COMP335: Midterm

Slides: 335_Intro_01

- What Automata
- Different Kinds of Automata
- PROOF TECHNIQUES
  - Proof by induction
  - Proof by contradiction

Slides: 335-Languages

- What are languages
- Alphabets and Strings
- Operation on Strings
- Operations On languages

Slides: 335_FA

- What is a Finite Automata
- Transition Graph for Deterministic FA
- Formalities for DFA: $M = (Q, \Sigma, \delta, q_0, F)$
- Recursive definition for transition function
- When the language is accepted by DFA
- How to design a DFA for a language
- What and when the language is Regular

Slides: FA_Cont

- What is a Non deterministic Finite Automata (NFA)
- Formal Definition of NFAs
- When two FA are equivalent
- Languages accepted by NFA = Regular Languages
- How to convert and NFA to DFA

Slides: RL&RE

- How to convert NFA with more than one finite state to single finite state.
- Properties of regular languages
- Regular expression and regular languages
- Languages Generated by Regular Expressions= regular Languages
- any regular expression is a regular language
- Reducing the number of states: reducing regular expression

Slides: RL_Cont

- Standard Representations of Regular Languages:
    - DFA
    - NFA
    - Regular Expression
    - Regular Grammar

Q: Given any regular language L and any string w in the alphabet, how can we check if $w \in L$

A: Take a DFA that accepts L and check if w is accepted .

Q: Given any regular language L, how can to check if $L = \varnothing$? That is, is L empty?

A: Take a DFA M that accepts L Then check if there is any path from the initial state to a final state.

Q: Given a regular language L, how can we check if it is finite?

A: Take a DFA that accepts L Then check if there is a walk with cycle from the initial state to a final state

Q: Given regular languages L_1 and L_2, how to decide if $L_1 = L_2$ **?**

A: Check if : $(L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2) = \varnothing$

- How to show that a language is not regular?
    - The Pumping Lemma (proof by contradiction)

- Given an infinite regular language L
- there exists an integer m
- for any string $w \in L$ with length $|w| \geq m$
- we can write $w = x\, y\, z$
- with $|x\, y| \leq m$ and $|y| \geq 1$
- such that: $w_i = x\, y^i\, z \in L, \qquad i = 0, 1, 2, \ldots$

Slides: CFG

- What is a context free Grammar
- Leftmost derivation
- Right most derivation
- Derivation Tree
- Parsing a string is an exhaustive search problem
- Time complexity of the exhaustive search
- What is an ambiguous Grammar.

Text Book: From Chapter 1~Chapter 5 included,

Tutorials,

Assignment 1 and Assignment 2.