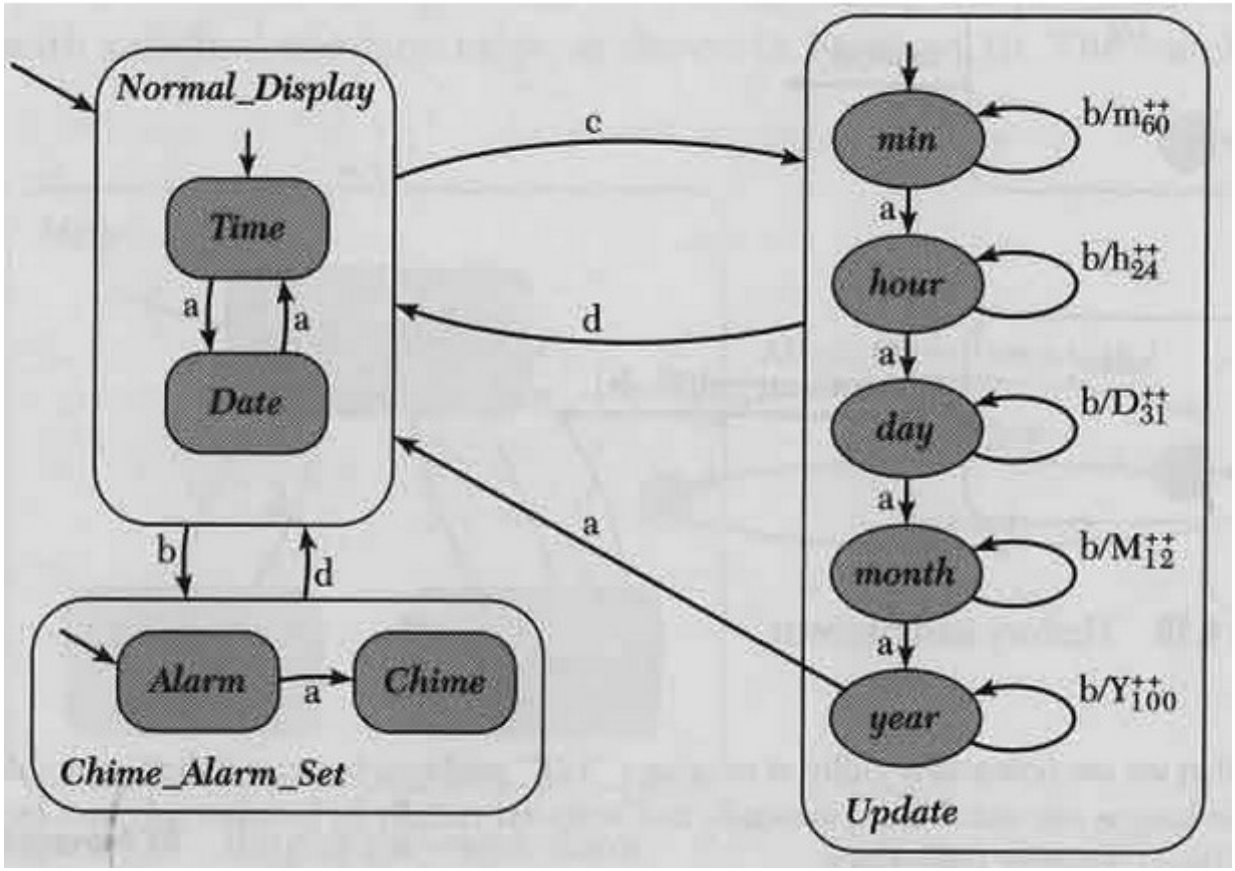
State machine examples

- A simple digital watch (two modes)



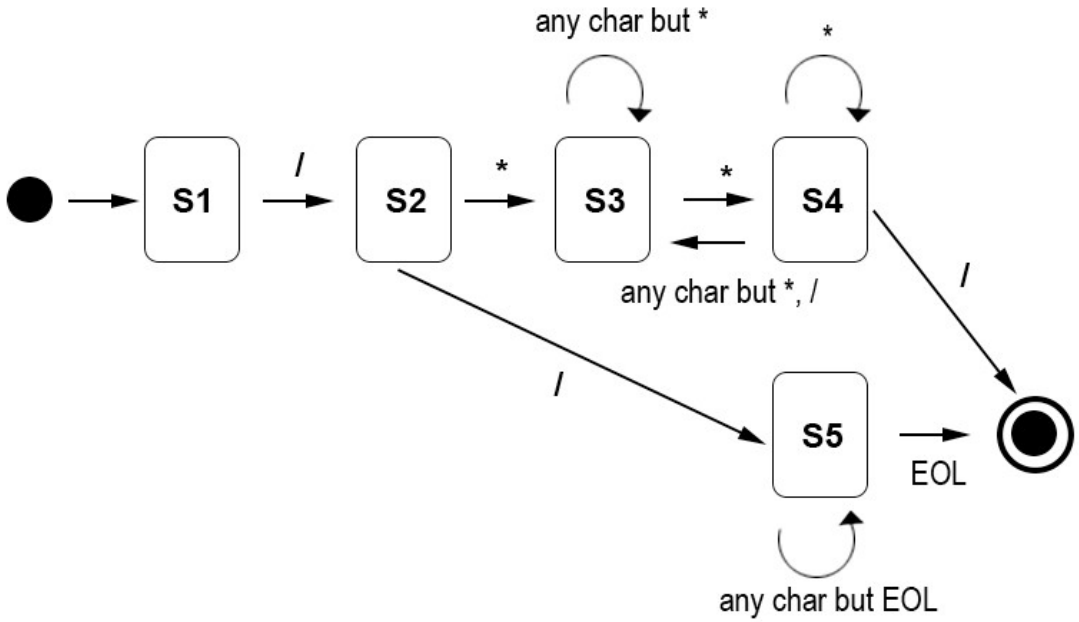
State machine examples

- A digital watch with three modes



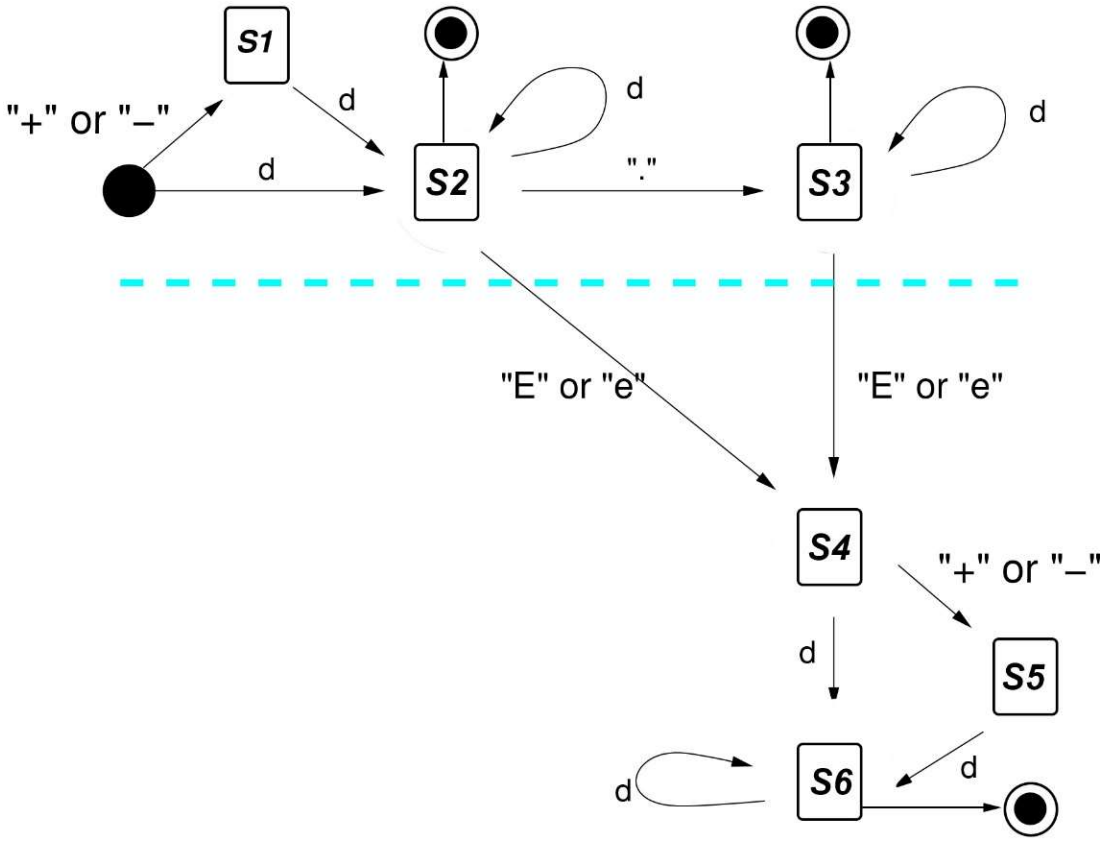
State machine examples

- An lexical analyzer object handling C/C++ comments
- More precise transitions can be stated as, e.g., `readchar()[char == "*"]`



