Computer Science 311 Course Objectives

Ryan Bernstein

By the end of this course, students should be able to:

0 Review

- Use informal proof strategies to write out natural-language arguments
- Create and modify sets using standard notation and operations such as union, intersection, and complement

1 Regular Languages

- Discuss alphabets, strings, and languages in terms of their underlying sets
- Create finite automata (deterministic and nondeterministic) that decide given regular languages, using:
  - State diagrams and
  - a formal definition
- Identify the languages decided by given finite automata
- Create regular expressions that describe regular languages
- Identify the languages described by given regular expressions
- Intelligently discuss the concept of nondeterminism as it pertains to abstract machines
- Use constructive proofs to describe methods for creating expressions or abstract machines for languages whose definitions include one or more variables
- Use the pumping lemma to show that a given language is not regular
- Discuss (and prove) the closure properties of regular languages under arbitrary operations

2 Context-Free Languages

- Show that a given language is context-free using pushdown automata or context-free grammars

- Discuss pushdown automata and context-free grammars conceptually, using their formal definitions where appropriate
- Discuss the closure properties of context-free grammars under arbitrary operations
- Show that a given language is not context-free (if applicable) using the context-free pumping lemma

3 Turing Machines and Decidability

- Discuss the entire Chomsky hierarchy of languages as we have defined it, including:
  - Finite languages
  - Regular languages
  - Context-free languages
  - Turing-decidable languages
  - Turing-recognizable languages
  - Turing-unrecognizable languages
- Create Turing machines that decide or recognize a given language using high-level, natural language descriptions
- Discuss the properties of Turing machines, using their formal definition as a 7-tuple where appropriate
- Use mapping reductions to show the relationships between language difficulties
- Show that a given language is:
  - Decidable
  - Recognizable
  - Unrecognizable
- Discuss (and prove) closure properties of decidable and recognizable languages under arbitrary operations
- Show equivalence between Turing machine variants, or identify the languages that can be decided by a variant that is not equivalent to a standard Turing machine
- Use algorithms discussed in previous sections as subroutines in the creation of Turing machines
- Operate on encoded objects without knowing the details of the encoding in question

4 Time Complexity

- Find the runtime of a given Turing machine (deterministic or nondeterministic)
- Find upper bounds for the time complexity classes of decision problems
- Show that a problem is in the class P or the class NP

- Discuss the significance of poly-time reductions as they pertain to relative problem difficulties
- Define and discuss the problem classes:
  - P
  - NP
  - NP-Hard
  - NP-Complete

  Discuss what is currently known about the relationships between these classes