

CS 3350 Automata, Computability, and Formal Languages Fall 2020 Syllabus

Class Time: Mondays and Wednesdays 9-10:20 am, CCSB 1.0202

General Prerequisites: CS 2302 "Data Structures" and either Discrete Mathematics or Discrete Structures, both with grades C or higher.

Alternative Prerequisites: CS 2401 "Elementary Data Structures and Algorithms" and either Discrete Mathematics or Discrete Structures, both with grades B or higher.

Instructor: [Vladik Kreinovich](#), email vladik@utep.edu, office CCSB 3.0404, office phone (915) 747-6951.

- The instructor's office hours are Mondays and Wednesdays 10:30-11:30 am, 12:30-1:30 pm, or by appointment.
- If you want to come during the scheduled office hours, there is no need to schedule an appointment.
- If you cannot come during the instructor's scheduled office hours, please schedule an appointment in the following way:
 - use the instructor's appointments page <http://www.cs.utep.edu/vladik/appointments.html> to find the time when the instructor is not busy (i.e., when he has no other appointments), and
 - send him an email, to vladik@utep.edu, indicating the day and time that you would like to meet.

He will then send a reply email, usually confirming that he is available at this time, and he will place the meeting with you on his schedule.

Teaching Assistant (TA): TBA.

Instructor of Another Section of Automata: [Luc Longpre](#), email longpre@utep.edu, office hours TBA or by appointments; go to <https://www.utep.edu/cs/people/longpre.html>, select "Student appointments" for instructions on how to make appointments at other times.

Course Objectives: Theoretical computing models and the formal languages they characterize: Finite state machines, regular expressions, pushdown automata, context-free grammars, Turing machines and computability. Capabilities and limitations of each model, and applications including lexical analysis and parsing.

Major Topics Covered in the Course

- Regular languages, finite automata (FA), non deterministic FA (NFA)
- Context-free languages, pushdown automata (PDA)
- Parsing, normal forms, ambiguity
- Pumping lemmas and closure properties
- Turing machines and other equivalent models
- Decidable languages, non-decidable languages, recognizable languages, Chomsky hierarchy

Learning Outcomes

Level 1: Knowledge and Comprehension

Level 1 outcomes are those in which the student has been exposed to the terms and concepts at a basic level and can supply basic definitions. The material has been presented only at a superficial level.

Upon successful completion of this course, students will be able to:

- 1a. Describe implications of Church-Turing thesis.
- 1b. Describe problems for which an algorithm exists, and problems for which there are no algorithms (non-recursive, non-recursively enumerable languages) and describe the implications of such results.
- 1c. Describe and explain the diagonalization process as used in proofs about computability.
- 1d. Describe the difference between feasible and non-feasible algorithms, describe the limitations of the current formalization of feasibility as polynomial-time.
- 1e. Describe the main ideas behind the concepts of NP and NP-hardness, know examples of NP-hard problems.

Level 2: Application and Analysis

Level 2 outcomes are those in which the student can apply the material in familiar situations, e.g., can work a problem of familiar structure with minor changes in the details.

Upon successful completion of this course, students will be able to:

- 2a. Convert a non-deterministic FA (respectively transition graph) into an equivalent deterministic FA.
- 2b. Convert a transition graph or NFA into an equivalent regular expression.
- 2c. Convert a regular expression into an equivalent FA.
- 2d. Construct a regular expression (respectively a context-free grammar) for a regular language (respectively context-free language).
- 2e. Convert a context-free grammar into an equivalent pushdown automaton.
- 2f. Construct a context-free grammar for a given context-free language.
- 2f. Design an algorithm for a machine model to simulate another model.
- 2h. Build simple Turing machines.
- 2i. Prove formally properties of languages or computational models.
- 2j. Apply a parsing algorithm.
- 2k. Build a parse tree or a derivation from a context-free grammar.
- 2l. Use the closure properties in arguments about languages.

