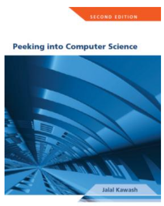
Finite State Machines

Peeking into Computer Science



Reading Assignment

- Mandatory: Chapter 3 – Section 3.6

Finite State Machines

Graphs as Solutions

3

At the end of this section, the student will be able to:

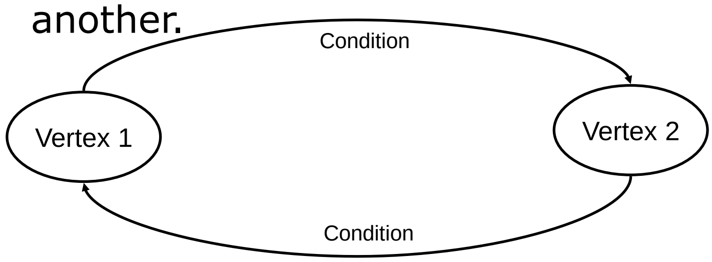
1. Define Finite State Machines (FSMs)
2. Represent FSMs using state-tables and state-diagrams
3. Use FSMs to design High-Level programs
4. Understand FSM examples

Objectives

Peeking into Computer Science

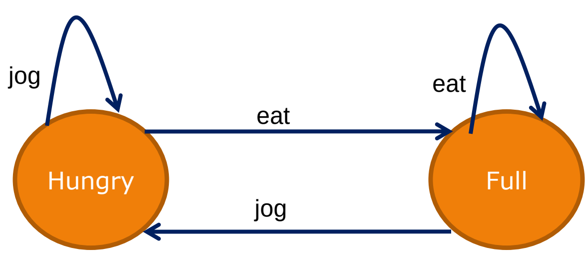
JT's Extra: What Is A Finite State Machine?

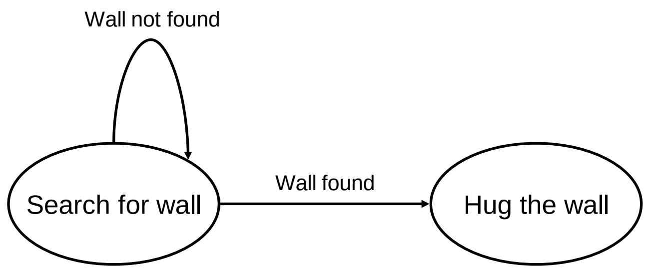
- It's a special form of multi graph (vertices and edges) that have conditions that show how you go from one vertex to another.



Finite State Machines

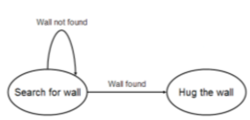
- Vertices are **States**
- Edges are labeled **Transitions**





JT' Extra: First Example: Robot Redux (State Machine)

- A state machine is another level of abstraction that specifies what a computer is supposed to do.
  - Level I: Finite state machine



  - Level II: Human language instructions (pseudo code)

Search for wall
If RS = W, then done this phase
If FS = W, then L, done this phase
If FS = S, then F

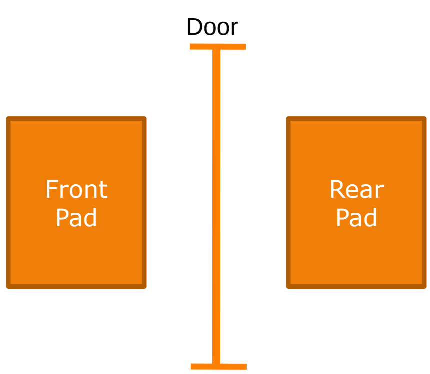
Hug the wall
Repeat the following steps:
If RS = W and FS = S, then F
If FS = W, then L
If RS = S and FS = S, then R and F

JT's Extra: Why Learn About State Machines

◦ Level III: Programming language instructions

```
if (robot.wallToRight () == true)
{
  isDone = true;
  return;
}
```

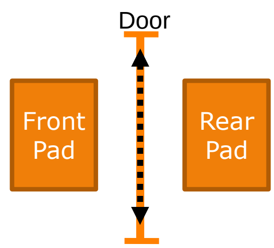
JT's Extra: Why Learn About State Machines



