Self-Driving Cars

- 15-Point Checklist
Embedded Systems

- Are sets of tasks
  - The task is the primary abstraction to be used
- If we want to understand the system
  - We need to understand the tasks
  - We need to understand the interactions
    - Between tasks
    - With the environment (real world)
- Should we model the whole system
  - Or should we just model the individual tasks
How to Model a Task

- **English description**
  - Typically used first
  - Imprecise, hard to check, incomplete

- **Formal mathematics**
  - Very precise
  - Difficult to understand and write (correctly)
  - Hard to express external events

- **Formal diagrams**
  - Combine formal math with understanding
  - Diagrams make it easier to comprehend what is there
  - Provide a framework for the math and understanding
Modeling Control Systems

- Control oriented systems
  - Typically respond to external stimuli
- Reaction based on previous interactions
  - These determine the state of the system
- Stimulus moves system to a new state
  - Control is state-oriented
- Natural model is **finite state automata**
  - Also called a finite state machine
Basic Finite State Automata

What language does this recognize?
Problem

- Want to automatically trigger switch
  - Only once if possible (not continually)
  - When train goes from R to L
  - What does sensing look like (at A, B, C, D)
    - Events: A, $\bar{A}$, B, $\bar{B}$, C, $\bar{C}$, D, $\bar{D}$

- When should you trigger the switch
  - What does the train look like to the sensor
Problem

- Can only trigger one switch at a time
  - Switch trigger
    - Should be momentary (>100ms, <500ms)
  - What does this task do?
    - How many tasks should it be?
    - Different approaches
- Is it a FSA?
Problem

Problem: when is the train on which block
  When is train completely on A/C or B or D side of switch

Complexities
  Handling initial conditions
  Handling reversing directions
Issues with Basic FSA

- Need to specify actions
- Semi-external events: timers
- Internal events from other tasks
- Conditional events
- Multiple FSA’s in parallel (multiple tasks)
- Counters and other variables
- State space blowup
Actions: Moore vs Mealy Machine

Mealy Automaton

Moore Automaton

start

0

0/1

1/0

start

0

1/1

0

0

1

1

20
Moore vs. Mealy

- Which is more useful?
- Are they equivalent?
Extended State Machines

- Each arc has condition/action
  - To traverse arc, condition must hold
  - When traversing arc, action is taken
- There are a finite set of (bounded) variables
  - Conditions can reference these
  - Condition can be
    - Event
    - Time out
    - Expression over variables
    - Combination of the two

- Is the result still a finite state machine?
Extended State Machines

- **Actions**
  - Can access and change variables
  - Can be \{ var := expression; ... \}
  - Can do something external

- **When no condition is true**
  - Stay in same state

- **When multiple conditions are true**
  - What should be done
    - Nondeterminism
    - Allow priority specification
    - Disallow
Extended State Machines
Extended State Machines

NIP (NIP not correct) and attempts < 3
/display ("try again");
attempts = attempts + 1;

NIP (NIP not correct) and attempts = 3
/display ("card confiscated");

NIP (NIP is correct)
/display ("OK, please select amount")

amount_entered (request)
[request <= available]
/display ("please take your money");
dispense_money;

money_taken /display ("bye");
eject_card;

(card Entered)
/display ("hello, please enter NIP");
attempts = 0;
Periodic Automata

- Specify a Period
  - Action taken every period
  - Not on timeout
  - Not when events occur

- Extended automata
  - Variables, actions

- Good model for Arduino tasks
Periodic Automata

- Period = 100 ms
Statecharts

- David Harel (80’s)
- Part of UML
- Extended FSM notation
Meta States

- Hierarchical Automata
- Start states
- Stop states
- Enter States
Homework

- Read Chapter 4
- Exercises 4.3, 4.5, 4.7