

## Class Information and Syllabus

**Instructor:** [Stephen R. Tate \(Steve\)](#)

**Lectures:** Mon/Wed 3:30-4:45, NMOR 329

**Office:** Petty 157

**Office Hours:** Mon/Wed 1:30-3:00, or by appointment

**E-mail:** [srtate@uncg.edu](mailto:srtate@uncg.edu)

**Class Web Page:** <https://www.uncg.edu/cmp/faculty/srtate/452/>

**Catalog Description:** Finite state automata and regular expressions, context-free grammars, push-down automata and their use in parsing, overview of language translation systems, models for programming language semantics, computability and undecidability.

**Prerequisites:** Grade of C or better in CSC 350, or permission of instructor.

**Longer Description:** In this class we explore the most fundamental question in computer science: *What does it mean to compute something?* This includes exploring issues such as processes/models that can mechanize computations, problems that can and cannot be solved with different models of computation, and fundamental limitations that restrict what problems can be solved computationally. While the focus of an algorithms class is on how *fast* problems can be solved by modern computers, this class digs deeper into the fundamental question of what problems are *possible* to solve in various computational models. The approach in this class is formal, using precise mathematical models and a significant amount of formal reasoning and mathematical proofs.

**Student Learning Outcomes:** Upon successful completion of this course students should be able to

1. *Understand* the basic theoretical models of computability: deterministic and nondeterministic finite automata, pushdown automata, and variants of Turing machines.
2. *Design* finite automata corresponding to given regular sets, and *describe* the regular set recognized by a given finite automaton. Do the same with pushdown automata and context-free languages, and with Turing machines and recursively enumerable sets.
3. *Comprehend* and *apply* a number of algorithms such as: the subset construction to transform a nondeterministic finite automaton into a deterministic one; the DFA state minimization algorithm to minimize the number of states in a deterministic finite automaton; and conversion algorithms from regular expressions to finite automata and vice-versa.

4. *Understand* limitations of finite automata (respectively, pushdown automata) and prove that some sets are not regular (respectively, context-free) by using the pumping lemma for regular languages (respectively, context-free languages).
5. *Simulate* CFGs by NPDAs and vice-versa, that is, convert a given context-free grammar to an equivalent nondeterministic pushdown automaton, and convert a nondeterministic pushdown automaton to an equivalent context-free grammar.
6. *Apply* algorithms to transform context-free grammars into normal forms such as the Chomsky and Greibach normal forms.
7. *Prove* that some problems are decidable or undecidable using techniques such as diagonalization and reduction.

**Textbook and Readings:** The required textbook is

Michael Sipser. [\*Introduction to Theory of Computation, 3rd edition\*](#), Cengage Learning, 2013.

ISBN-13: 978-1133187790 (this is the hardcover print edition – the eBook edition in the link is much more affordable).

**Topics:** The topics to be covered are shown below, with estimated times by each topics. For an updated week-by-week schedule, please see the class web site.

Class Overview [1 class]  
 Mathematical structures and proofs technique review (Chapter 0) [2 classes]  
 Regular Languages (Chapter 1) [5 classes]  
 Context-Free Languages (Chapter 2) [5 classes]  
 The Church-Turing Thesis (Chapter 3) [2 classes]  
 Decidability (Chapter 4) [2 classes]  
 Reducability (Chapter 5) [2 classes]  
 The Recursion Theorem (Section 6.1) [1 class]  
 Time Complexity (Chapter 7) [2 classes]  
 Selected Topics [2 classes]

**Teaching Methods and Assignments:** This class will meet for two 75-minute periods per week, and class meetings will consist of a combination of lecture/presentation, discussion, and in-class exercises. Students must come to class prepared, having done all required readings, and are expected to participate in in-class activities. Grades are based on student work done in assignments and exams.

*Assignments:* For practice and to demonstrate abilities, students will be given 5-6 assignments over the course of the semester (approximately every two weeks, adjusted to exclude exam weeks). Assignments will include some written problems and some programming problems. Written solutions may be prepared electronically using LaTeX or may be neatly handwritten, and are to be turned in on paper at the beginning of class on the due date. *Do not use a standard word processor such as Word or LibreOffice unless you properly write mathematics using the equation editor capabilities.* Programming problems can be completed in Java, C++, or Python; other languages may be used if pre-approved by the instructor.