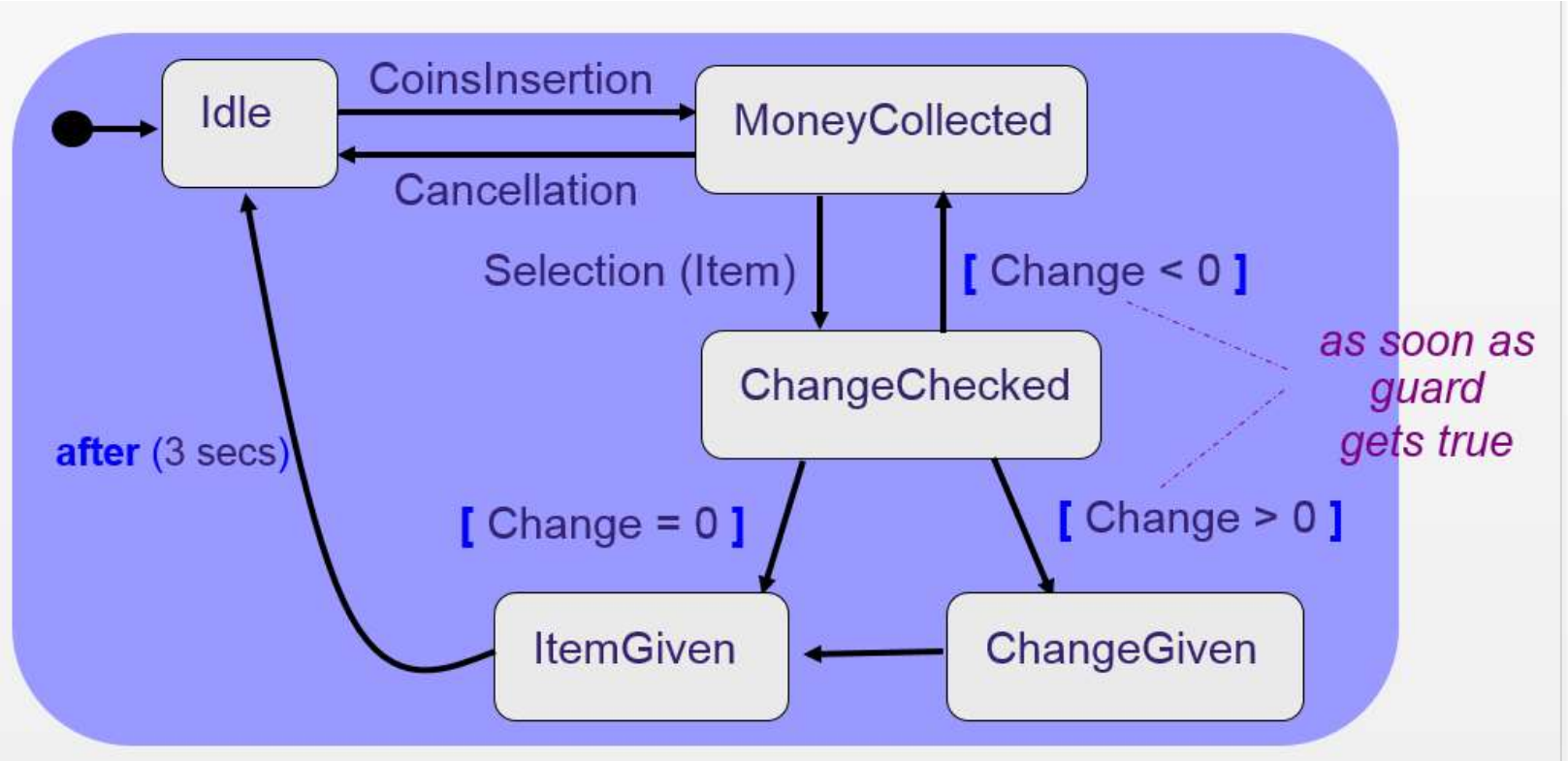
State machine examples

- An lexical analyzer object handling C/C++ floating point numbers



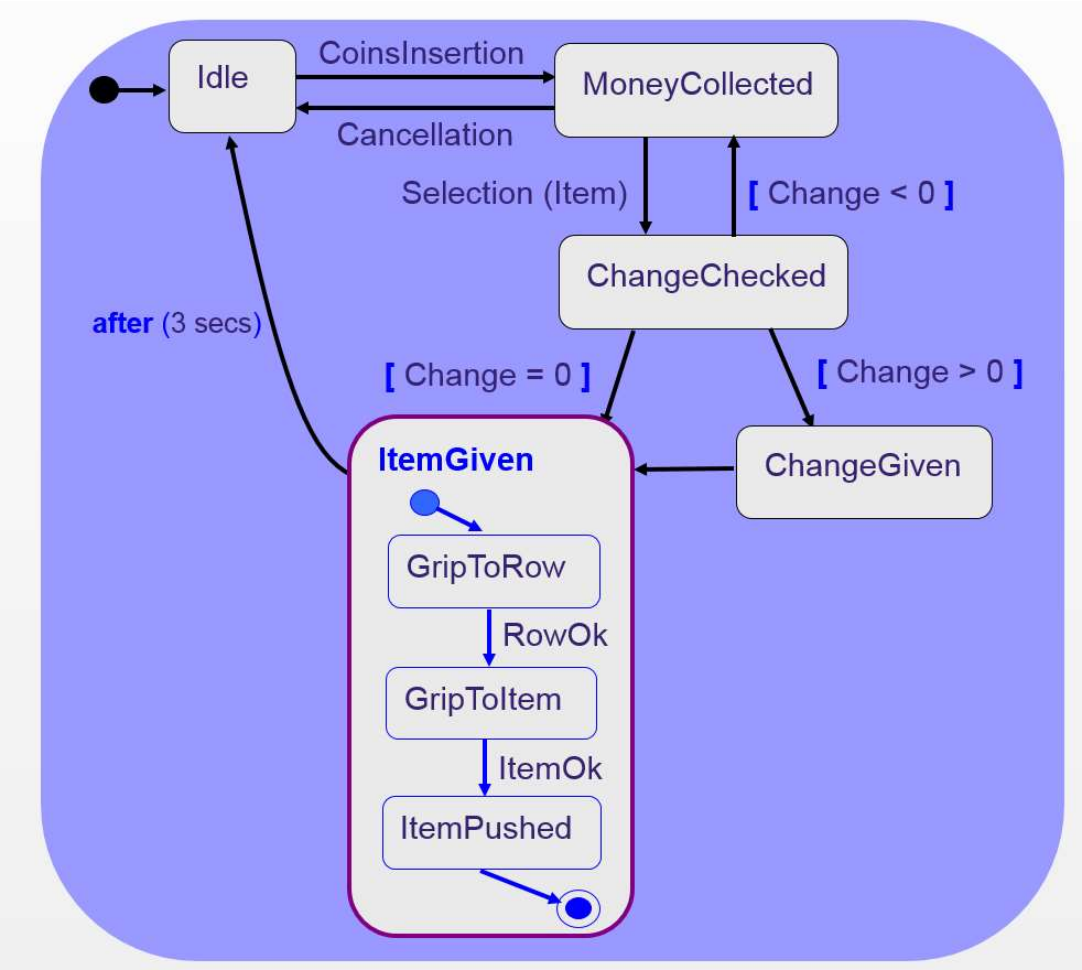
State machine examples

- High level spec of a vending machine



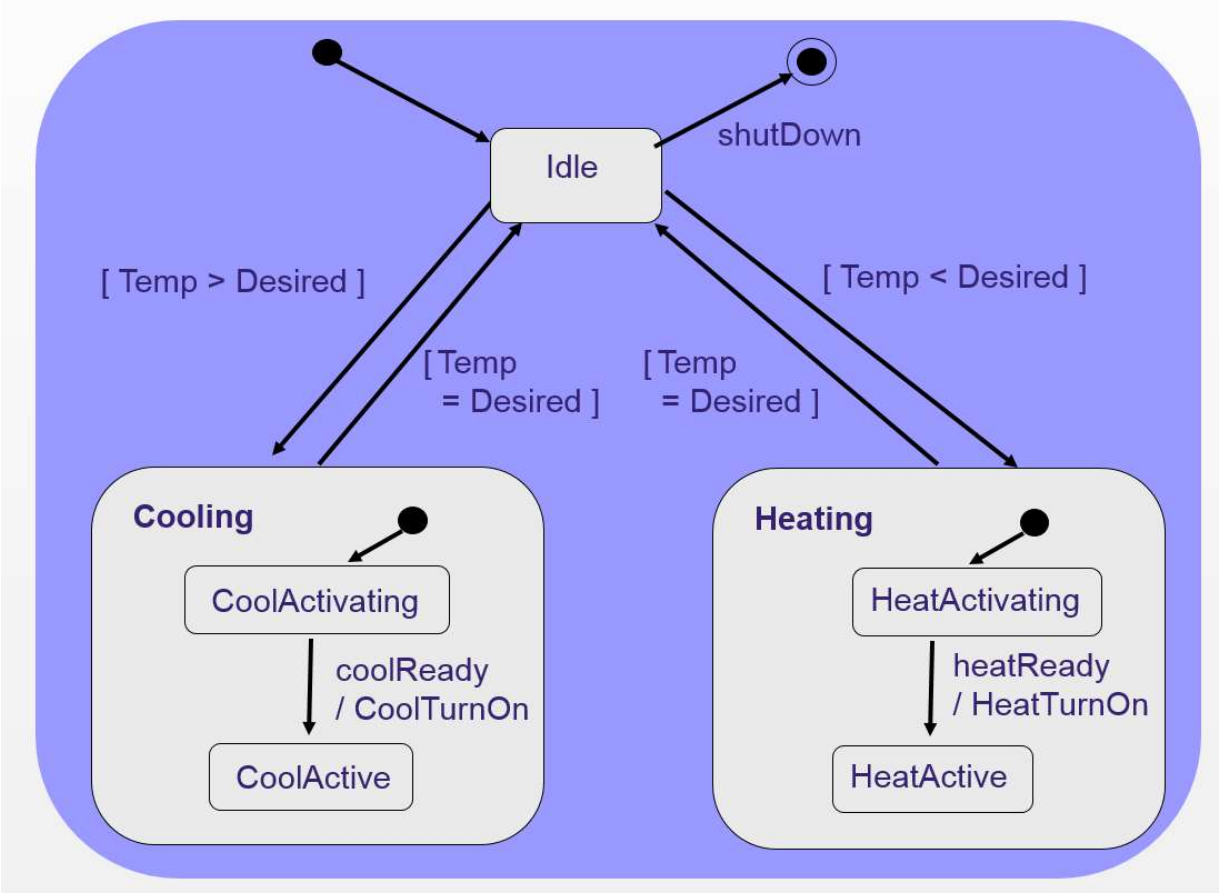
State machine examples

- Start with high level specs and gradually elaborate



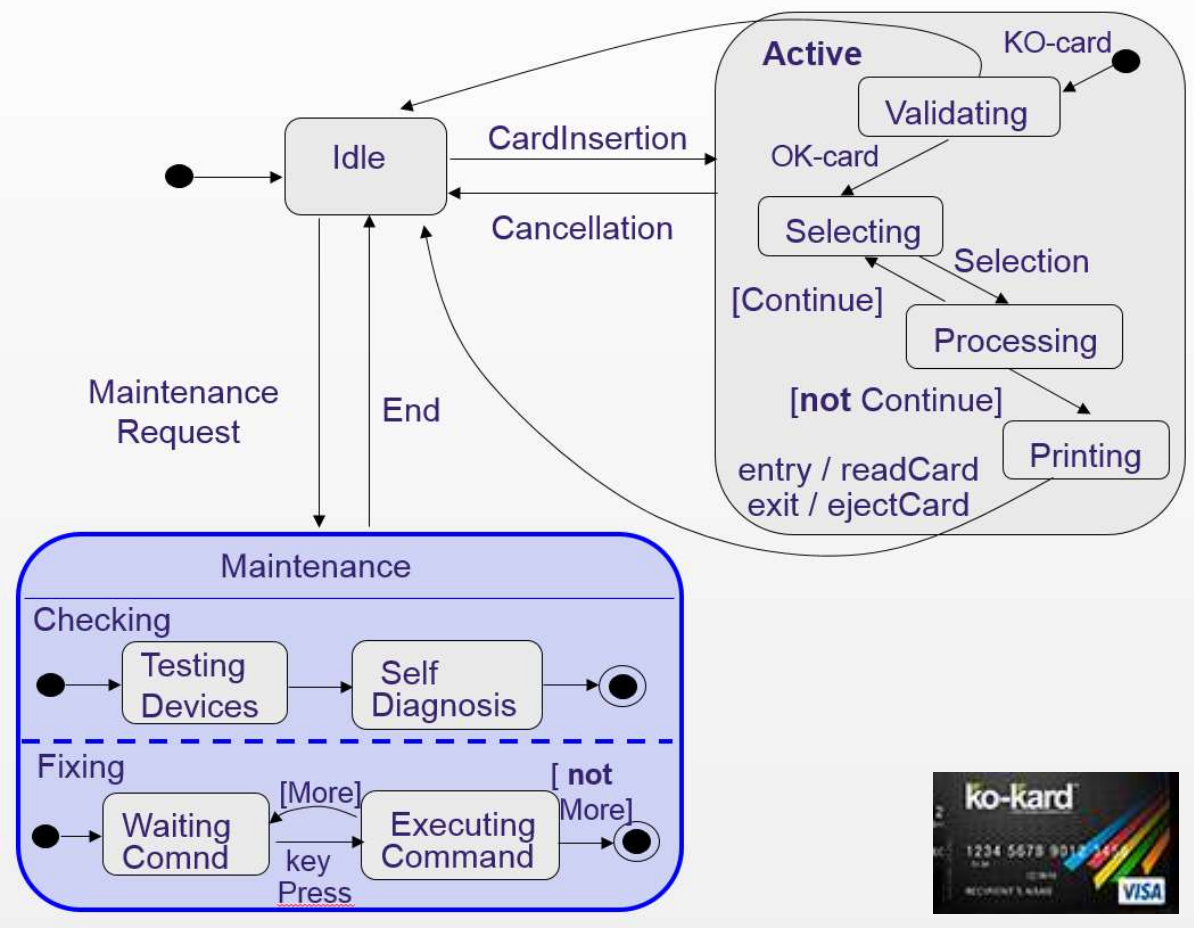
State machine examples

- A simple thermostat controller



State machine examples

- An ATM machine



Behavioral modeling examples and summary

Professor Hossein Saiedian

EECS 448: Software Engineering

December 8, 2022

THE UNIVERSITY OF KANSAS

State transition (statechart) modeling

- Used for modeling a system (or an object) behavior as a set of states it will have
- The states represent the object's reaction to events
- Upon an event, a system (object) may stay in the same state or make a transition to another state
- The FSMs are the basis for the statecharts, state modeling, state transition modeling, etc
- UML has added many features to make state modeling (requirements, design) more friendly

