

**Spring 2009 Syllabus**  
**Automata, Languages and Computability**  
**CS3350**

**Class meetings:** Monday, Wednesday and Friday from 11:30am to 12:20pm in room CS322.

**Instructor:** Tanja Magoc

Room: 202C

Email: T.Magoc@gmail.com

Msn messenger: tmagoc@hotmail.com

Office hours: Monday and Wednesday from 12.30pm to 1.30pm or by appointment

**Teaching Assistant:** Melisa Vela

Room: 304

Email: smvelaloya@miners.utep.edu

Office hours: Monday 12:30-1:30, Tuesday 11:00-12:00, Wednesday 4:30-5:30,

Thursday 12:30-1:30, Friday 12:30-2:30, or by appointment

**Course Objectives:** the objective of this course is to present the students with the fundamental ideas and models of computation. The following models will be presented:

Part I: finite-state automata and regular languages

Part II: pushdown automata and context-free languages

Part III: Turing machines and introduction to computability, algorithms, and Church-Markov-Turing Thesis

At the end of this course, students will be able to build an FSA (resp. PDA) for a regular language (resp. CFL) and vice versa, convert a non-deterministic FA into an equivalent deterministic one, show that a language is or is not a regular language or a context-free language using the different pumping lemmas, build parse trees and derivations, Turing machines for recursive languages. Finally, students will be able to understand the inherent limits of Computer Science through the introduction of computability.

**Text book:** Introduction to the theory of computation, 2<sup>nd</sup> edition, Michael Sipser, Thomson Course Technology edition, ISBN: 978-0-534-95097-2

**Additional reading:**

- Introduction to automata theory, languages and computation (2<sup>nd</sup> edition), Hopcroft et al. (difficult)
- Automata theory with modern applications, Anderson (difficult)

**Grading:** Your semester grade will be based on a combination of homework assignments, quizzes, a project, 3 tests and a final exam.

1. Quizzes: 20%
2. Homeworks: 10%
3. Project: 10%
4. Tests: 30%
5. Final: 30%

Note: To pass the class, the homeworks and the project have to be turned in on time.

**Scaling:**

1. 80% or higher is an A
2. 70% and under 80% is a B
3. 60% and under 70% is a C
4. 50% and under 60% is a D
5. Below 50% is an F

These criteria may be adjusted based on the overall class performance.

**Standards of conduct:** you are expected to conduct yourself in a professional and courteous manner (this means for example, being on time for class) as prescribed by the Standards of conduct. Graded work is to be done independently and should be unmistakably your own work, although you may discuss your work with other students in a general way. You may not represent as your own work, material that is transcribed or copied from another person, book, web page, or any other source. Professors are required –and will- report academic dishonesty and any other violation of the Standards of conduct to the dean of students.

**Disabilities:** if you feel that you may have a disability that requires accommodation, contact the disabled student service office at 747-5184, Room 106E Union, or email [dss@utep.edu](mailto:dss@utep.edu)

## **Learning Outcomes**

### **1. Knowledge and Comprehension**

- a. Understand the differences between regular languages, context-free languages, recursive and recursively enumerable languages
- b. Understand the differences between finite automata, PDA and TM
- c. Know how to recognize a deterministic from a non-deterministic machine
- d. Know which class of languages is accepted by which type of machine
- e. Know the Chomsky hierarchy of languages (finite, regular, context-free, recursive, recursively enumerable)
- f. Know how to build a parse tree or a derivation from a context-free grammar
- g. Understand how a finite automaton can be seen as a special type of pushdown automaton and how a PDA can be transformed into an equivalent TM
- h. Know the closure properties of regular languages and context-free languages

### **2. Application and Analysis**

- a. Be able to convert a non-deterministic FA (resp. transition graph) into an equivalent deterministic FA, to convert a transition graph or NFA into an equivalent regular expression, and to convert a regular expression into an equivalent FA
- b. Be able to construct a regular expression (resp. a context-free grammar) for a regular language (resp. context-free language)
- c. Be able to convert a context-free grammar into an equivalent PDA
- d. Be able to give a context-free grammar for a given regular language and be able to convert a right linear grammar into a transition graph
- e. Be able to convert a multi-tape Turing machine into a Turing machine and a non-deterministic Turing machine into a deterministic Turing machine
- f. Be able to build a Turing machine for a language and a Turing machine that performs a simple computation e.g. adding two integers
- g. Be able to use the pumping lemma for regular languages (resp. context-free languages) or closure properties for regular languages (resp. context-free languages) to show that a language is not regular (resp. not context-free)
- h. Be able to apply bottom-up and top-down parsing

### **3. Synthesis and Evaluation**

- a. Identify regular, context-free, recursive, recursively enumerable languages
- b. Understand the implications of Church-Turing-Markov's thesis
- c. Identify problems for which an algorithm exists, and problems for which there is no algorithms (non recursive, non recursively enumerable languages) and understand the implications of such results
- d. Identify the complexity class of a problem (P, NP, NP-complete, NP-Hard)
- e. Understand the diagonalization process to show that the Halting problem is not recursive and to show the existence of languages that are not recognizable
- f. Understand, apply and use extensions of results seen in class (for instance "a language is recursive if and only if it is recursively enumerable and co-Turing recognizable")