Based on Sipser, *Introduction to the Theory of Computation*, Thomson

Automatic Door Controller

- Identify the door states
- Identify events that trigger transitions
- Version 1 : sliding door



Automatic Door Controller

- Identify the door states
- Identify events that trigger transitions

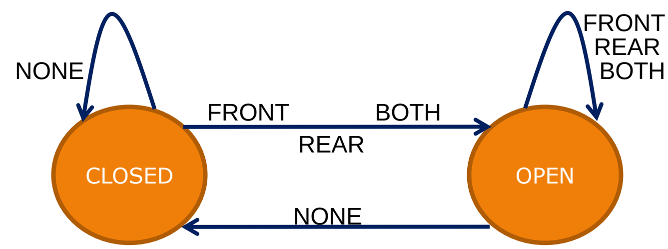
Events

		NONE	FRONT	REAR	BOTH
Door States	CLOSED	CLOSED	OPEN	OPEN	OPEN
	OPEN	CLOSED	OPEN	OPEN	OPEN

FSM – Sliding Door

Events

Door States	NONE	FRONT	REAR	BOTH
CLOSED	CLOSED	OPEN	OPEN	OPEN
OPEN	CLOSED	OPEN	OPEN	OPEN



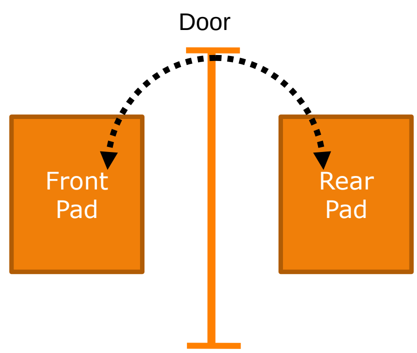
FSM Diagram – Sliding Door

	NONE	FRONT	REAR	BOTH
CLOSED	CLOSED	OPEN	OPEN	OPEN
OPEN	CLOSED	OPEN	OPEN	OPEN

Input	Current State	Next State
NONE	CLOSED	CLOSED
NONE	OPEN	CLOSED
FRONT	CLOSED	OPEN
FRONT	OPEN	OPEN
REAR	CLOSED	OPEN
REAR	OPEN	OPEN
BOTH	CLOSED	OPEN
BOTH	OPEN	OPEN

Alternative Table

- Version 2 : door opens both ways



Door Open Both Ways

- Identify the door states
- Identify events that trigger transitions

Events

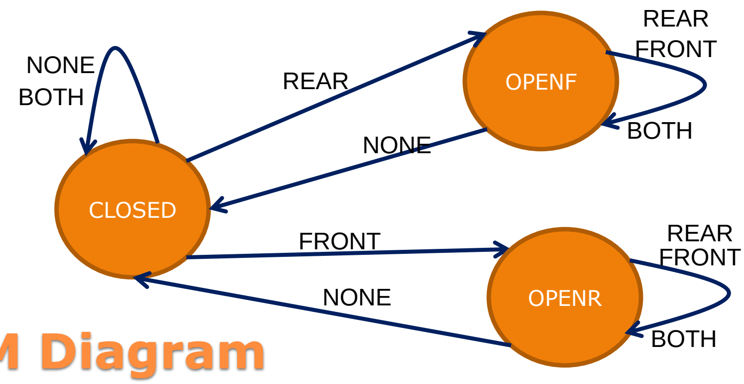
Door States		NONE	FRONT	REAR	BOTH
	CLOSED	CLOSED	OPENR	OPENF	CLOSED
	OPENF	CLOSED	OPENF	OPENF	OPENF
	OPENR	CLOSED	OPENR	OPENR	OPENR