A state

- Models based on FSM (statecharts) divide complexity into components known as states
- A situation during the life of an object during which it satisfies some condition, performs some action, or waits for some event
- Example of states:
  - An elevator: stopped, going up, going down
  - A modem: idle, sending, receiving
  - An airplane: on ground, taking off, flying, landing

A transition

- A transition is the changing from one state of an entity to another
- Transitions are the statechart representation of responses to events
- Transitions may have associated actions
- Transitions may be guarded

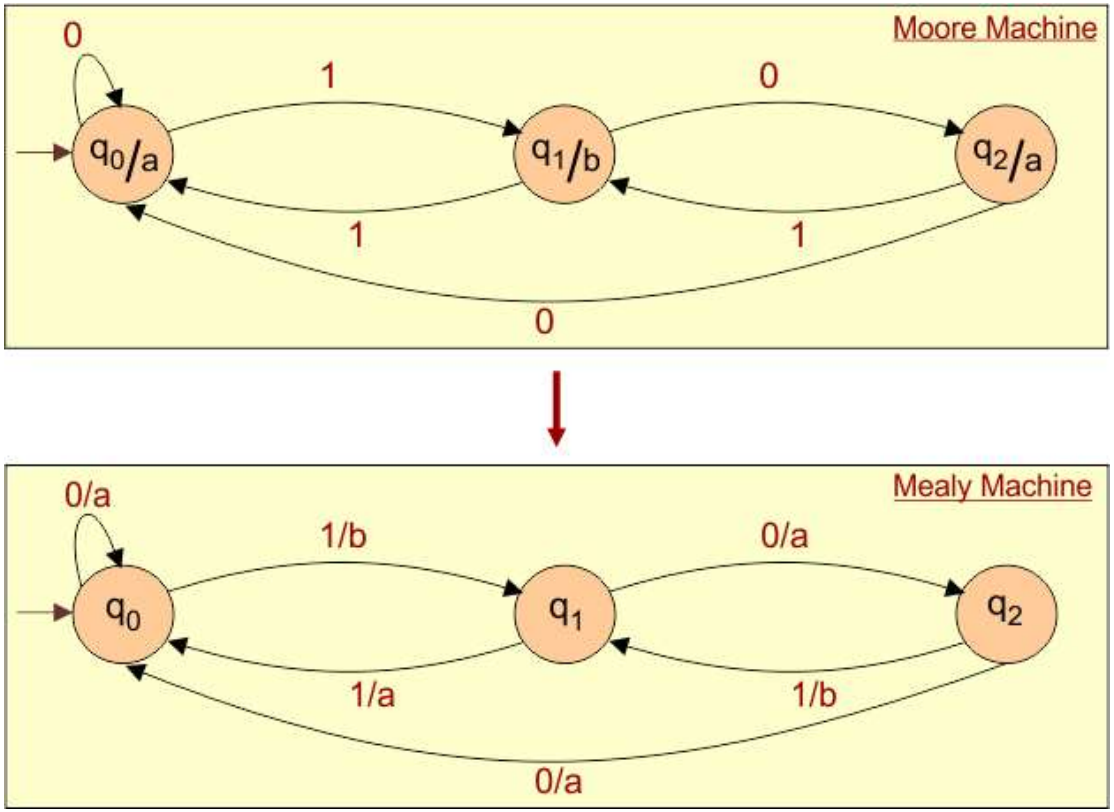
Actions

- Actions are functions that take an insignificant amount of time to perform
- Actions are implemented via an object's operations (or externally available functions)
- An action may occur when
  - A transition is taken
  - A state is entered
  - A state is exited

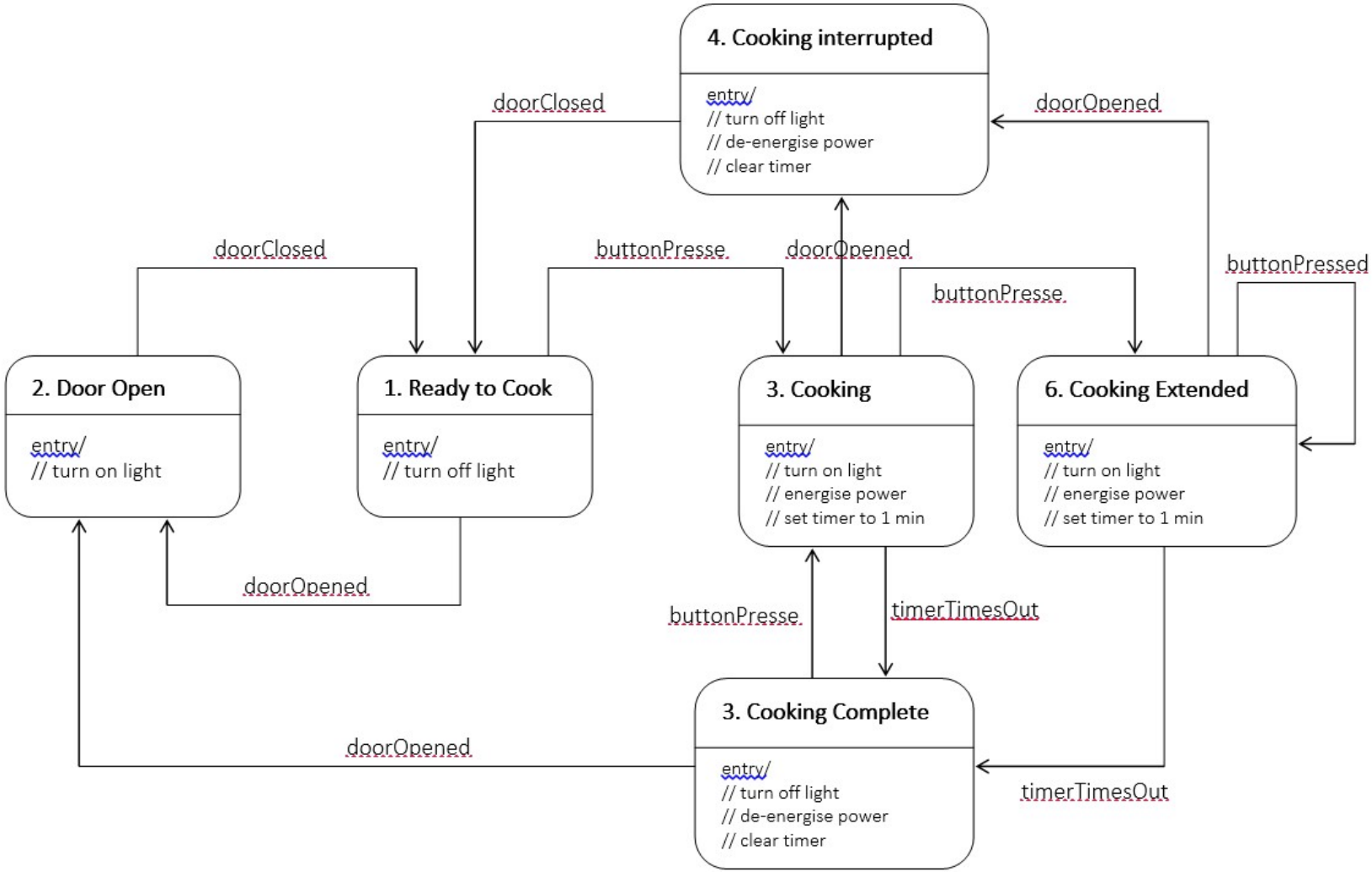
The UML notation

- More structured
- Additional syntactical and semantical elements
- Notations for better abstraction
  - Model systems

