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 Data Structures
CS 216 Discrete Structures

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 (among other things):

1. Construct mathematical proofs that demonstrate certain formal languages are (or are not) regular and/or context-free. *(Contributes to performance criterion **J-2**).*
2. Create mathematical descriptions of deterministic and nondeterministic finite automata, pushdown automata, and Turing machine programs. *(Contributes to performance criterion **A-2**).*
3. Construct regular expressions and their corresponding finite automata for string pattern matching. *(Contributes to performance criterion **K-3**).*
4. Construct parsers for simple context free languages. *(Contributes to performance criterion **K-3**).*

Required Textbooks

Automata and Computability by Dexter C. Kozen, Springer-Verlag, ©1997, ISBN 0-387-94907-0

Reference Material

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.
5. Parsing techniques for simple context-free languages.

Laboratory Projects

None Specified

CSAB Category Content

	FUNDAMENTAL	ADVANCED		FUNDAMENTAL	ADVANCED
Data Structures	0	0	Computer Organization and Architecture	0	0
Algorithm & Software Design	0	0	Concepts of Programming Languages	2	1

Oral and Written Communications

None Specified

Social and Ethical Issues

None Specified

Theoretical Content

Topic	Hours
Formal languages (regular, context-free, context-sensitive, Turing-complete).	12
Theoretical models of computation (finite automata, pushdown automata, Turing machines; determinism, nondeterminism).	12
Computability.	6
Regular expressions and pattern matching.	3
Parsing context-free languages.	3

Problem Analysis

1. Classic theoretical models are used to explore and classify what we know about the process of computation. Mostly 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 programs 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.

Solution Design

1. Methods for string pattern matching using regular expressions and finite automata are described.
2. Given a context-free grammar, several techniques for parsing various types of context-free are studied.

CC2001

This course provides coverage of topics in the following areas (hours listed are minimums):

DS1. Functions, relations, and sets [core]	3
DS3. Proof techniques [core]	6
AL2. Algorithmic strategies [core]	3
AL5. Basic computability [core]	6
AL7. Automata theory [elective]	12

Course Coordinator: Wayne O. Cochran
Last Updated: May 11, 2006 (Approved)
Syllabus Version Number: 1.1