Air Force Institute of Technology
Graduate School of Engineering and Management
Department of Electrical and Computer Engineering

CSCE 532 Automata and Formal Languages
Winter 2019
4 Quarter Credit Hours
MW 1000-1150
Bldg 646 Rm 216

Prof Laurence D. Merkle
Bldg 640 Rm 305C
Appointments via Outlook recommended, but not required
937-255-3636 x4526
laurence.merkle@afit.edu

SYLLABUS

COURSE CATALOG DESCRIPTION

The objective of this course is to prepare the student with a basic foundation in the concepts of automata and formal language theory. Topics covered will include Turing machines, finite state automata, combinatorics, and formal language theory.

LEARNING GOALS

A year or more after successfully completing this course, students will be able to:

FOUNDATIONAL KNOWLEDGE

- Explain relationships among corresponding automata, formal languages, and grammars
- Recall that problems exist that no computer can solve
- Recall common asymptotic complexity classes and their order
- Explain computational nondeterminism in terms of randomness, oracles, and set-based states.
- Explain the significance of Chomsky's Hierarchy and associate storage requirements
- Explain the Church-Turing Thesis
- Recall a proof of the undecidability of the Halting Problem
- Explain the process of proving undecidability by reducing one problem to another
- Explain the practical implications of common asymptotic complexity classes, including intractability
- Explain the significance of the question $P \overset{?}{=} NP$ and NP-completeness
- Explain the process of proving NP-completeness by reducing one problem to another

APPLICATION

- Evaluate formal proofs involving automata, formal languages, and grammars.
- Evaluate formal proofs involving decidability.
- Evaluate formal proofs involving asymptotic complexity.
- Prove relationships among given specific automata, formal languages, and grammars.
- Prove relationships among given classes of automata, formal languages, and grammars.
- Analyze given computational problems to identify the relevant formal languages.
- Given an automaton, formal language, or grammar, convert it to one of the other representations.
- Determine the asymptotic complexity of given algorithms.
- Use JFLAP to support other learning goals.

INTEGRATION

- Identify the relationships among grammars, automata, and formal languages contained in Chomsky's Hierarchy
- Connect Turing machines and decidability
- Compare and contrast undecidability and intractability
- Identify the interactions between formal language classes and real-world storage requirements.
- Identify the interactions between asymptotic time complexity and real-world efficiency/effectiveness tradeoffs.
- Connect decidability and intractability with security.

HUMAN DIMENSION

- Come to see themselves as being capable of developing novel and rigorous proofs, provided they allow themselves sufficient time for the creative process.
- Be able to explain to their peers or superiors why wishful thinking will not change
  - Storage requirements dictated by classes of formal languages
  - Undecidability of certain problems
  - Intractability of certain algorithms

CARING

- Be more interested in computational theory
- Value mathematical rigor
- Get more excited about formal mathematics

LEARNING HOW TO LEARN

- Keep "paper and pencil" handy when reading mathematical material for deep comprehension.
- Schedule significantly more time per page when reading mathematical material than when reading nontechnical material.
- Schedule nontrivial elapsed time when developing rigorous proofs to allow the necessary room for creativity.
- Frame useful questions in automata theory, language theory, computability theory, and complexity theory.
- Identify sources of information on automata theory, language theory, computability theory, and complexity theory.
- Set learning agendas regarding automata theory, language theory, computability theory, and complexity theory.

COURSE ORGANIZATION

The topics covered in the course will be:

- Regular Languages
- Context-Free Languages
- The Church-Turing Thesis
- Decidability
- Reducibility
- Time Complexity
- Space Complexity

MATERIALS

- *Introduction to the Theory of Computation,* Michael Sipser, Third Edition, Cengage Learning, 2013
- *JFLAP* Software

PREREQUISITE

- CSCE 531 Discrete Mathematics

LEARNING ACTIVITIES

- Textbook readings
- Web-based videos and readings
- In-class exercises
- Homework

EVALUATION AND GRADING POLICY

Grades will be determined according to the following:

Homework	30%
Midterm Exam	30%
Final Exam	30%
Participation	10%

Course Average	Grade
[92.0 100.0]	A
[90.0 92.0)	A-
[87.0 90.0)	B+
[82.0 87.0)	B
[80.0 82.0)	B-
[77.0 80.0)	C+
[72.0 77.0)	C
[70.0 72.0)	C-
[60.0 70.0)	D
[0.0 60.0)	F

COURSE POLICIES AND EXPECTATIONS

- Attendance and active participation are expected.
- Academic integrity as described in AFIT, EN, and ENG policies is expected. Collaboration on exams is prohibited. Collaboration on homework is strongly encouraged.
- Exams must be completed as scheduled unless the instructor approves extenuating circumstances. Approval is more likely if it is sought in advance. Regardless, approval will not be granted on the basis of lack of preparedness.
- Laptop and other computing device usage directly related to current classroom activities is encouraged. Students are responsible for ensuring that devices do not become a distraction to themselves or others.
- The instructor will treat the students with respect. Students are expected to treat the instructor and their fellow students with respect.
- The instructor is available by appointment via Outlook, often has an open door during duty hours, and will respond to emails within one duty day. The instructor will typically not be available outside of duty hours.

COURSE CALENDAR

The course will proceed sequentially through the topics listed above. The pace will be quick. A tentative schedule will be posted at the beginning of the course and updated periodically. Note that there will be no class meetings on New Year’s Day, Martin Luther King, Jr.’s Birthday, or President’s Day. Also, the instructor will be TDY one week for the ACM SIGCSE Technical Symposium on Computer Science Education.

ADVICE

- Students should keep pencil and paper (or a suitable electronic substitute) handy while doing the assigned reading to verify and better understand the less obvious points.
- Students should begin homework assignments early to allow time for the creative process.