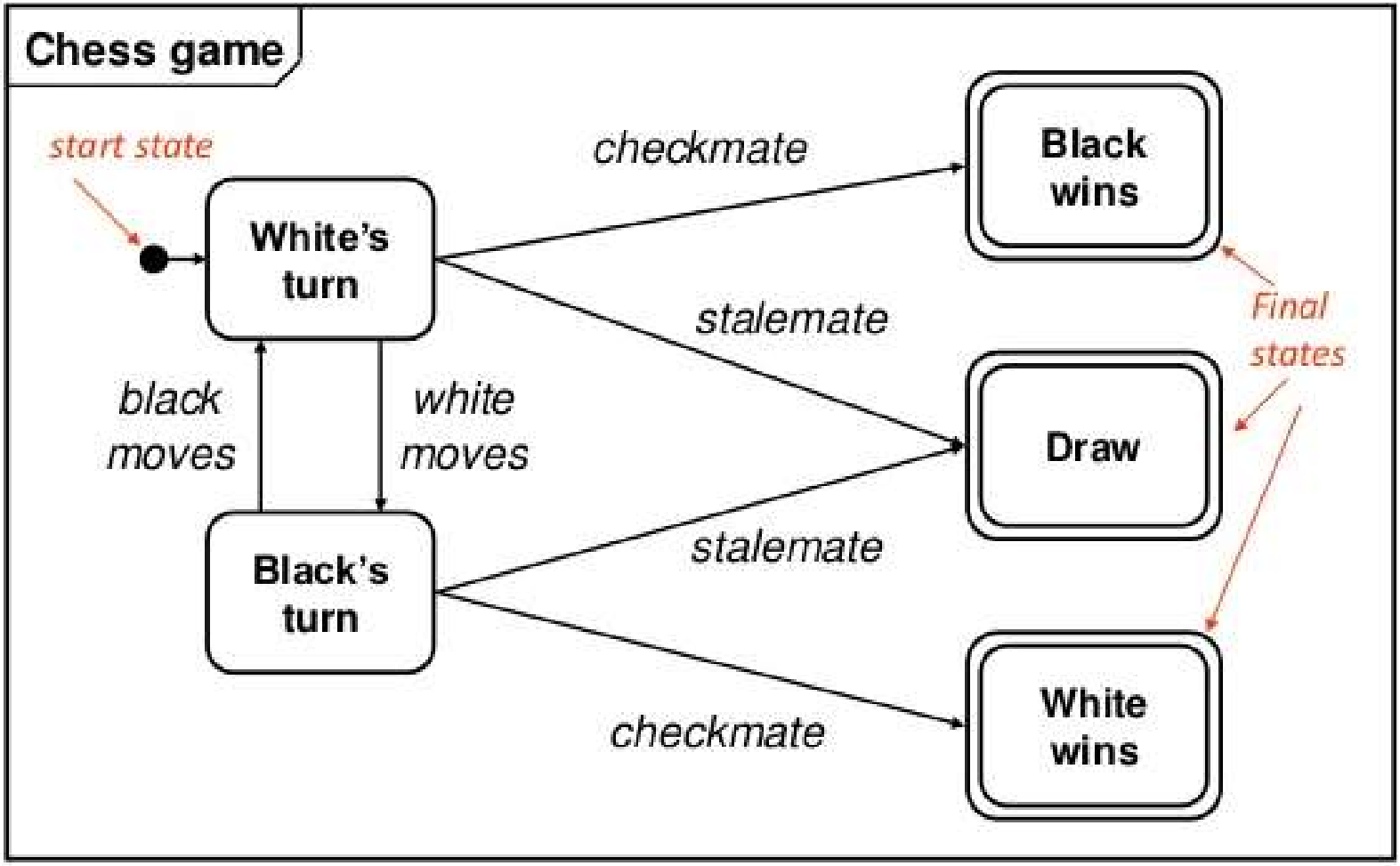
Moore vs Mealy vs UML



Example: a microwave

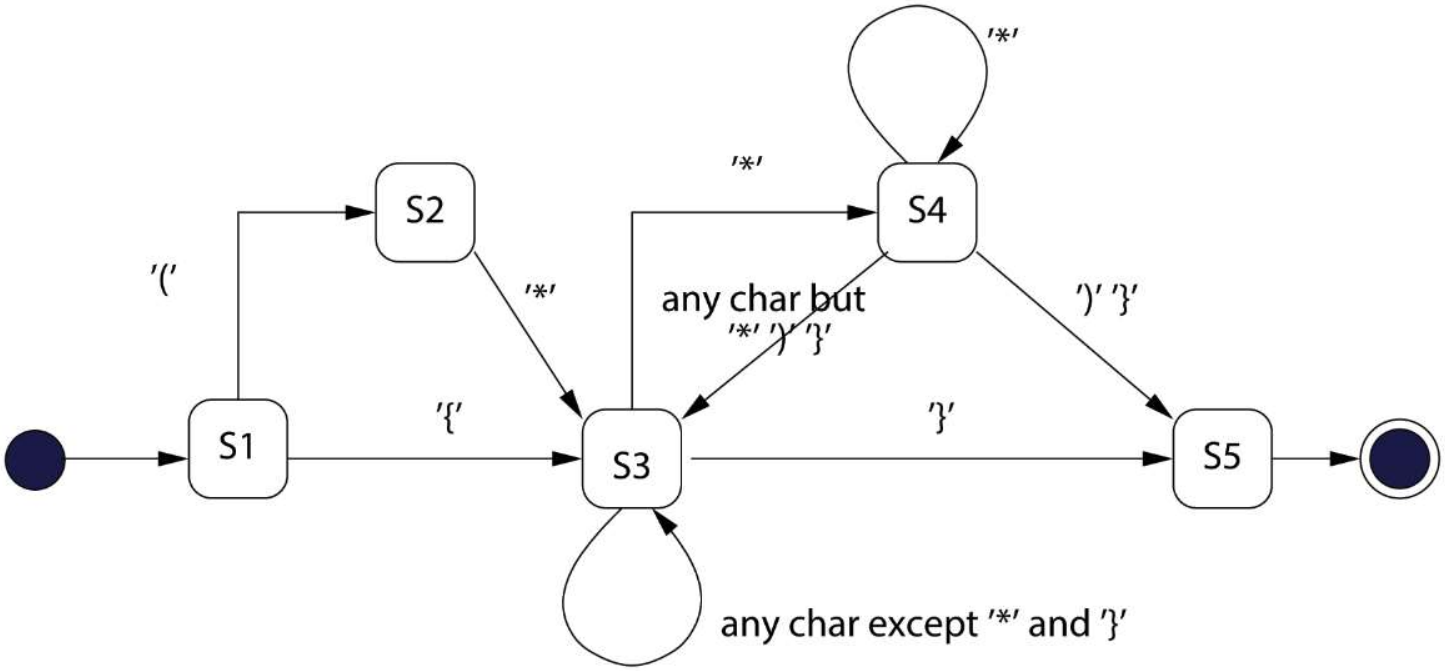


Example: a chess game



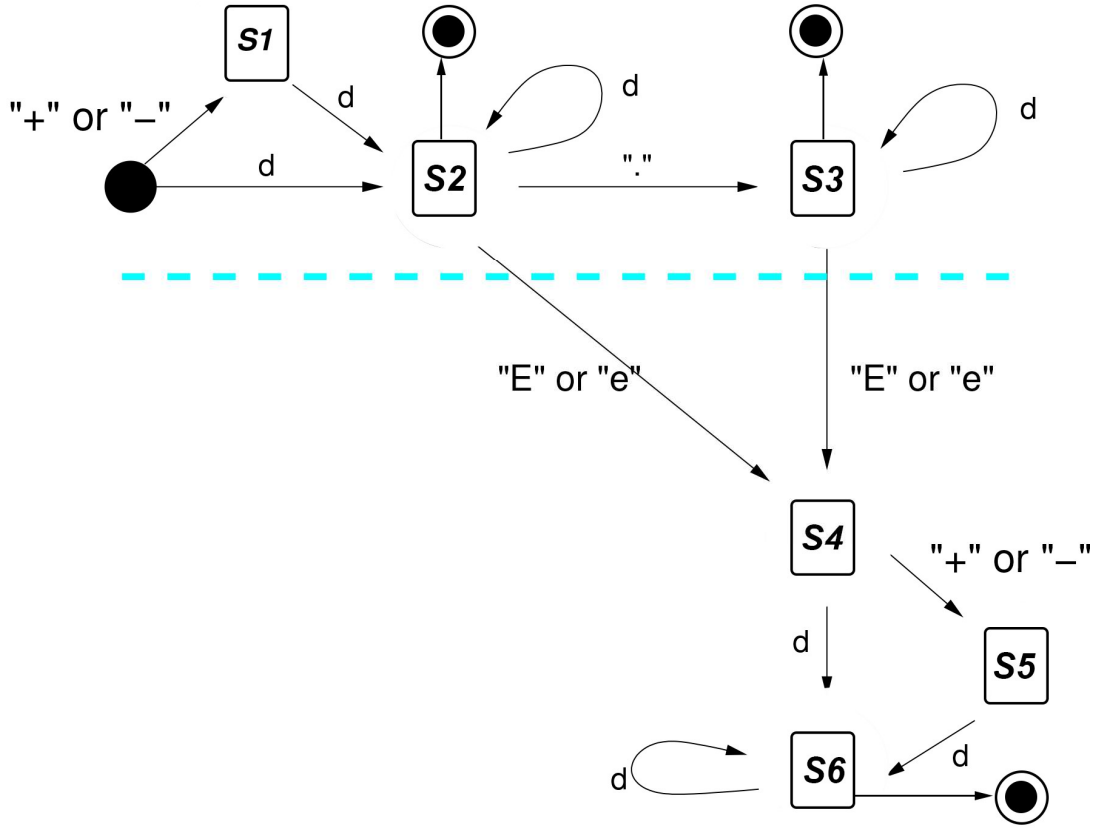
Example: a machine for accepting comments

- Pascal-like comments
  - Start with (* or {
  - Close with *) or }



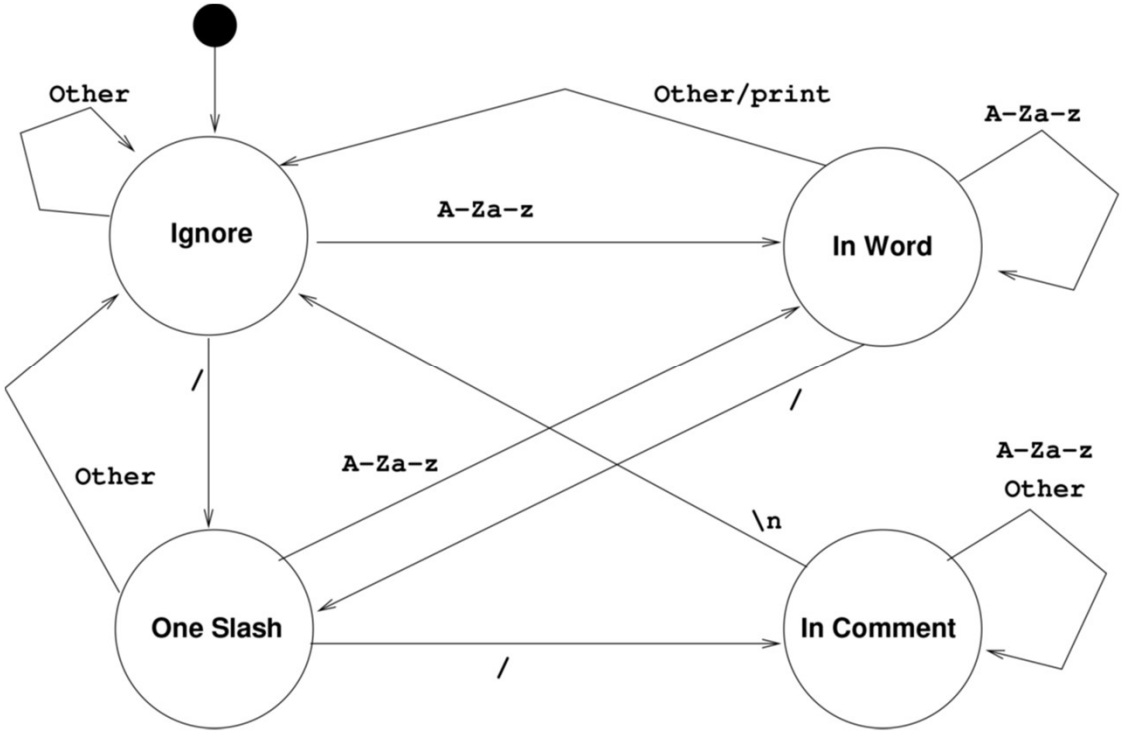
Example: a machine for accepting floating-points

