CSCI 1900
Discrete Structures

Graphs and Finite State Machines
Reading: Kolman,
Sections 8.1 and 10.3

State Machines

- A machine that has input, an internal memory that can keep track of information about the input history, and an optional output is called a ***state machine***. (Paraphrased from textbook, p. 390)
- "The complete internal condition of the [state] machine and all of its memory, at any particular time, is said to constitute the ***state*** of the machine at that time." (Quoted from textbook, p. 390)

More Informal Definition of a State

- The current condition of any machine is defined by certain properties including:
  - the inputs that brought the machine to this state;
  - the current output of the machine; and
  - the response the machine will have to new inputs.
- These conditions are referred to as states.

Finite State Machine

- Assume that we have a finite set of states that a machine can be in, S, and a finite set of possible inputs to that machine, I.
- A machine is a Finite State Machine (FSM) if when any input from the set I is input to the machine causing it to change state, the state it changes to will be contained in the finite set of states, S.

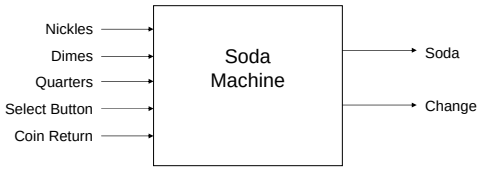
Real World Examples

- All of the math aside, many common items or even non-technical tasks can be modeled with a finite state machine.
- Examples:
  - Soda machine
  - Software applications
  - Spell checker
  - Driving to school/work

Example – Soda Vending Machine

- Assume a soda vending machine that accepts only nickles, dimes, and quarters sells only one kind of soda for 25¢ a piece.
- Identify all of the possible states
  - 0¢: no money at all – waiting for any money
  - 5¢, 10¢, 15¢, and 20¢: some money, but not enough – waiting for coin return or more money.
  - 25¢, 30¢, 35¢, 40¢, and 45¢: enough money – waiting for select button, or coin return button
- Finite states (10) and finite inputs (5¢, 10¢, 25¢, select button and coin return button)

Soda Machine Block Diagram



State Transitions

- A ***state transition*** is the definition of the destination state based on the current state and the system input.
- A state transition must be defined for every input out of every state.
- Exactly one destination state is defined for every allowable input.
- Example: In testing software for quality control, every possible user input from every possible state must be considered even if the input makes no sense. In addition, no user input can take the system to more than one state.

State Transition Functions

- The function, f_x, that defines which state the machine will go to after an input, x, is called the ***state transition function***.
- Denoted $f_x(s)$ where x denotes the input and s denotes the current state.
- Example: If the soda machine is in the state representing 10¢ received, inserting a nickle will move it to the state representing 15¢ received.

$$f_{\text{Nickle}}(10¢) = 15¢$$

Set of State Transitions = Relation

- Since the state transition function defines every transition of an FSM, a set of tuples of the form (s_m, s_n) can be created where s_m is the state where a transition starts and s_n is the state where that transition ends.
- Assume a relation R_M is the set of tuples described above. An FSM then can be defined as the set of all states, $\boldsymbol{S}$, the set of inputs, $\boldsymbol{I}$, and the relation defining the state transitions, R_M.

$$R_M = \{(s_m, \boldsymbol{f}_{s_m}(x)) \mid \forall\, x \in I \text{ and } \forall\, s_m \in \boldsymbol{S}\}$$

State Transistion Table

- A table summarizing all of the state transitions of a finite state machine is called a ***state transition table***.
- Each row of a state transition table represents the current state while the columns for that row show the destination or next states based on different possible inputs.

Soda Machine State Transition Table

Current state	Nickle	Dime	Quarter	Select button	Coin return button
0¢	5¢ [1]	10¢ [1]	25¢ [1]	0¢	0¢ [4]
5¢	10¢ [1]	15¢ [1]	30¢ [1]	5¢	0¢ [4]
10¢	15¢ [1]	20¢ [1]	35¢ [1]	10¢	0¢ [4]
15¢	20¢ [1]	25¢ [1]	40¢ [1]	15¢	0¢ [4]
20¢	25¢ [1]	30¢ [1]	45¢ [1]	20¢	0¢ [4]
25¢	25¢ [2]	25¢ [2]	25¢ [2]	0¢ [3]	0¢ [4]
30¢	30¢ [2]	30¢ [2]	30¢ [2]	0¢ [3]	0¢ [4]
35¢	35¢ [2]	35¢ [2]	35¢ [2]	0¢ [3]	0¢ [4]
40¢	40¢ [2]	40¢ [2]	40¢ [2]	0¢ [3]	0¢ [4]
45¢	45¢ [2]	45¢ [2]	45¢ [2]	0¢ [3]	0¢ [4]

[1] – Accepts inserted money
[2] – Any money inserted is returned
[3] – Soda delivered and change returned
[4] – All inserted money is returned

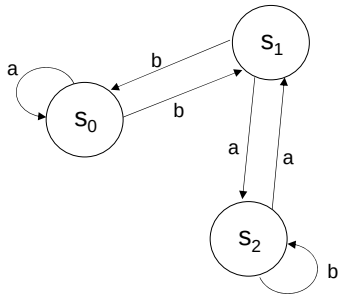
FSM Labeled Digraph

Since a relation R_M can be defined for an FSM using a state transition function, then a digraph can be created to represent the relation. It must be a labeled digraph to indicate which input specifies which transition

Current state	a	b
s_0	s_0	s_1
s_1	s_2	s_0
s_2	s_1	s_2

$R_M = \{(s_0, f_a(s_0) = s_0),$
$(s_0, f_b(s_0) = s_1),$
$(s_1, f_a(s_1) = s_2),$
$(s_1, f_b(s_1) = s_0),$
$(s_2, f_a(s_2) = s_1),$
$(s_2, f_b(s_2) = s_2)\}$

FSM Labeled Digraph (continued)



Soda Machine Digraph

(Select button transitions are missing)