CS 252, Sections 1 & 2, Winter 2016
Introduction to Computational Theory
Final Exam Review

- **Chapter 15. P and NP**
  - Time complexity of DTMs/NTMs
  - (In)Tractability on *decidable* problems: polynomially bounded DTMs (NTMs)
  - Definitions of **P** and **NP**
  - Problem representation & (polynomial-time) reduction (on *decidable/tractable* problems)
  - Problems in **P** and **NP** classes: P, NP, NP-*hard*, and NP-*complete*
  - **P** = ($\neq$) **NP**, complexity class relations
  - The Satisfiability Problem

- **Chapter 14. Time Complexity**
  - Time and space complexity: number of moves versus number of storage space required
  - Time complexity: worst-case performance, complexity analysis of the complexity of a TM
  - Rates of growth: constant terms, linear terms, lower-order terms, most significant contributors, Big O Hierarchy

- **Chapters 11 & 12. Decision Problems & Undecidability**
  - Decision problems: Yes/No answers, (un)decidable/(un)solvable problems, Church-Turing Thesis, Turing computable functions
  - *Halting problems* for TMs: no algorithm that solves the halting problem
  - *Recursively enumerable* vs. *recursive* languages
  - The Universal (Turing) Machine

- **Chapter 10. Chomsky Hierarchy**
  - Unrestricted grammars (Type 0)/TMs, context-sensitive grammars (Type 1)/LBAs, context-free grammars (Type 2)/PDAs, regular grammars (Type 3)/FSAs

- **Chapter 8. Turing Machines (TMs)**
  - TMs: state, tape alphabet, input alphabet, transition, transition function
  - Machine operations: read, write, move (L/R), stay (S), halt
  - TMs and unrestricted grammars (recursively enumerable languages)
  - TMs as language acceptors: acceptance by *final state*, acceptance by *halting*, normal (abnormal) termination (halting)
  - Types of TMs: deterministic, non-deterministic, k-tape TMs

- **Chapter 19. LL(k) Grammars**
  - Parsing: left-to-right/left-most derivation, deterministic top-down parsing, lookahead principle

  - Lookahead (LA) sets: lookahead strings, $LA_k(A)$, $LA_k(A \to w)$, FIRST/FOLLOW sets
  - Lookahead (LA) set and CFLs: Construction of $FIRST_k$ Sets (Algorithm 19.4.1), Construction of $FOLLOW_k$ Sets (Algorithm 19.5.1), Construction of LA_k Sets

- **Chapter 7. Push-Down Automata and CFLs**
  - PDAs: states, input symbols, stack symbols, transitions, control, (non-)deterministic transition functions
  - Transitions: atomic, regular, extended
  - PDAs and CFLs: $L(M)$, acceptance by *final state*, acceptance by *empty stack*, transformations between PDAs and CFLs
  - Two-stack PDAs: an extension of PDAs, non-CFLs

- **Chapter 3. Context-Free Grammars**
  - CFGs: Context-free grammar rules
  - Derivation: sentential form, leftmost/rightmost derivation, derivation/parse tree
  - CFLs and Chomsky/Greibach Normal Form
  - *Ambiguous* and *inherently ambiguous* grammars

- **Chapter 6. Properties of Regular Languages**
  - Regular grammars, expressions, and sets
  - Expression graphs for generating regular expressions (Algorithm 6.2.2)
  - Transformation between regular grammars, regular expressions, regular sets, and finite automata
  - Pumping lemma: non-regular languages

- **Chapter 5. Finite Automata**
  - Finite state automata: states, input symbols, control, transitions, state (transition) diagram, state (transition) table
  - Different FSAs: deterministic (DFA), non-deterministic (NFA)
  - Transformations: elimination of λ-transitions (λ-closure), removing non-determinism, converting NFA(λ-)s to DFAs (Algorithm 5.6.3)

- **Chapter 2. Languages**
  - Strings: grammars & languages
  - Regular languages: regular sets, regular expressions
  - Different types of grammars and accepting machines

Breakdown on Chapters

1. Midterm Exams (2): Chapters 2, 3, 5, 6, 7, 8, 10, and 19 ($\sim$40%)
2. Chapters not covered in midterm exams: Chapters 11, 12, 14, and 15 ($\sim$60%)