- An FSM to accept floating point numbers in C/C++

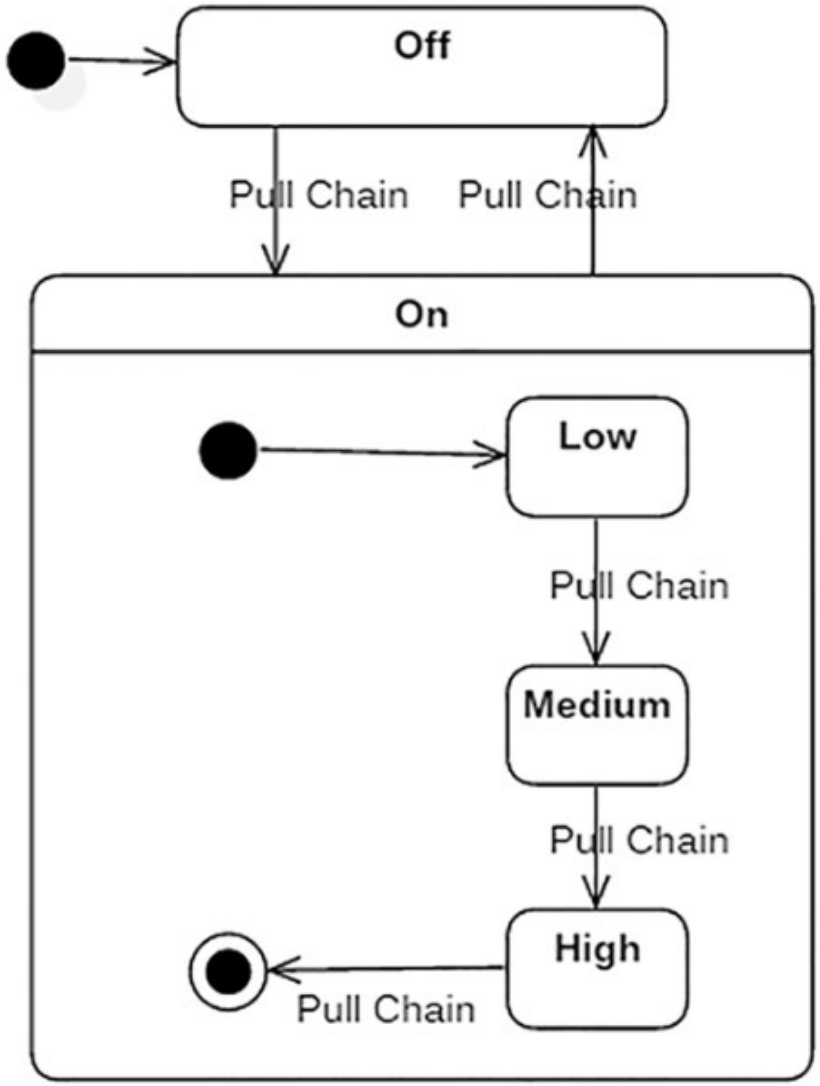


Example: accepting C++ identifiers

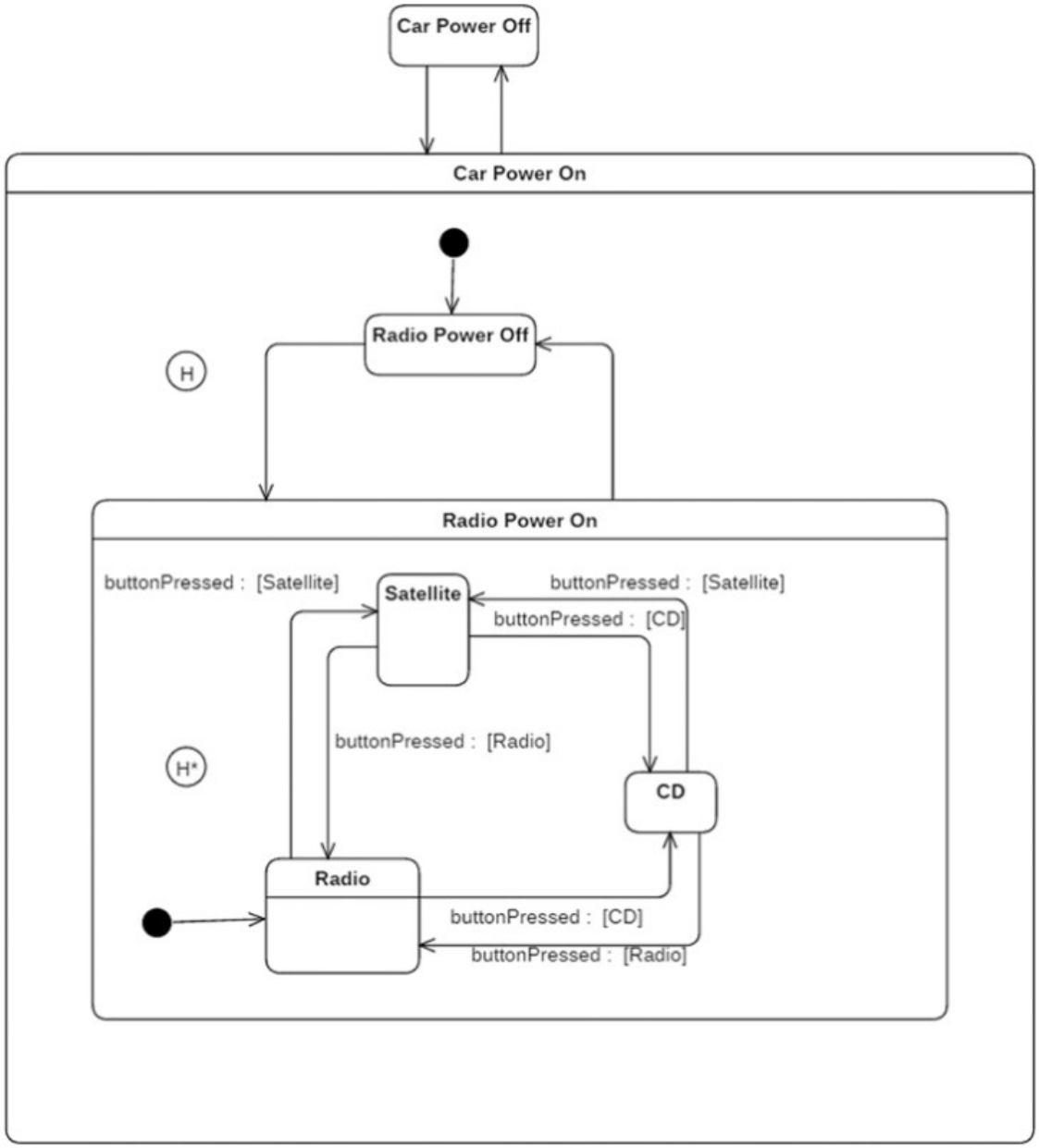
- Model a machine that recognizes words (identifiers) in a C++ source file but ignores the words in comments
  - Words: any alpha chars including "/"
  - Comments: Start with // and extend to the end of the line



