CSCI 1323 (Discrete Structures), Spring 2005

Review for Exam 1

1 Format of the exam

The exam will be in class March 23. You will have 50 minutes. You may use your textbook and any notes or papers you care to bring, but you may not use other books, a calculator or computer, or (of course!) each other's papers.

Most questions will be similar in form to those in the quizzes and homework assignments.

2 Lecture topics to review

You are responsible for all material covered in class or in the assigned reading. (See Lecture Topics and Assignments[1] for a list of assigned reading.) You should review in particular the following topics. This list is *not necessarily exhaustive*, but should give you an idea of what topics I think are most significant.

- Propositional logic:
  - Translating English into propositional-logic wffs (emphasizing understanding of propositional logic connectives over ability to untangle complicated English).
  - Proving that a propositional-logic wff is a tautology using truth tables.
  - Proving that a propositional-logic wff is a tautology using proof rules.
- Predicate logic (propositional logic plus quantifiers):
  - Translating English into predicate-logic wffs (emphasizing understanding of quantifiers over ability to untangle complicated English).
  - Determining whether a predicate-logic wff is true in a given interpretation.
  - Proving that a predicate-logic wff is valid using proof rules.
- Proofs of program correctness:
  - Rules for assignment, conditional statements, loops.
  - Combining these rules to verify correctness of simple programs.
  - Meaning of Hoare triples.
- Proof techniques:
  - Direct proofs, proof by cases, proof by contraposition, proof by contradiction.
  - Proofs by induction.
- Recursion and recurrence relations:

[1] `http://www.cs.trinity.edu/~bmassing/Classes/CS1323_2005spring/HTML/schedule.html`

  - Recursive definitions of sequences, sets, operations, and algorithms.
  - Defining and solving recurrence relations.

- Analysis of algorithms:
  - Defining and solving recurrence relations to estimate the number of basic operations performed by a recursive algorithm.