

COURSE DESCRIPTION

Department and Course **CS 350** Course Coordinator **Sprague**
 Number

Course Title **Automata and Formal Language Theory** Total Credits **3**

Current Catalog Description

Finite-state automata and regular expressions, context-free grammars and pushdown automata, Turing machines, computability and decidability, and complexity classes.

Textbooks

Introduction to Automata Theory, Languages, and Computation, 2nd ed., by Hopcroft, Motwani, and Ullman, Addison-Wesley, 2001.

References *None*

Course Goals

The goal of this class is for students to become knowledgeable about restricted models of computation (regular languages and context free languages), computability, and the theory of NP-completeness.

Prerequisites by Topic

Discrete Structures, Object-Oriented Design, and Calculus I, all with "C" or better

Major Topics Covered in the Course

Deterministic and nondeterministic automata, regular expressions, regular languages. Context free grammars, pushdown automata, context free languages. Equivalence of definitions for the above classes. Turing machines, recursive languages, recursively enumerable languages. Computability. P, NP, NP-completeness; the notion of intractability.

Laboratory projects (specify number of weeks on each)

None

Estimate CSAB Category Content

	CORE	ADVANCED		CORE	ADVANCED
Data Structures	_____	_____	Computer Organization and Architecture	_____	_____
Algorithms Software Design	40	_____	Concepts of Programming Languages	_____	_____

Oral and Written Communications

None

Social and Ethical Issues

None

Theoretical Content

All the topics covered in the course may be considered theoretical. See the response to “Major topics covered in the course”.

Problem Analysis

Assignments emphasize problem solving, and design.

Examples of problem solving:

Prove that (a particular language L) is not regular.

They know that, given a DFA to recognize language L , it is easy to construct a DFA for the complement of L . The same trick does not work for NFAs. An assignment problem is to find two words x and y such that x is accepted by both a given NFA and its “pseudo-complement”, and y is rejected by both machines.

Solution Design

Assignments emphasize problem solving, and design.

Examples of design:

Design an automaton for a specific language L .

Design a context free grammar for a specific language L .