TCSS 440 Master Syllabus
Version: March 2012
(Approved: May 11, 2012)

Catalog Description

Covers languages, finite automata, regular expressions, context-free grammars, and other automata such as pushdown store machines and Turing machines. Includes models of computation, computable and non-computable functions, nondeterminism, space and time complexity, tractable and intractable functions. Prerequisite: a minimum grade of 2.0 in TCSS 343.

Preconditions

- Recognize and use mathematical formalisms (e.g., sets, logic, summations, proof).
- Translate problem descriptions into mathematical formalisms.
- Manipulate (procedural knowledge) and apply mathematical formalisms to solve problems.

Student Learning Goals (to be added to syllabus handed out to students)

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

- Apply and use different mathematical abstractions to model computational phenomena.
- Recognize and apply theorems and procedural knowledge to prove properties of classes of formal languages.

CSS Degree Student Learning Outcomes that this course contributes to (to be added to syllabus handed out to students)

a. an ability to apply knowledge of computing and mathematics appropriate to the discipline;
b. an ability to analyze a problem, identify and define the computing requirements appropriate to its solution

UWT Student Learning Goals that this course contributes to (to be added to syllabus handed out to students)

Inquiry and Critical Thinking
Students will acquire skills and familiarity with modes of inquiry and examination from diverse disciplinary perspectives, enabling them to access, interpret, analyze, quantitatively reason, and synthesize information critically.

Communication/Self-Expression
Students will gain experience with oral, written, symbolic and artistic forms of communication and the ability to communicate with diverse audiences. They will also have the opportunity to increase their understanding of communication through collaboration with others to solve problems or advance knowledge.

Topics covered

1. Finite automata and regular languages
    a. Deterministic finite automata
    b. Nondeterministic finite automata
    c. Subset construction
    d. Regular expressions
    e. Equivalence of models
    f. Pumping lemma for regular languages
2. Context-free languages
    a. Context-free grammars
    b. Pushdown automata
    c. Equivalence of models
    d. Chomsky normal form
    e. Pumping lemma for context-free languages
3. Computability
    a. Turing machines
    b. Variations of Turing machines (multi-head, multi-tape, nondeterministic)
    c. Turing-recognizable languages
    d. Turing-decidable languages
    e. Diagonalization
    f. Reducibility
    g. P, NP, NP-completeness