Welcome to COMP 330 Fall 2015
Theory of Computation

Instructor : Prakash Panangaden

3rd September 2015

Welcome to COMP 330. This is the principal undergraduate course on the principles of theoretical computer science at the School of Computer Science (SOCS). In this course we will focus on the foundations of the subject. It is fascinating that the history of computer science (what a silly name for a subject!) began several thousand years before electronic computers were invented. In ancient times people invented algorithms for all kinds of purposes, but it was only in the last 70 years that the electronic computer was invented. If you look up the word "computer" in a 1946 dictionary it is defined as a *person* who performs computations.

The question of what constitutes an algorithm was raised early in the twentieth century[1]. Early investigations by logicians: Hilbert, Gödel, Tarski, Post, Markov, Kleene and above all Turing and Church gave answers that we accept today. Remarkably, they discovered that there were *intrinsic* limits of computability. By "intrinsic" I mean inherent in the nature of logic and nothing whatever to do with technology. This led to the subject of computability theory: one of the main topics of this term. All this also happened before the invention of the electronic computer.

It may seem that machine architecture is far removed from these theoretical considerations. In fact, the idea that algorithms could be represented by symbolic data, and that there was a "universal" machine which could execute these algorithms with a fixed instruction set, was a direct result of the logical investigations. Today, of course, these symbolic representations are called "software" and the universal machines are available for sale to every household: we call them computers. They are even embedded in mobile phones, tablets, cars, washing machines and microwave ovens.

Two other important inputs into the subject are (1) the theory of the brain which led to automata theory (the first one-third of the course) and (2) linguistics, which led to the notion of formal languages. One type of formal language is called a "context-free" language and is the basis for the syntax of *all* programming languages today.

Lectures

There two lectures per week: on Tuesdays and Thursdays at 11:35 to 12:55 in Room 112 of the Rutherford Building.

[1] Actually, in the last year of the 19^{th} century.

Administration

I am responsible for all the administration of the class and for all the assignments. Please send me questions by email to `prakash@cs.mcgill.ca` and I will try to get back to you. There are four teaching assistants as well: Giulia Alberini, Harrison Humphrey, Yiqiao Li and Patricia Olson. They mark the assignments *according to strict guidelines made up by me*. To dispute assignment marks please see me first. We also have an undergraduate mentor, Pascale Gourdeau who will work with us; she will not mark any assignments but will help students who feel that they need additional support. Harrison, Pascale and Tricia were all students in 330 last year so they are familiar with the experiences of students taking the course.

Assignments

There will be 6 assignments. You will be expected to give *mathematically rigorous* proofs in these assignments. The assignments should be written on paper, not submitted electronically or carved on stone tablets. I will collect them in class on the due date **at the start of class**. The assignments will be mostly marked by the teaching assistants under my supervision and according to my grading guidelines. I will mark some of the questions from time to time. You can discuss with a TA whether your answer is correct but you cannot discuss with them how much a question is worth. You can discuss that with me if you like, but so far I have never been persuaded to change my mind. Some assignments will include alternate questions: these are usually much harder, please do not attempt them unless you are sure of what you are doing. I also include spiritual growth questions: these are even harder and will earn you **no credit whatsoever** for the class. However, if you solve one of them you will feel great personal satisfaction.

References

The textbook for the course is called

> Introduction to the Theory of Computation by Michael Sipser.

I will follow it *loosely* - to do well in the course you must know the material that I covered in my lectures. Another excellent book – in fact, the textbook in an earlier year – is *Automata and Computability by Dexter Kozen*. I will also *occasionally* put some notes on the web site.

Grading

The homework accounts for 30% of the total course marks. The midterm examination accounts for 10% and the final examination will be 60% of the total. The midterm examinations will be in class, the final examination will be in the final examination period. The mid-term and the final exam will be closed book.

Staying in touch

I will hold office hours twice a week:

> **Mondays 11:30 to 1:00 and Thursdays 1:30 to 3:00 in Room 105 N McConnell.**

This is the North wing of McConnell. Please do come and see me during those hours. Do not feel that you are infringing on my time - my office hours represent time that I have reserved specifically to see you. During office hours do not loiter outside hoping that I will see you or wait for the person(s) inside to leave. Some people hang around in my office indefinitely so you need to barge in and demand my attention.

Outside my office hours finding me is a hit or miss proposition, wandering by and looking for me is not recommended as I might be not there or working on something else. I recommend using email to ask me questions. The TAs and Pascale will also hold office hours which I will post on the course web page. **There will be a class web page**. Please ensure that you have access to it. The URL is
`www.cs.mcgill.ca/~prakash/Courses/Comp330/comp330.html`

I will use the myCourses system to maintain a discussion board. Your marks will be disseminated through this system. I will check the discussion board from time to time but do not rely on it as a way of getting a last-minute response to a crucial homework question.

