Course Number: CSC 520
Course Title: Theory of Computing
Number of Credits: 3
Schedule: Three hours of lecture/discussion per week
Prerequisites: A grade of C or better in CSC 213 or CSC 220, MATH 325, and CSC 230 or CSC 330.

Catalog Description:

Automata, formal languages, and the notion of computability. Sequential machines as language acceptors. Context free and context sensitive grammars. Computable and recursive functions, universal Turing machines. Unsolvable problems.

Expanded Description:

Alphabets, Strings, Languages

- Operations on Strings – concatenation, reversal, substrings, prefix, suffix
- Operations on Languages – union, intersection, complement, concatenation, reversal, Kleene star

Finite State Automata

- Deterministic and nondeterministic finite automata
- Conversion of nondeterministic automata to deterministic equivalents
- Minimization of deterministic finite automata
- Combining finite automata – union, intersection, complementation, concatenation, Kleene star
- Regular (representing) expressions
  - Construction of finite automata from regular expressions
  - Determining regular expressions from finite automata
- Regular language Pumping Theorem for proving languages nonregular
- Deterministic finite automata and equivalence relations
- Closure properties of regular languages
- Decidability properties of finite state automata

Pushdown Automata and Context Free Grammars

- Deterministic and nondeterministic pushdown automata
- Context Free grammars
  - Generation of strings and languages
  - Derivations and derivation (parse) trees
  - Right linear and left linear grammars and regular languages
- Equivalence of Pushdown Automata and Context Free Grammars
  - Construction of nondeterministic pushdown automata from context free grammars
- Nonequivalence of deterministic and nondeterministic pushdown automata
- Closure properties of context free languages
- Decidability properties of context free grammars and nondeterministic pushdown automata
- Context free Pumping Theorem for proving languages not context free

Growth rates of context free languages – Parikh's Theorem

Turing Machines

Deterministic Turing Machines

Acceptance of languages vs. deciding languages

Turing machines as computers of functions

Equivalent formulations of Turing machines

The Universal Turing machine

Recursive and recursively enumerable languages

Unsolvability of the Halting Problem for Turing machines

Other unsolvable problems – reductions to the Halting Problem

Closure properties of the recursive and recursively enumerable languages

Unrestricted grammars – generation of recursively enumerable languages

Course Objectives:

Introduce students to the theory of computation through formal languages and abstract computing devices (automata)

Relate the results of computation theory to real-world computational problems

Sharpen students' logical and mathematical skills as they apply to computation in general

Prepare students for much of the material that appears on the advanced subject exam in Computer Science for the GRE.

Method of Evaluation:

Students will be evaluated on the basis of their performance on

Two midterm exams

A comprehensive final exam

Required Textbook:

An Introduction to Formal Languages and Automata, *4th ed*., Peter Linz; Jones and Bartlett, 2006.

Prepared by: Robert Wall
Last Revision: June 2, 2010