Course Overview and Final Prep Checklist

Anup Rao

December 8, 2018

In this course, we discussed the broad strokes of many important results in complexity theory. Here I want to remind you of the big takeaways from the course, to summarize what we have learnt and give you a list of topics to revisit to prepare for the final exam.

We started the course by talking about models of computation. We focused on Turing machines and Circuits. We talked about:

Uniform vs non-uniform Turing machines are a uniform model of computation, circuits are non-uniform.

Uncomputable functions We used diagonalization to find functions that cannot be computed by Turing machines. We showed that Halting cannot be computed by a Turing machine.

Tight bounds on circuit size We showed that every function $f : \{0,1\}^n \to \{0,1\}$ can be computed by a circuit of size $O(2^n/n)$, and this it tight—we used counting arguments to show that there are functions that cannot be computed with smaller circuits.

Hierarchy theorems We showed that having strictly more time/space gives a strict increase in power for Turing machines—there are functions computable with more power that cannot be computed with less power. The same can be shown for circuits (strictly more depth or size gives strictly more power).

Open problems We do not know how to prove lower bounds on the running time for many specific problems we are interested in. For example, we do not know of any specific problem that requires superlinear sized circuits.

We began exploring the power of non-determinism, defining the classes **P** and **NP**. We talked about

Polynomial time reductions which give us a way to show how the complexity of different problems are related to each other.

NP-completeness We showed that many combinatorial problems are **NP**-complete, like satisfiability, independent set and hamiltonian path, and 3-coloring.

Oracles We talked about oracle machines and showed that there are oracles A, B such that $\mathbf{P}^A = \mathbf{NP}^A$ and $\mathbf{P}^B \neq \mathbf{NP}^B$.

Two definitions We discussed how **NP** can be defined using verifiers, but also using non-deterministic Turing machines.

We moved on to talking about space bounded computations. Here discussed several results

Configuration graph We talked about how every small space machine has a small configuration graph. This allowed us to show that every space s algorithm can be simulated in time $2^{O(s)}$.

Savitch's algorithm We gave a $\log^2(n)$ space algorithm for graph connectivity, which allowed us to show results like **NPSPACE** $=$ **PSPACE** and that $L \subseteq$ **DSPACE**$(\log^2 n)$.

NL = coNL . This was a nice proof that was based on giving an **NL** algorithm for *dis*connectivity.

TQBF . We showed that the TQBF problem is **PSPACE**-complete.

In the last part of the course, we discussed advanced topics like randomized algorithms. Here we talked about

Randomized algorithms Including definitions for the classes **RP**, **BPP**, **ZPP**. We discussed various ways in which these definitions are robust to the choice of the constants used in the definitions.

Schwartz-Zippel lemma This lemma gives a bound on the number of roots of a multivariate polynomial can have in a grid.

Permanent vs Determinant We talked about these two polynomials on matrices. Any algorithm for the permanent would give an algorithm for every problem in **#P**. In contrast, we have polynomial time algorithms for the determinant.

IP We talked about interactive proofs, and showed that one can compute the permanent with an interactive protocol. We have **IP** $=$ **PSPACE**.

The final exam will be almost entirely True/False questions like you had on the mid-term. This is to make sure that getting stuck on any one question does not derail your whole exam. Make sure you answer the easy questions first before moving on to the hard ones. Remember that if you are having a hard time, then your classmates are probably having a hard time too. You do not need to get every answer right to do well in this class!