Hints for taking this course

This course is considered by many to be the hardest course at SOCS and by some others the easiest. I am not sure why there is such a wide variation in performance, but here are a few hints that might help.

1. **Do the homework**. Finding videos of lectures on the web is no substitute for thinking for yourself. You can, of course, find many homework questions solved on the web, but if you do not engage with the material you will learn nothing and **you will suffer during the tests**.

2. **Read the solutions**. I write them with some care. You can learn a lot from seeing how the solutions are presented.

3. The lectures are intended to be thought about and discussed. If you do not understand everything that is going on in the classroom, do not be either discouraged or surprised. I expect that you have to think about these ideas before they sink in. Do try and come to class. Looking at notes taken by others just does not have the same impact. The course is conceptual not factual.

4. **Please come to my office hours**. Sometimes it requires a dialogue to unblock a misconception. There is no shame in seeking help.

5. Learn to read mathematics. For most of you this is your biggest obstacle. It has nothing to do with intelligence: it has everything to do with care and discipline. I have seen people confuse "for all" and "there exists" even when they can explain perfectly what the difference

is. If you express yourself clearly and read accurately there is little in the course that is really tricky.

6. Use language carefully and precisely. In speaking to me, I will insist on correct usage because much of the confusion about a topic goes away if you learn to express yourself precisely.
7. Don't even ask me if the lectures are going to be recorded. My experience with COMP 302 last year tells me that for too many of you this is an invitation to blow off the class and then you **never** catch up. If you miss a class you can find notes on the website or material in the book. You can also come see me or the TAs.

Academic Integrity I am required to include the following statement in my course outline:

> McGill University values academic integrity. Therefore all students must understand the meaning and consequences of cheating, plagiarism and other academic offenses under the Code of Student Conduct and Disciplinary Procedures (see `http://www.mcgill.ca/integrity` for more information). Most importantly, work submitted for this course must represent your own efforts. Copying assignments or tests from any source, completely or partially, or allowing others to copy your work, will not be tolerated.

I am okay with students working together, but in the end you should understand what you turn in. I will take action if you turn in a perfect solution which you then cannot explain to me.

Submission of written work in French. In accord [*sic*] with McGill University's Charter of Students' Rights, students in this course have the right to submit in English or in French any written work that is to be graded.

Chaque étudiant a le droit de soumettre en français ou en anglais tout travail écrit.

Lecture Schedule

Week 1	Lecture 1	Sept 8	Introduction, basic maths	P, *
	Lecture 2	Sept 10	Deterministic finite automata	P, 1.1; A1 out
Week 2	Lecture 1	Sept 15	Nondeterministic finite automata	P, 1.2
	Lecture 2	Sept 17	Regular expressions, Kleene's theorem	P, 1.3
Week 3	Lecture 1	Sept 22	Minimization	P, *
	Lecture 2	Sept 24	The Myhill-Nerode theorem	P, *; A1 due, A2 out
Week 4	Lecture 1	Sept 29	Brzozowski's algorithm and duality	P, *
	Lecture 2	Oct 1	Learning an automaton	K, *
Week 5	Lecture 1	Oct 6	Special topic	K, *
	Lecture 2	Oct 8	The pumping lemma	P, 1.4; A2 due, A3 out
Week 6	Lecture 1	Oct 13	The pumping lemma	P, 1.4
	Lecture 2	Oct 15	**MIDTERM**	
Week 7	Lecture 1	Oct 20	Context-free languages	P, 2.1
	Lecture 2	Oct 22	Pushdown automata	P, 2.2; A3 due, A4 out
Week 8	Lecture 1	Oct 27	The pumping lemma for CFLs	P, 2.3
	Lecture 2	Oct 29	Using the pumping lemma	P, 2.3
Week 9	Lecture 1	Nov 3	Models of computation	P, 3.1, 3.2
	Lecture 2	Nov 5	Basic computability theory	P, * 5.1; A4 due, A5 out
Week 10	Lecture 1	Nov 10	Reducibility	P, *,5.1
	Lecture 2	Nov 12	Reducibility, Rice's theorem	P, 5.1,5.2,5.3
Week 11	Lecture 1	Nov 17	Special topic	K
	Lecture 2	Nov 19	Undecidable problems about CFGs	P, 5.1; A5 due, A6 out
Week 12	Lecture 1	Nov 24	The Post Correspondence Problem	P, 5.2
	Lecture 2	Nov 26	Logic and undecidability	P, *
Week 13	Lecture 1	Dec 1	The recursion theorem	P, 6.1
	Lecture 2	Dec 3	Review for the final	P; A6 due

The section numbers on the right correspond to sections of the book. A $*$ indicates that the topic is not in the book. Notes will be provided on the website for topics that are not in the book. The letter **P** means that I will be giving the lecture, **K** indicates that some else will give the lecture.