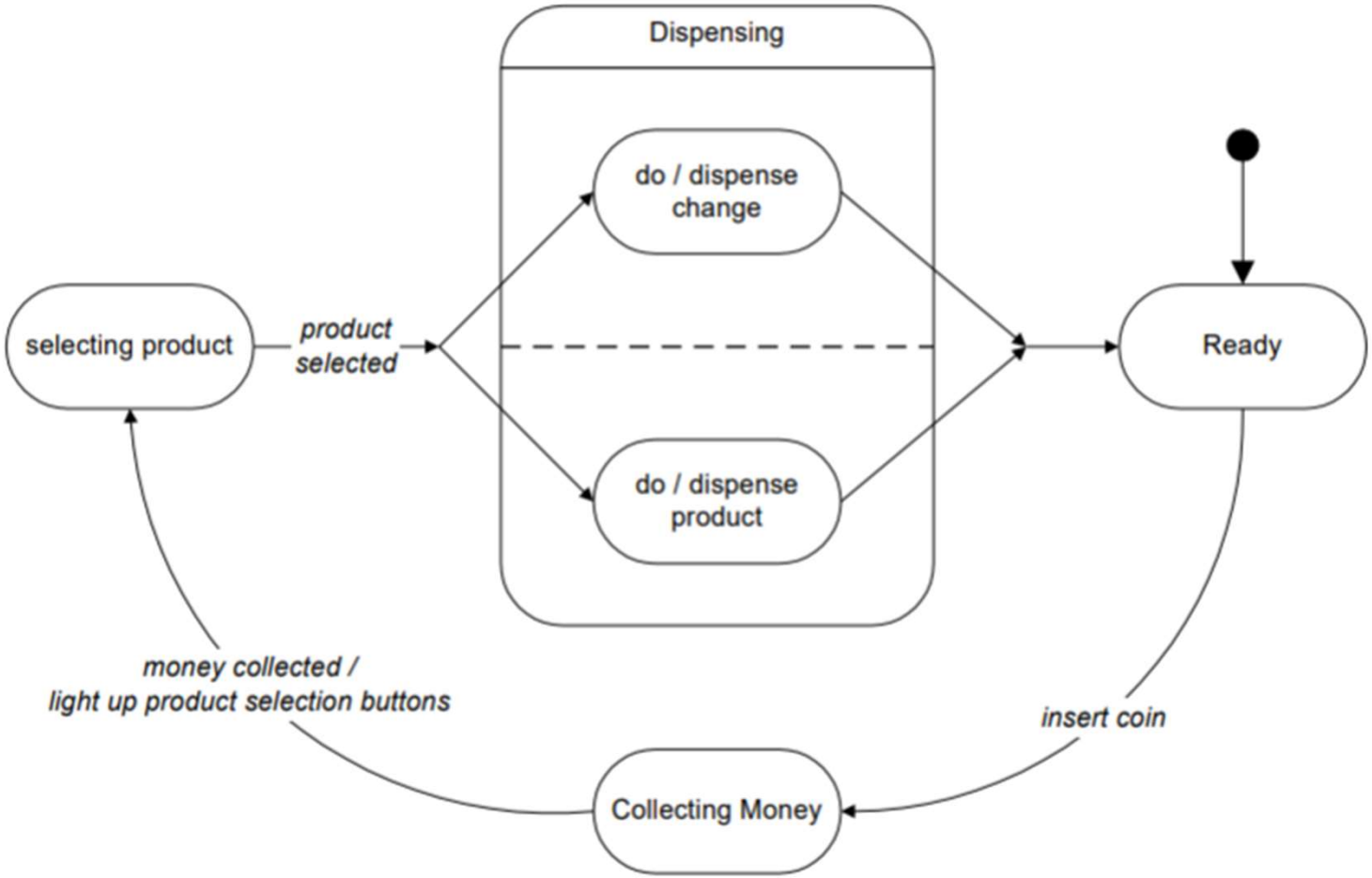
Example: modeling a ceiling fan



Example: modeling a a car stereo system

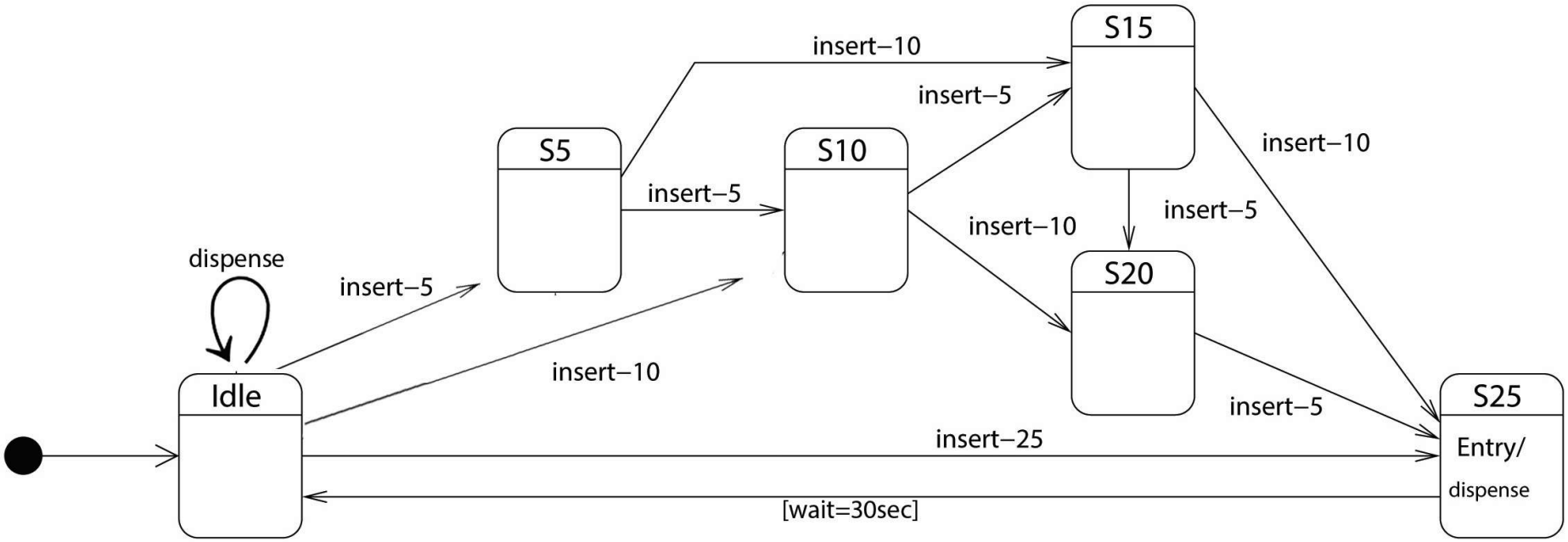


Example: modeling a dispensing machine



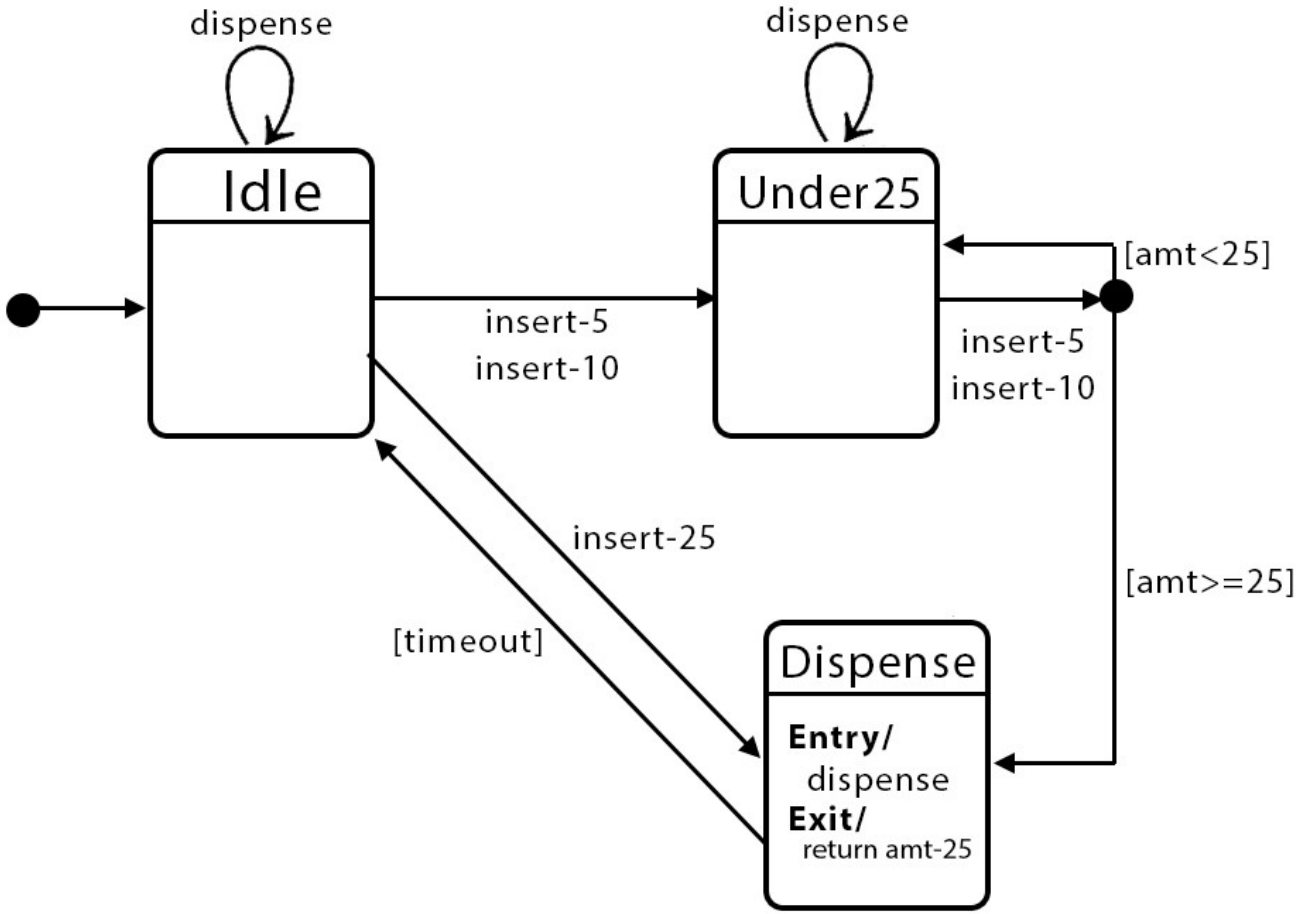
Example: modeling a dispensing machine

- Accepts coins only; 25-cent items only



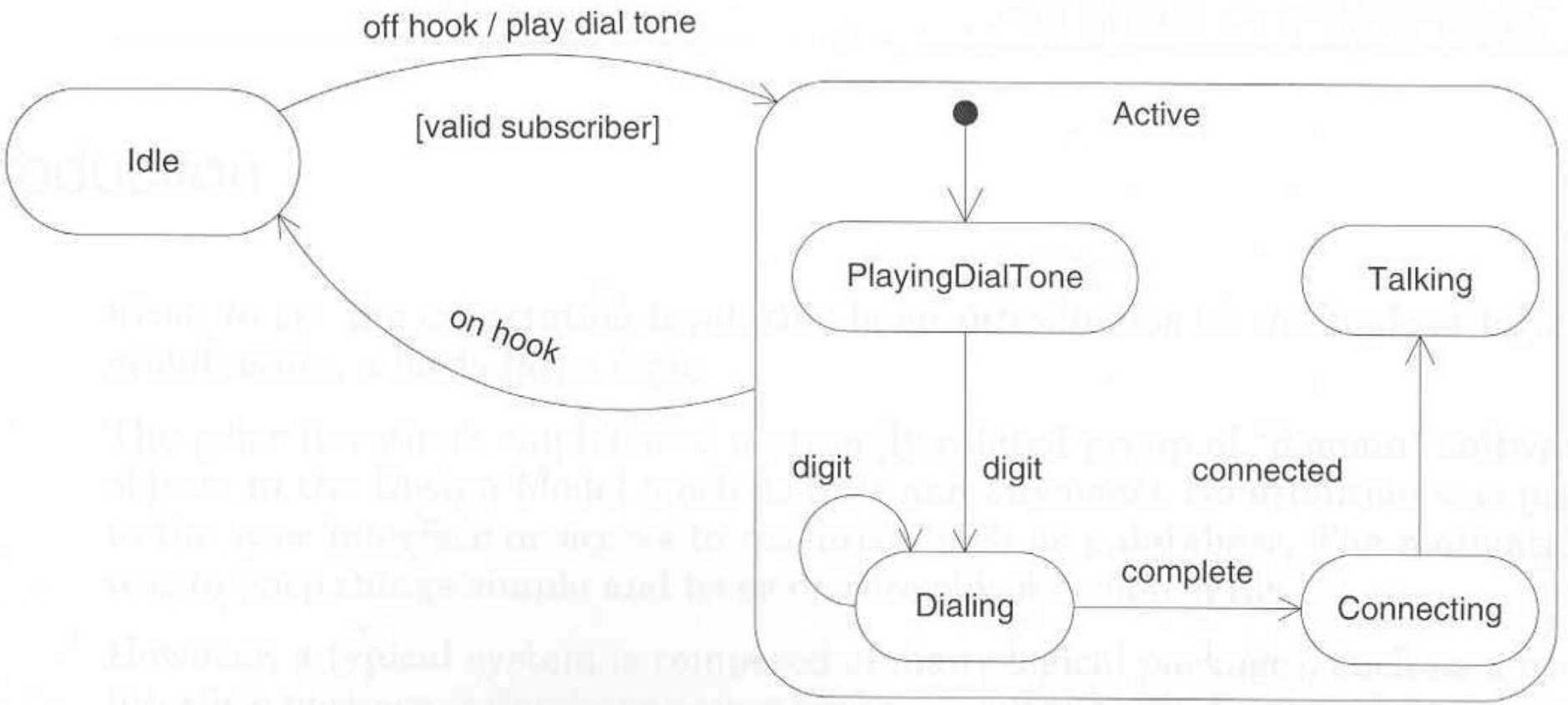
Example: modeling a dispensing machine

- Accepts coins only; 25-cent items only



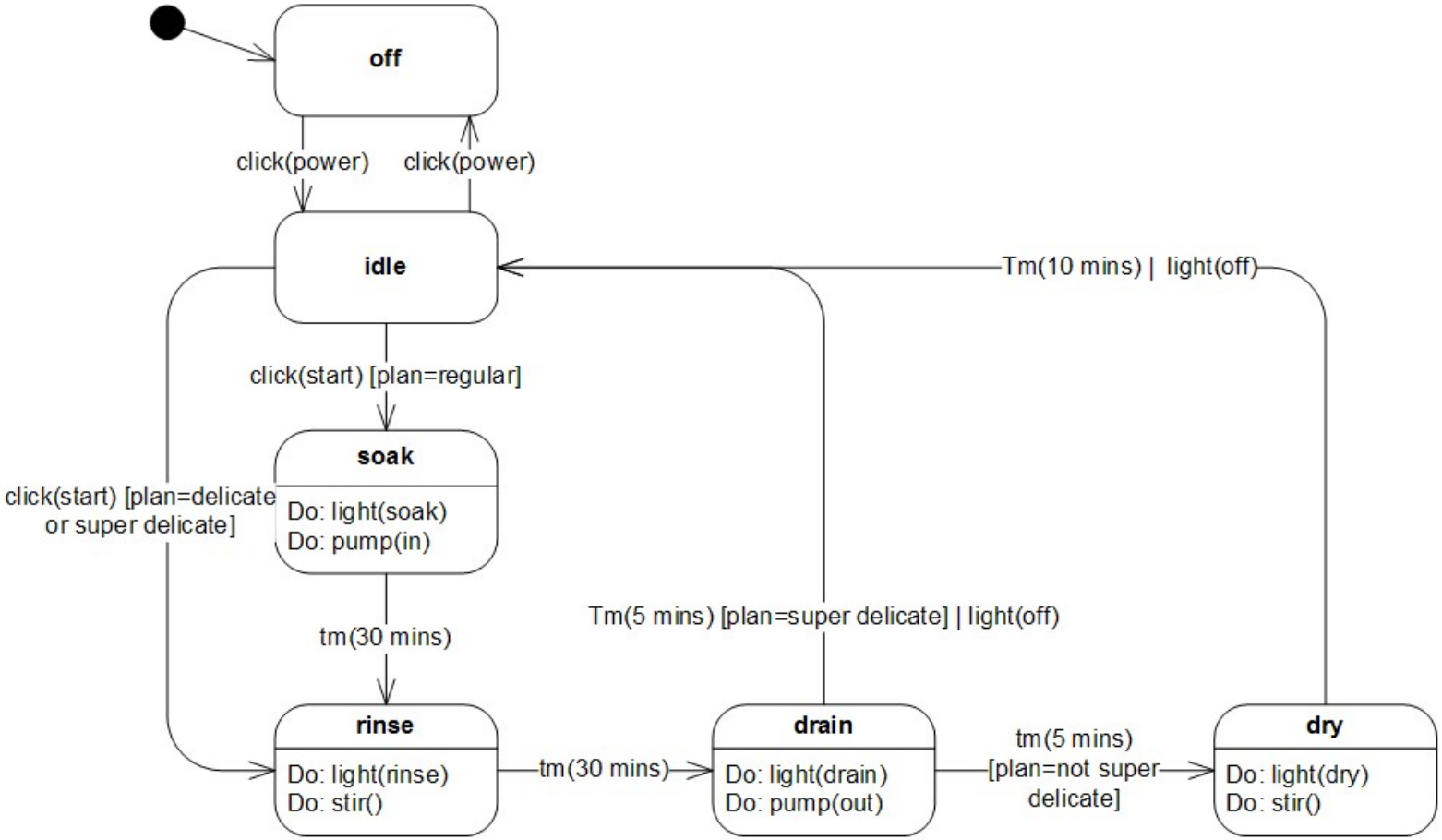
Example: a telephone device

- A telephone unit (Larman, 2002)