FSM – Door Opens Both Ways

Events

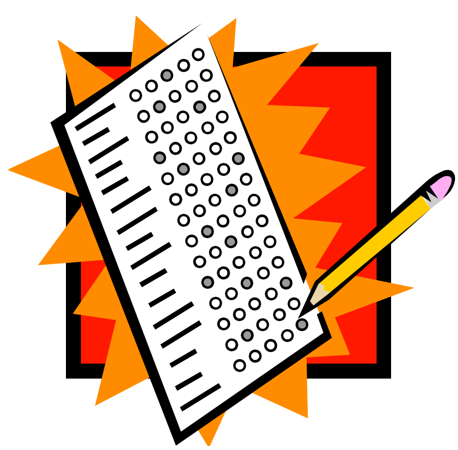
	NONE	FRONT	REAR	BOTH
CLOSED	CLOSED	OPENR	OPENF	CLOSED
OPENF	CLOSED	OPENF	OPENF	OPENF
OPENR	CLOSED	OPENR	OPENR	OPENR

Door States



FSM Diagram

Peeking into Computer Science



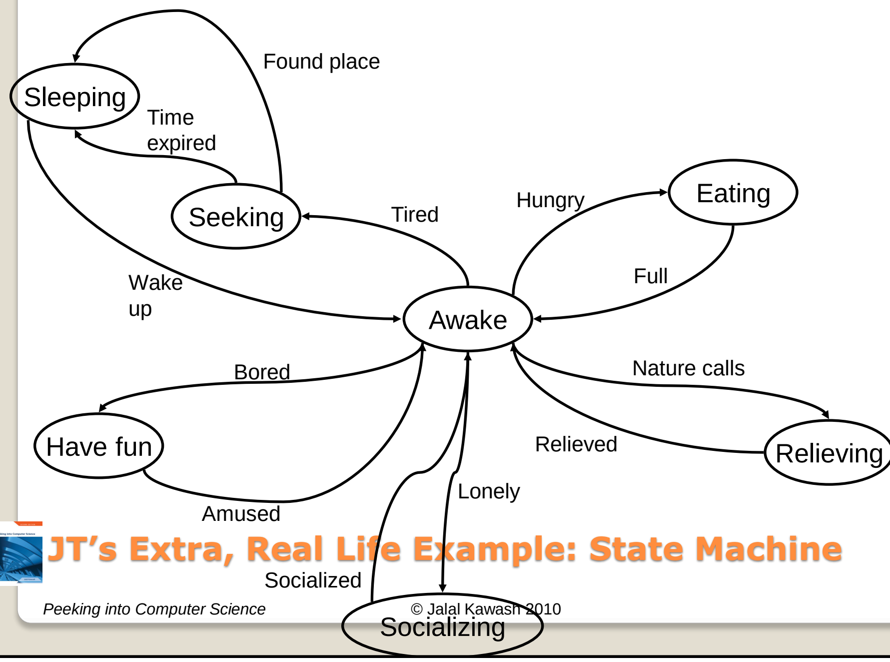
1

- The Sims™ Electronic Arts is a game that simulates the life of one or more humans (a "sim").

JT's Extra: Real Life Example: The Sims © EA

JT's Extra, Real Life Example: Table Of States

Events	States
Tired	Seek (place to sleep)
Found sleep place	Sleep
Seek time expired	Sleep
Wake up	Awake
Hungry	Eating
Full	Return (awake)
Nature calls	Relieving
Relieved	Return (awake)
Lonely	Socializing
Socialized	Return (awake)
Bored	Have fun
Amused	Return (awake)



JT's Extra, Real Life Example: State Machine

Exercise

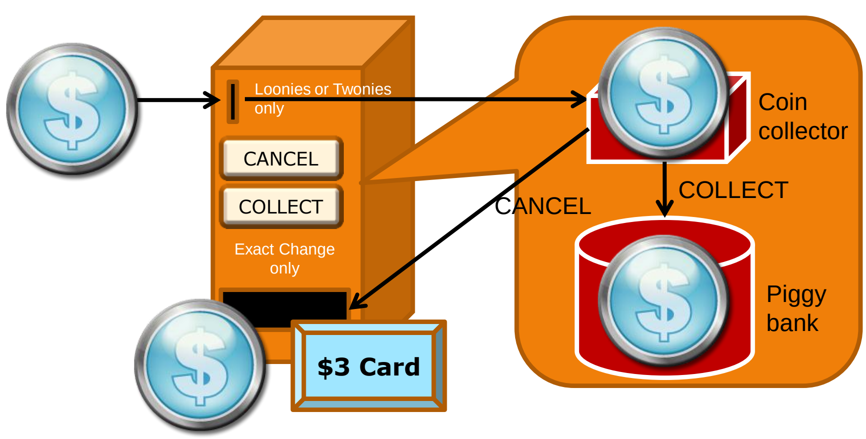
- Design a controller for a garage door
- The door receives input from one remote control only
- It also responds to sensing obstacles