*Exams:* There will be two mid-term exams and one final exam. Tentative dates for each exam are provided on the class web site schedule.

*Graduate Students:* Students taking this class for graduate credit (as CSC 652) or as a contract honors course will complete a research project on an algorithms topic of their choosing. More information, including a list of potential topics, will be given approximately halfway through the semester.

**Evaluation and Grading:** Each student work product will be graded, and the student's final grade will be determined by assigning each category of work a weighted score according to the following distribution:

<u>Undergraduates</u>		<u>Graduate Students</u>	
Assignments	35%	Assignments	30%
Mid-term Exam 1	20%	Mid-term Exam 1	18%
Mid-term Exam 2	20%	Mid-term Exam 2	18%
Final Exam	25%	Final Exam	24%
		Research Project	10%

**Academic Integrity:** Students are expected to be familiar with and abide by the UNCG Academic Integrity Policy, which is online at <https://academicintegrity.uncg.edu/>.

Assignments in this class are for individual work, unless explicitly stated otherwise. General concepts and material covered in the class may be discussed with other students or in study groups, but specific assigned problems should not be discussed and all submitted work should be entirely your own. If you use external references (including web sites, books, etc.) in preparing your solutions, you should clearly mark the part(s) of your solution influenced by these references and provide clear citations to the source of information you are using. *Just doing a Google search for solutions to assigned problems is a violation of academic integrity, whether or not you use what you find in your answer.* Sharing your own work is a serious violation of academic integrity, and if homework is copied then *both* the person who actually did the work and the person who copied it will be punished. Any incidents of academic dishonesty will be handled strictly, resulting in either a zero on the assignment or an F in the class, depending on the severity of the incident. Significant incidents will be reported to the UNCG Office of Student Rights and Responsibilities. Note that the Department of Computer Science maintains records of all academic integrity incidents, and multiple violations, even in different classes or semesters, will always result in reporting to the university and serious penalties.

**Attendance Policy:** Attendance will not be taken in class, and is voluntary; however, all students are responsible for everything done or said in class (this can include changes in assignments, due dates, etc.). Many topics covered in this class are challenging, and it is highly unlikely that a student who regularly misses classes will be successful in the course. If attendance becomes a problem, the instructor reserves the right to give pop-quizzes or graded in-class exercises.

The university allows for a limited number of excused absences for religious observances. Students who plan to take such an absence should notify the instructor at least two weeks in advance so that accommodations can be made. It is the student's responsibility to obtain notes from another student if they miss class.

**Late Policy and Makeup Exams:** Assignments are due at the beginning of class on the due date, and no credit will be given for late assignments. Students with planned absences, whether for university events, religious observance, or other reason, are expected to make arrangements with the instructor to turn in assignments or take exams before the scheduled date of the assignment or test.

Exam/test dates will be announced at least two weeks in advance, and may be made up only if it was missed due to an extreme emergency and arrangements are made before the exam date. Exams (including the final) may not be taken early or late due to personal travel plans.

**In-class Behavior:** When you are in class you should be focused on the class, and you should act in a professional and mature manner. During class there should be no eating, drinking, e-cigarettes, cellphone use, non-class related laptop use, or anything else that does not pertain to the class activities. Any distracting items may be confiscated at the discretion of the instructor.

**ADA Statement:** UNCG seeks to comply fully with the Americans with Disabilities Act (ADA). Students requesting accommodations based on a disability must be registered with the Office of Accessibility Resources and Services located in 215 Elliott University Center: (336) 334-5440 (or on the web at <https://oars.uncg.edu>). Note that if you require a separate testing environment you must make arrangements more than one week before any exam.

**University Closings:** If university facilities are closed due to flu outbreak or other emergencies, it does not mean that classes are canceled. In such an event, please check the class web page and Canvas site for information about if and how the class will proceed.

**Commercial note-taking services:** Selling class notes for commercial gain or purchasing such class notes in this or any other course at UNCG is a violation of the University's Copyright Policy and of the Student Code of Conduct. Sharing notes for studying purposes, or borrowing notes to make up for absences, without commercial gain, are not violations.