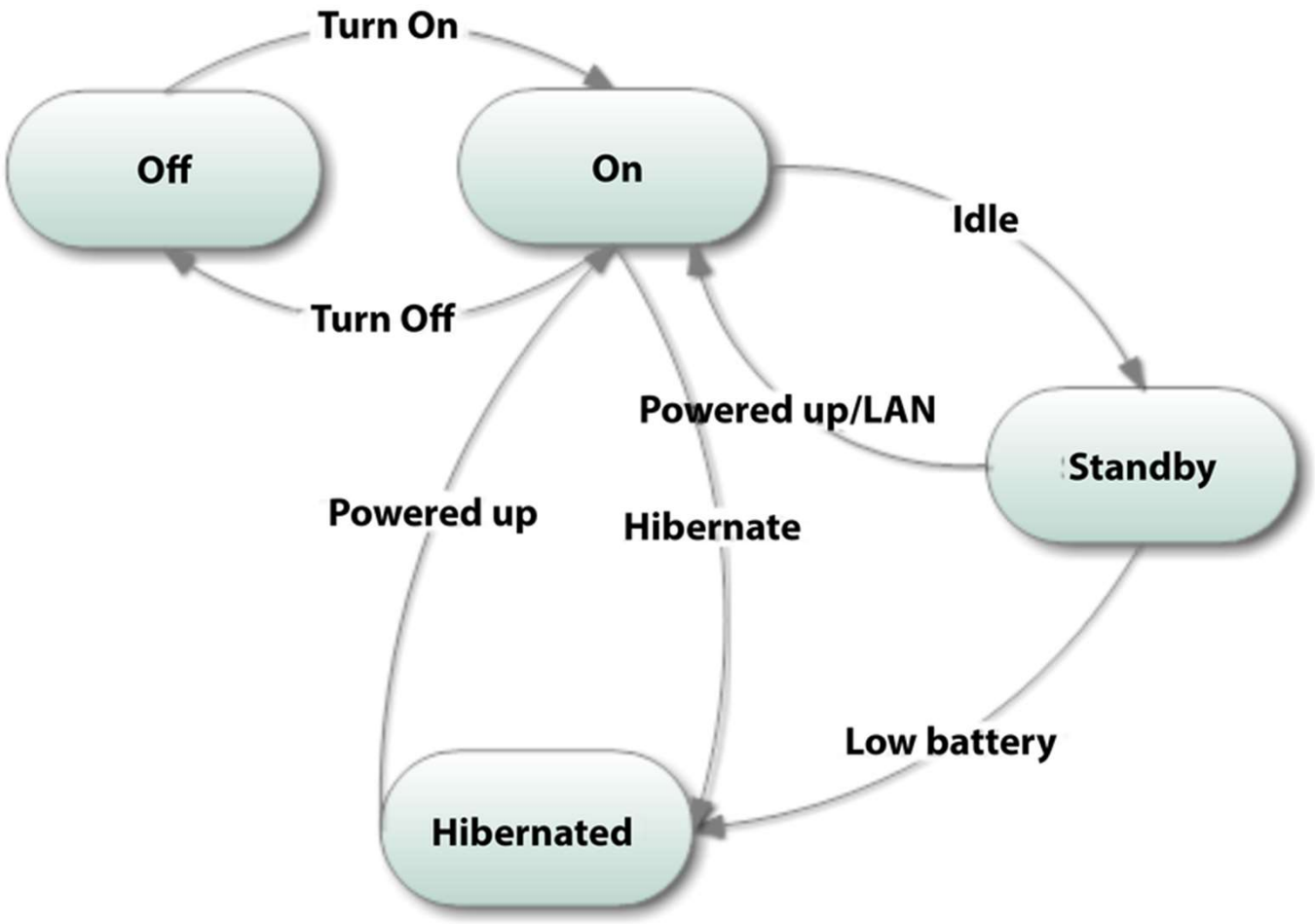


Example: a washing machine

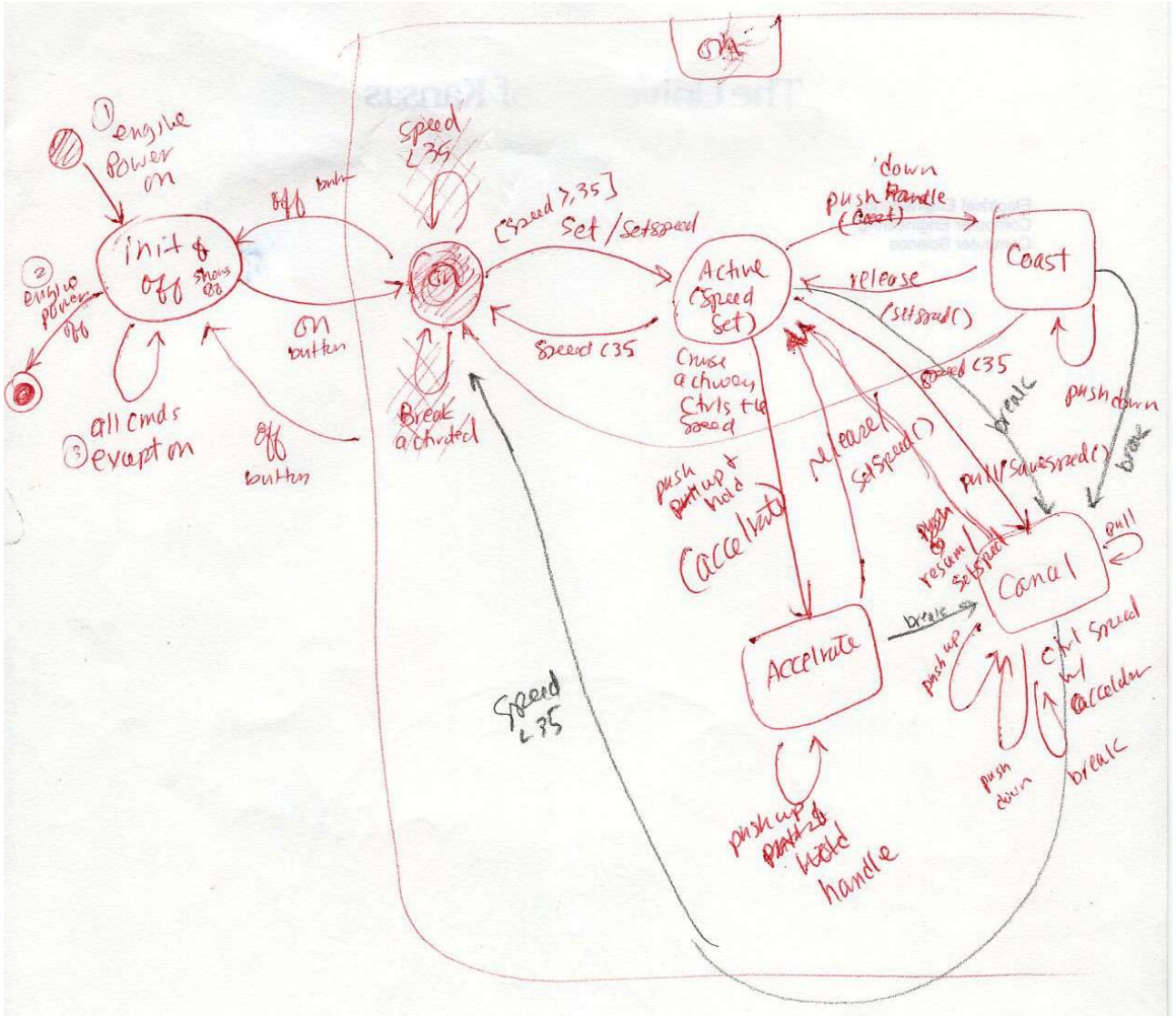
- A washing machine



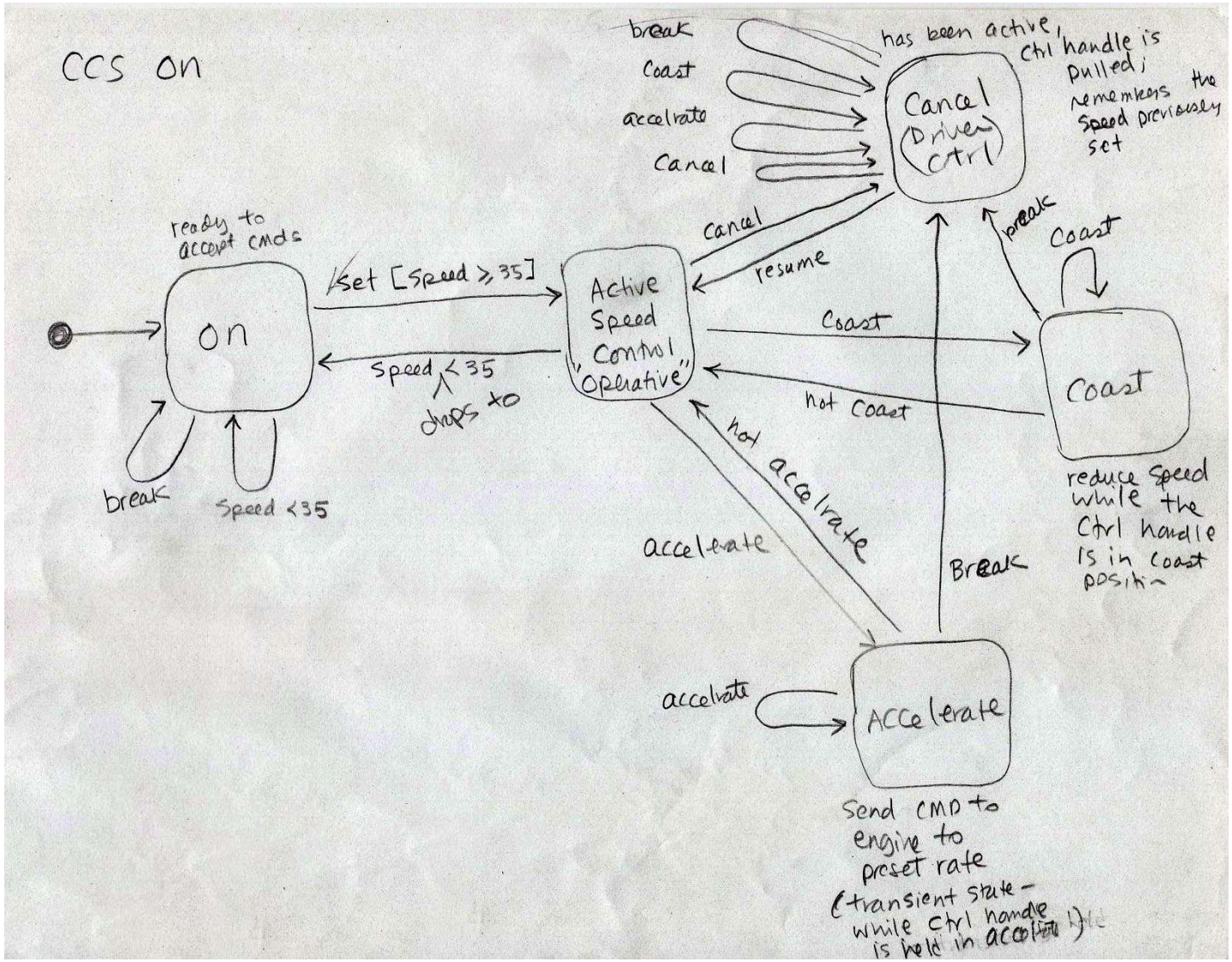
Example: States of a device like a laptop



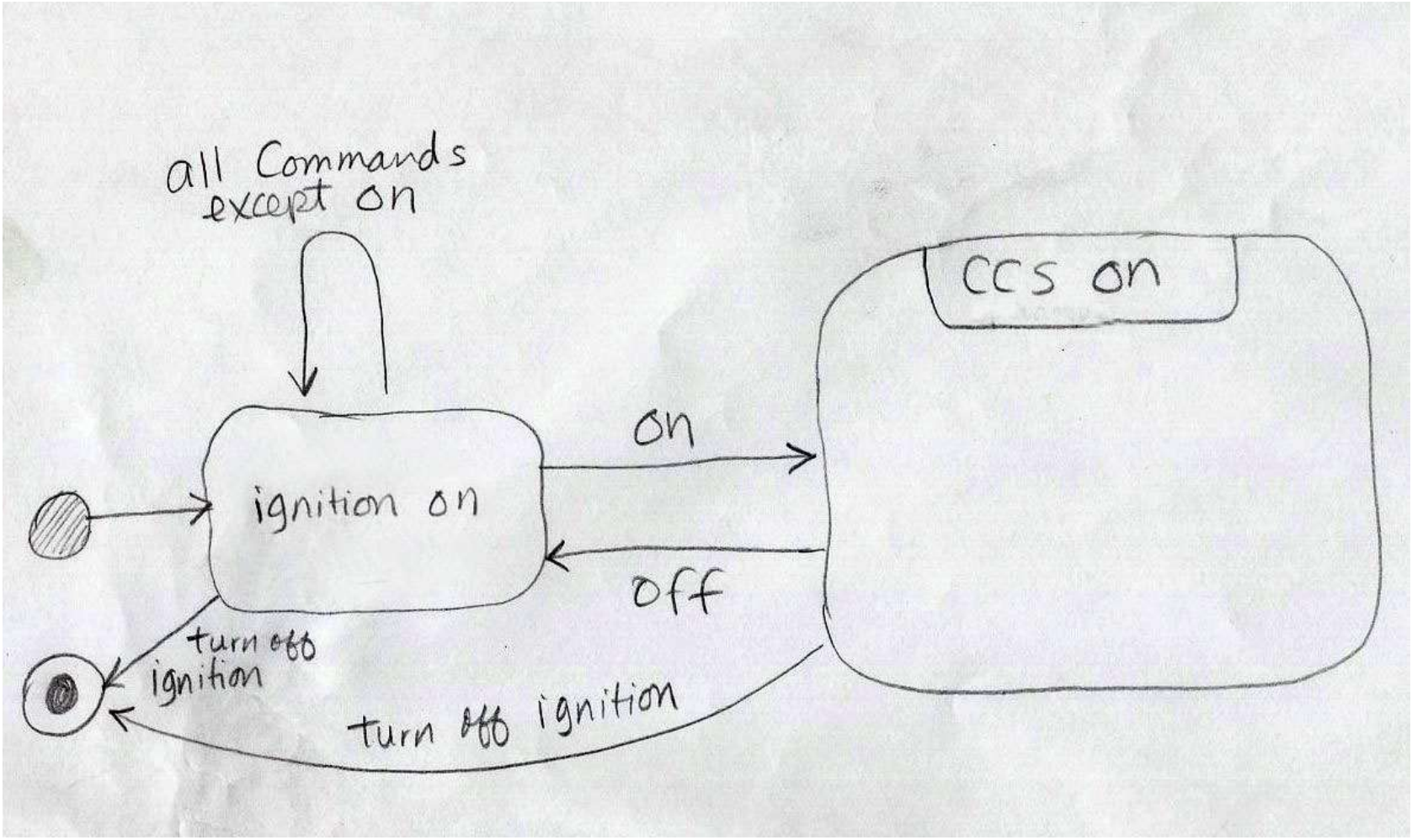
Example: A cruise control system (CCS)



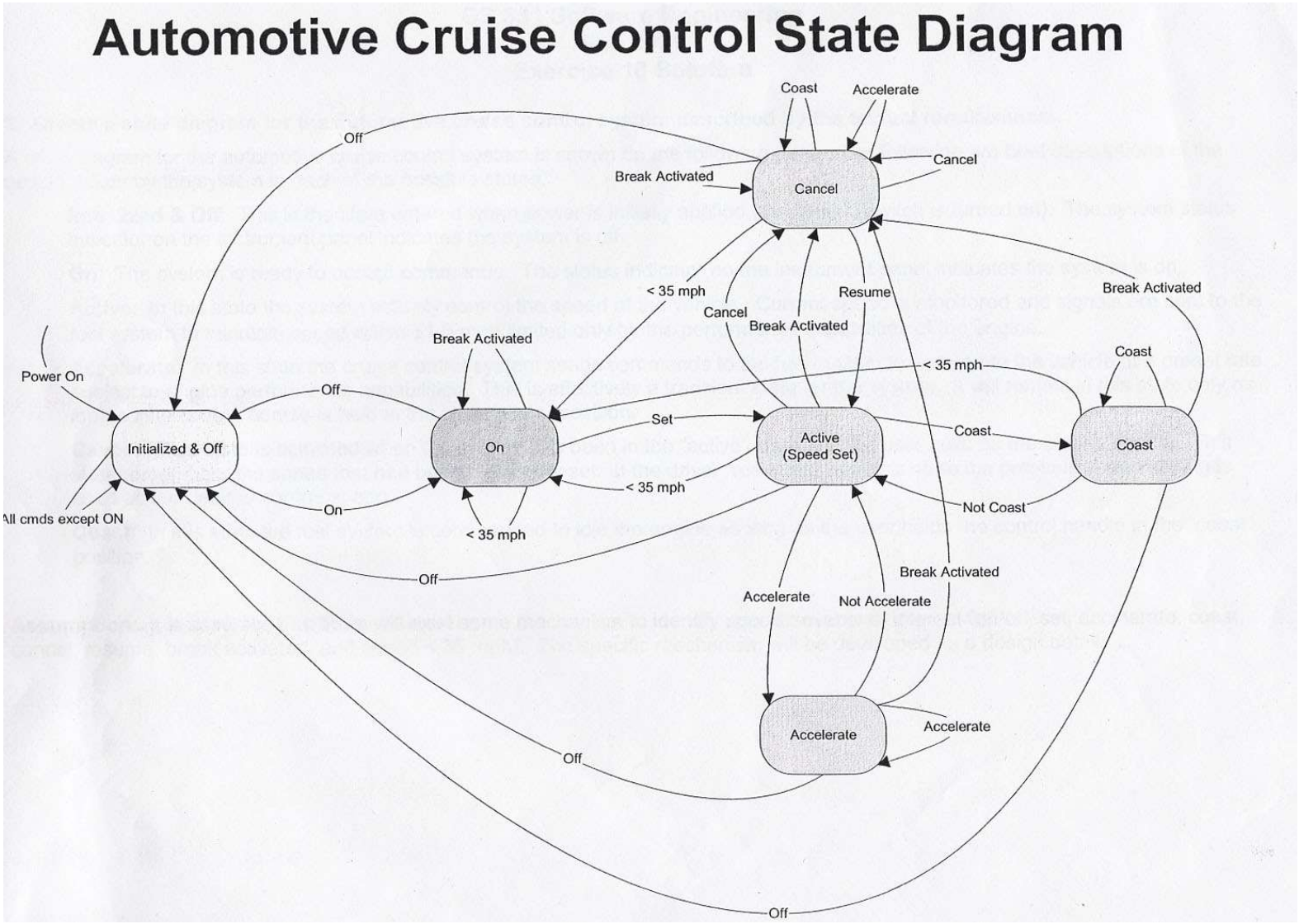
Example: A cruise control system (CCS)



Example: A cruise control system (CCS)



A cruise control system (CCS)



UML state diagrams summary

- UML uses an object-oriented variant of state machines modified for software modeling needs
- Ideal for modeling the behavior of active event-driven objects
- Includes a number of useful features such as entry/exit actions and state activities
- Facilitates hierarchical modeling of complex systems
  - Via the composite states (also called nested states, hierarchical states, or super states)
  - Allow the reader to zoom in or zoom out (especially if graphical tools are used)

References

- IBM Rational UML/UP Course Resources
- Microsoft Online UML Tutorials
- OMG UML Tutorials
- Samek, M., *Practical UML Statecharts in C/C++*, Elsevier, 2009

State machine examples

- A slightly more interesting microwave system (a safety issue: the microwave should not operate when the door is opened)