Overview

⇒ What is the Course About?

• Chapter Introduction
• Finite Automata
• Formal Definition
Expected Outcomes

• Be able to do symbolic proofs

• Understand that there are limitations to computation

• Understand the power of using simpler models of computation

• Understand why heuristics might be needed for some problems

In practical terms,

• Il from a formal standpoint as well

• Just as you might know how computation happens at gate level,

• Operating system level, programming level, good to understand

• Different models, complexity, complexity

• Learn about the theory of computation

Goals of Course

• Automata

  - What is a formal model of computing?
  - Are there different classes of problems?
  - How much time or space do problems take?

• Computability

  - Are there some problems that are not possible to compute?

• Complexity

  - Are there different models with different power?
  - What is a formal model of computation?

• Automata
Homework

- Homework is due Saturday at 11:55pm
- If needed, you can email me with a request for an extension
- Extension for one homework will be automatically granted
- Homework must be submitted via Sakai
- Homework should be typeset
- Upload a single pdf
- Can use tikz package to draw them
- Can include hand-drawn solutions taken with cellphone
- Will need to include diagrams of FSAs and PDAs

Logistics

- Grading: 60% homework, 20% midterm, 20% final
- Course website: cslu.ohsu.edu/~heeman/cs533

Grading:

- 60% homework
- 20% midterm
- 20% final
Do not google the answer
- Nothing in, nothing out. Wait at least an hour between question means.
- e.g., worked on entire question together versus glanced when the question was.
- Document how you collaborated.
- You can do the homeworks with your colleagues.

Academic Integrity

See sample homework for how to format this.
- Understand whether your answer was correct.
- Should show how you reviewed and understood answer key and really good explanations with even get more.
- Explain what you did wrong and why you made that mistake.
- You will have until Sunday at 11:55pm to submit a critique.
- Sakai will give you answer key after you submit your answers.

Critique
Chapter 1: Regular Languages

- What is a computer?
  - Definition gets into all sorts of complicated issues
  - Types of memory
  - How it is addressed
  - What types of operations are supported

- A computational model

Let's focus on an idealized computer

- What is a computer?

Formal Definition
Finite Automata

Overview

What is the course about?
Automatic Door Problem

- If someone steps on front pad and rear pad at the same time, it should not open the door; otherwise it might knock over the person on the rear pad.
- If someone is on the rear pad and door is close, it should keep the door closed.
- If someone is on the front pad, knock the person over.
- If someone is on the front pad and door is close, it should keep the door open until the person is off of the rear pad, so that the automatic door that swings open does not knock over the person on the rear pad.
- Only for people going into store.
- If someone steps on both pads at the same time, it should not open the door, otherwise it might knock over the person on the rear pad.

What can a machine with extremely limited amount of memory do?

Overview

- What is the Course About?
- Chapter Introduction
- Formal Definition
- Finite Automata

© P. Heeman 2017
Many problems can be solved by an FA—e.g., Elevator controller, Thermostat, Dishwasher, ... If problem solvable by an FA, why use something more complicated?

+ Use simplest computational model possible!

So, need to be more formal about this model
- Is capable of
- And not capable of

So, let's understand what this model is capable of
- Use simpler computational model possible?
- If problem solvable by an FA, why use something more complicated?
- E.g., Elevator controller, Thermostat, Dishwasher...

Many problems can be solved by an FA

Finite Automata

A Finite Automata

Transition Table:

If sensor input, it might traverse:
- Is the door open or closed?

So, machine needs two states
Still need a formal definition

- Must there be an accept state? Can there be several?
- Must there be a reject state? Can there be several?
- Must there be a transition for every character from each state?
- Can there be several transitions from a state for the same character?

But, we still need a formal definition

Still need a formal definition

What is an Finite Automata?

- Consider the following FA M

What is an Finite Automata?

- Processes an input
- Transitions: what state it transitions from/to with a certain input
- 3 states: start state, accept state
- 1 state: reject state
- Called a state diagram

- So, for any input it just makes a binary decision:
  - Accept or reject

Example: 1101

- If in an accepting state at end of input, accept, else reject
- As it processes each character, it transitions from state to state

- What are the strings that M accepts?

Need a precise definition for giving proofs about its power!
Formal Definition

A finite automaton is a 5-tuple (Q, Σ, δ, q₀, F), where:

1. Q is a finite set called the states
2. Σ is a finite set called the alphabet
3. δ : Q × Σ → Q is the transition function
4. q₀ ∈ Q is the start state
5. F ⊆ Q is the set of accepting states
If $A$ is the set of all strings that machine $M$ accepts -
We say that $A$ is the language of machine $M$ -
We write $L(M) = A$.

We reserve the word accepts to talk about individual strings: $M$ accepts $w$. $M$ may accept a lot of different strings.

If the machine accepts no strings, it still recognizes one language: $\emptyset$.

For our example:

Let $A = \{ w | w$ contains at least one 1 and an even number of 0s following the last 1 $\}$.

$L(M) = A$ or $M$ recognizes $A$.

Example

What is a formal definition of $M$?
• Definition of $M_2$:

• Language of $M_2$:

• Definition of $M_3$:

• Language of $M_3$:

Yet Another Example

Another Example
Formal Definition of Computation

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a finite automaton. Let $w = w_1w_2...w_n$ be a string over $\Sigma$. We say that $M$ accepts $w$ if there is a sequence of states $r_0, ..., r_n$ such that:

1. $r_0 = q_0$
2. $r_{i+1} = \delta(r_i, w_{i+1})$ for $i = 0, ..., n-1$
3. $r_n \in F$

Back to language of an FA

We say that $M$ recognizes language $A$ if $A = \{w | M$ accepts $w\}$.

Design a FA that takes a sequence of digits (0-5) that has the property that sum of digits is evenly divisible by 3.
Designing Finite Automata

- Think of what states you need to represent the important part of what you have seen in the input so far.

- Think of how you can transition between states when you have seen in the input so far.

- Think of what states you need to represent the important part of.

### Designing Finite Automata

**Example: Language of \{good, bad\}**

**Example:** Language in which all strings contain the substring 01.

**Example:** Language of all strings with an odd number of 1s.

**Examples:**

- Language in which all strings contain the substring 01.
- Language of all strings with an odd number of 1s.

### Regular Language

**Definition:** A language is called a regular language if some finite automaton recognizes it.

- What it answers, accept to:

- Defining the computation that the automaton does:

**Note that a language defines what the automaton accepts, so:**

**Definition:** A language is called a regular language if some finite automaton recognizes it.

**Definition:** A language is called a regular language if some finite automaton recognizes it.