Simple Vending Machine

- Dispenses $3 phone cards
- Accepts $1 or $2 only
- No change
- Keeps coins in *coin collector* until a card is dispensed; then, coins are dropped into the piggy bank

- User
  - Insert coins
  - Press COLLECT to collect
  - Press CANCEL to cancel

That is

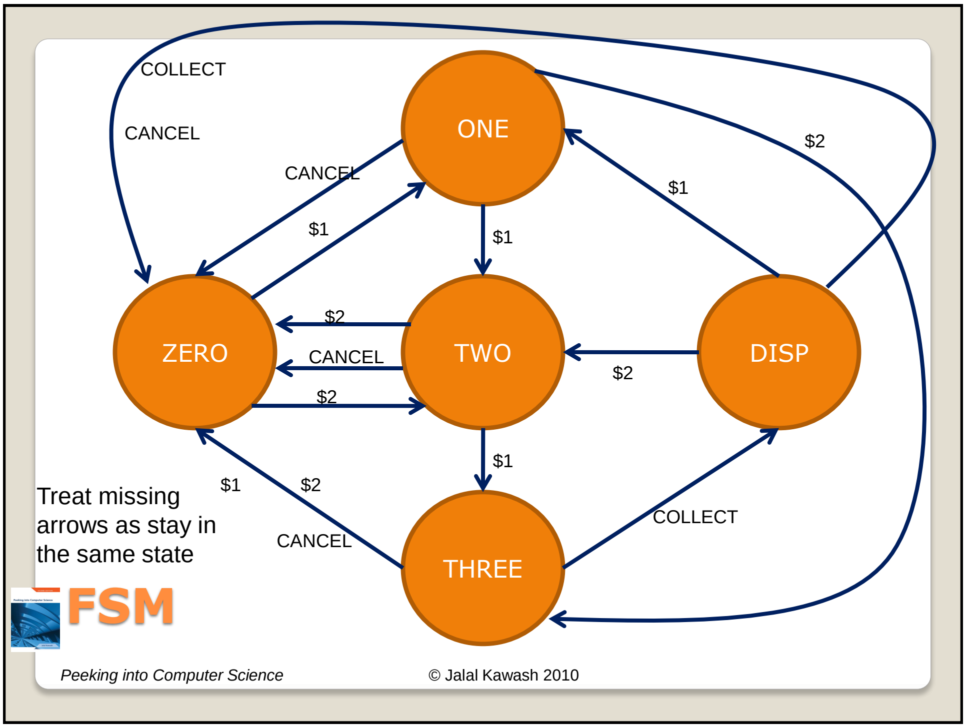


Vending Machine – States

- Need five states:
- ONE: Total in coin collector is $1
- TWO: Total in coin collector is $2
- THREE: Total in coin collector is $3
- DISP: dispenses a card, roll in coins to piggy (coin collector becomes empty)
- ZERO: return coins in coin collector; also serves as a start state
  - Initially coin collector is empty

Vending Machine – Events

- Events/Transition labels:
- $1: user inserts a loonie
- $2: user inserts twonie
- CANCEL: user presses CANCEL
- COLLECT: user presses COLLECT


COLLECT
CANCEL
CANCEL
ONE
$1
$1
$2
$1
ZERO
$2
CANCEL
$2
TWO
$2
DISP
$1
$1
$2
CANCEL
THREE
COLLECT
Treat missing arrows as stay in the same state
FSM
Peeking into Computer Science
© Jalal Kawash 2010