Level 3: Synthesis and Evaluation

Level 3 outcomes are those in which the student can apply the material in new situations. This is the

highest level of mastery.

Upon successful completion of this course, students will be able to:

3a. Compare regular, context-free, recursive, and recursively enumerable languages.

3b. Compare FA, PDA, and Turing machines.

Textbook: Reading and laboratory assignments will be drawn from *Introduction to the Theory of Computation*, by Michael Sipser (both 2nd and 3rd editions are OK). This book is available at the bookstore and through major online book retailers, and you are expected to acquire a copy for your use in this course. Photocopied textbooks are illegal and their use will not be tolerated.

Assignments: Reading and homework assignments will be handed out or announced in class and in labs. If you miss a class, it is your responsibility to find out what you missed. You should expect to spend at least 10 hours/week outside of class on reading and homework.

Exams: There will be three tests and the final exam.

The purpose of the exams is to allow you to demonstrate mastery of course concepts. Make-up exams will be given only in extremely unusual circumstances. If you must miss an exam, please meet with an instructor, BEFORE the exam if at all possible.

Grading: Each topic means home assignments (mainly on the sheets of paper, but some on the real computer). Maximum number of points:

- first test: 15
- second test: 15
- third test: 15
- home assignments and quizzes: 20
- final exam: 30
- attendance: 5

The nominal percentage-score-to-letter-grade conversion is as follows:

- 90% or higher is an A
- 80-89% is a B
- 70-79% is a C
- 60-69% is a D
- below 60% is an F

We reserve the right to adjust these criteria downward, e.g., so that 88% or higher represents an A, based on overall class performance. The criteria will not be adjusted upward, however.

Homework Assignments: Homework and lab assignments are designed to allow you to practice using the concepts presented in lecture and in your reading. Homework and lab assignments may include written problems, tutorial exercises, and programming problems. Assignments usually will be due at the start of the next class. Late homework will be accepted only in unusual circumstances, by prior arrangement if at all possible.

Homework must be done individually. While you may discuss the problem in general terms with other people, your answers and your code should be written and tested by you alone. If you need help, consult the TA or professors.

Quizzes: The purpose of a quiz is to ensure that you have read the weekly reading assignment and to verify that you have mastered the major concepts of recent lectures. Quizzes typically will be about 5-10 minutes in length and will cover the material assigned to be read for the upcoming lecture plus selected concepts from previous lectures. There will be no make-up on missed quizzes.

Special Accommodations: If you have a disability and need classroom accommodations, please contact the Center for Accommodations and Support Services (CASS) at 747-5148 or by email to cass@utep.edu, or visit their office located in UTEP Union East, Room 106. For additional information, please visit the CASS website at <http://www.sa.utep.edu/cass>. CASS's staff are the only individuals who can validate and if need be, authorize accommodations for students.

Scholastic Dishonesty: Any student who commits an act of scholastic dishonesty is subject to discipline. Scholastic dishonesty includes, but not limited to cheating, plagiarism, collusion, submission for credit of any work or materials that are attributable to another person.

Cheating is:

- copying from the test paper of another student;
- communicating with another student during a test to be taken individually;
- giving or seeking aid from another student during a test to be taken individually;
- possession and/or use of unauthorized materials during tests (i.e. crib notes, class notes, books, etc.);
- substituting for another person to take a test;
- falsifying research data, reports, academic work offered for credit.

Plagiarism is:

- using someone's work in your assignments without the proper citations;
- submitting the same paper or assignment from a different course, without direct permission of instructors.

To avoid plagiarism see: https://www.utep.edu/student-affairs/osccr/_Files/docs/Avoiding-Plagiarism.pdf

Collusion is unauthorized collaboration with another person in preparing academic assignments.

Instructors are required to -- and will -- report academic dishonesty and any other violation of the Standards of Conduct to the Dean of Students.

NOTE: When in doubt on any of the above, please contact your instructor to check if you are following authorized procedure.