CS 3530: Computational Theory

Fall 2007 Syllabus

Prerequisites: CS 2420, CS 2810, and Math 3310

An introduction to the theory of computation. Theoretical computation models are introduced, along with a formal treatment of their capabilities and limitations. Topics include regular languages and finite automata, context-free languages and pushdown automata, Turing machines and the Church-Turing Thesis, decidability, and complexity, including NP-completeness. Students will complete written exercises and also use computer simulations of the various formalisms.

Instructor Dr Russ Ross

Telephone 652-7971

Email russ@dixie.edu

Office Hours Tuesdays and Thursdays 2:30–4:00 pm, Wednesday 1:00–3:00 pm in Hazy 327

1 Objectives

At the end of the course, students will

- Understand and be able to reason about the capabilities of various computational models.
- Be able to use formal notations and apply rigor in their analysis of formal systems.
- Recognize and understand different classes of computational complexity.
- Understand the limits of computational models and the real-world systems that rely on them.

2 Resources

2.1 Texts

There are two texts for this course. Both are available from the campus bookstore:

1. *Introduction to the Theory of Computation, 2nd edition* by Sipser, ISBN: 0-534-95097-3
2. *JFLAP: An Interactive Formal Languages and Automata Package* by Rodgers and Finley, ISBN: 0-7637-3834-4

The first is required and is the primary text for the course. The second is optional. It provides documentation for and exercises using the JFLAP simulation system that we will use. Documentation can also be found online, but students may find it helpful to have a printed manual with more detailed directions and examples.

Information about JFLAP and downloads of the system are available at `http://jflap.org/`, and JFLAP is available on machines in the computer labs.

2.2 Labs

You may use the computers in Udvar-Hazy 151 and Udvar-Hazy 200. There will also be lab assistants in these labs.

2.3 Course Web Site

This course has an accompanying website. You are responsible for announcements, the schedule, and other resources posted on the website. To access it, you will need a valid CIT username and password. If you do not already have a CIT login, lab assistants can help you sign up for one. The course website is accessible at `https://moodle.cs.dixie.edu/` (*not* `http`).

3 Assignments and Exams

3.1 Reading

The student is responsible for reading the material in the textbook. A reading schedule is provided with the class schedule. The student is expected to read the material before the class in which it is discussed.

3.2 Exams

There will be a midterm and a final. Topics from lectures, assigned readings, and lab work are all eligible for examination. Exams may include extensions of homework assignments, so be sure to keep copies of all of your work.

3.3 Assignments

Assignments will be graded for correctness and elegance. It is important that you start early and get each of your assignments done before its due date. Many problems will take much longer to solve in a single sitting than in many shorter sessions. Give yourself time to think; sleep on difficult problems. Finish early so you can go back and refine your initial approach. Assignments are due at 11:59 pm on the date listed in the schedule, and must be submitted through the course website.

3.4 Submitting Assignments

All assignments must be typed and submitted electronically. Submissions should be PDF files unless otherwise requested. Students are advised to use LaTeX for best results. LaTeX is available on the Linux machines in the labs, and information is available online at `http://www.nhn.ou.edu/~morrison/LaTeX/index.shtml`

3.5 Grading

Assignments and exams each contribute to your point total. The midterm exam will count like a single assignment, and together the assignments and the midterm will comprise 70% of your grade. The final exam will be worth the remaining 30%. If you complete all homework assignments and all exams, your lowest homework score will be dropped at the end of the term. If you fail to complete any of the assignments, or you submit one beyond the hard deadline (two days after the initial deadline), your lowest score will *not* be dropped. Note that the midterm exam counts as an assignment for this policy. No specific grading scale is predefined.

4 Policies

4.1 Disability statement

If you are a student with a physical or mental impairment and would like to request accommodations, please contact the Disability Resource Center (652-7516) in Room 201 of the Student Services Center. The Disability Resource Center will determine your eligibility for services based upon complete professional documentation. If you are deemed eligible, the Disability Resource Center will further evaluate the effectiveness of your accommodation requests and will authorize reasonable accommodations that are appropriate for your disability.

4.2 Attendance

Students are responsible for material covered and announcements made in class. School-related absences may be made up only if prior arrangements are made. The class schedule presented is approximate. The instructor reserves the right to modify the schedule according to class needs. Changes will be announced in class.

4.3 Time Commitment

Courses should require about 45 hours of work per credit hour of class. This class will require about 135 hours of work on the part of the student to achieve a passing grade, which is approximately 9 hours per week. If you do not have the time to spend on this course, you should probably rethink your schedule.

4.4 Late Policy

Assignments are due at 11:59 pm on the date specified in the schedule. Handing them in after 11:59 pm is considered one day late. You may turn them in up to two school days late with penalties as described below. After two days late, you receive zero points.

For example: if an assignment is due Thursday at 11:59 pm:

- Before Friday 11:59 pm the assignment is considered 1 day late.

- Before Monday 11:59 pm the assignment is considered 2 days late.
- After Monday 11:59 pm the assignment will not be accepted.

Saturdays, Sundays, and school holidays do not count as late days. Late days do *not* extend beyond the last day of class.

Each student is given five free late days to use over the course of the semester. The lateness of an assignment will be determined according to the rules given above, and the first five late days used during the semester will be forgiven. After that, each late day will result in a 10% penalty.

Important notes:

- Even using free late days, students cannot submit assignments more than two days late and receive credit. No assignments will be accepted more than two days past the original deadline.
- Free late days are applied to the *first* five late days during the semester. Students cannot control which late days are penalized and which ones forgiven; the first five late days in the semester are forgiven, and the rest are penalized.
- Free late days only apply to students who submit *every* assignment within the two-day cutoff period. For example, if you fail to submit the fifth assignment, or submit it more than two days late, you will forfeit all free late days, *including those used for earlier assignments.*
- No other extensions will be granted, except under exceptional circumstances. Students should reserve their free late days to use in the event of illness, emergencies, traveling, sports conflicts, etc. Students are advised not to use their free late days early in the semester, as assignments tend to get more difficult and schedules tighter as the semester progresses.

4.5 Collaboration

Limited collaboration with other students in the course is permitted. Students may seek help learning concepts and developing programming skills from whatever sources they have available, and are encouraged to do so. Collaboration on assignments, however, must be confined to course instructors, lab assistants, and other students in the course. Students are free to discuss strategies for solving programming assignments with each other, but this must not extend to the level of programming code. Each student must code his/her own solution to each assignment. See the section on cheating.

4.6 Cheating

Cheating will not be tolerated, and will result in a failing grade for the students involved as well as possible disciplinary action from the college. Cheating includes, but is not limited to, turning in homework assignments that are not the student's own work.

You are encouraged to work in groups while studying for tests, discussing class lectures, discussing algorithms for homework solutions, and helping each other identify errors in your homework solutions. If you are unsure if collaboration is appropriate, contact the instructor. Also, note exactly

what you did. If your actions are determined to be inappropriate, the response will be much more favorable if you are honest and complete in your disclosure.

Where collaboration is permitted, each student must still create and type in his/her own solution. Any kind of copying and pasting is *not* okay. If you need help understanding concepts, get it from the instructor or fellow classmates, but never copy another's code or written work, either electronically or visually. The line between collaborating and cheating is generally one of language: talking about solutions in English or other natural languages is usually okay, while discussions that take place in C++ or other programming languages are usually not okay. It is a good idea to wait at least 30 minutes after any discussion to start your independent write-up. This will help you commit what you have learned to long-term memory as well as help to avoid crossing the line to cheating.