Master Course Syllabus
School of Engineering and Computer Science
Washington State University Vancouver
CS 317
Automata and Formal Languages
3 Semester Hours
(3 lecture hours)

Catalog Description

Finite automata, regular sets, pushdown automata, context-free languages, Turing machines and the halting problem.

Prerequisite Courses

- CS 122 with a C or better
- CS 166 with a C or better, or MATH 301 with a C or better

Prerequisite Topics

- Sets, sequences, relations, functions, graphs, and mathematical proofs.
- Proficient in at least one high level programming language.
- Implementation of common data structures for lists, trees, and graphs.

Measured Course Outcomes

Students taking this course will:

1. Construct mathematical proofs that demonstrate certain formal languages are (or are not) regular and/or context-free (Contributes to performance criterion 6-a.)
2. Construct regular expressions and their corresponding finite automata for string pattern matching (Contributes to performance criterion 6-d.)

Required Textbooks

- *Automata and Formal Languages*, by Michael Sipser, Course Technology.

Reference Material

None

Major Topics Covered in the Course

1. Deterministic and nondeterministic finite automata, pushdown automata, Turing machines, and other theoretical models of computation.
2. Regular, context-free, and Turing-complete languages.
3. Computability and the Church/Turing Thesis (halting problem for Turing-complete languages).
4. Regular expressions and pattern matching.

Projects

Programming Project Area	Weeks
(None)	

Design, Implementation and Analysis

1. Classic theoretical models are used to explore and classify what we know about the process of computation. This involves constructing algorithms for theoretical machines (finite automata, pushdown automata, Turing machines) that determine if a given input string is a member of some formally described language. Nondeterministic models are also formally described and analyzed.
2. The relationship between finite automata, regular expressions, and regular languages is analyzed.
3. The relationship between pushdown automata, context-free grammars, and context-free languages is analyzed.
4. Mathematical proof techniques are developed for demonstrating that certain formal languages are, or are not, regular and/or context-free.
5. The limits of computation are explored via the halting problem and the Church/Turing Thesis.

CS2013

This course provides coverage of CS2013 knowledge areas. Values listed are minimum course hours dedicated to the topic, percentages indicate the fraction of CS2013 knowledge area topics covered (acceptable values are: <25%, 25-75%, >75%, or 100%).

Area	Tier 1	Tier 2	Elective
AL/Basic Automata Computability and Complexity	6 (100%)	6 (25-75%)	
AL/Advanced Automata Theory and Computability			8 (25-75%)
DS/Sets, Relations, and Functions	2 (<25%)		
DS/Proof Techniques	2 (<25%)		

Course Coordinator:	Ben McCamish
Last Updated:	January 24, 2022
Syllabus Version Number:	2.2