COS522: Computational Complexity Fall 2011

Princeton University

Instructor: Sanjeev Arora

Lecture 7 notes Oct 6. Randomized computation

1. Randomized computation. Simulating arbitrary bias with fair coin and vice versa. Difference between expected running time and worst-case.
2. Example: finger printing trick (akin to hashing). To check if two n-bit files are different, only need to exchange log n bits!
3. Allow randomized algorithms to make error. RP, coRP, BPP.
4. Why 1/3 vs 2/3: can boost. (Law of large numbers.)
5. Polynomial identity testing. (i) univariate case: deg d poly has only d roots. (ii) Schwartz zippel: If vars chosen from set of size S, then prob of zero is at most d/S.
6. Application: detecting matchings. (Lovasz’s algorithm.)
7. Application: identity testing. Circuit of size m computes a poly of degree at most 2^m. However, answer can take 2^m bits to represent. Need fingerprinting trick.
8. Sipser Gacs Theorem.
9. Complete problems? Don’t know. Semantic vs Syntactic classes.
10. Randomized reductions.
11. Interactive proofs. NP: one round interactive proof.
12. Add more rounds? Nothing changes.
13. Add randomness? Changes a lot! (Next time)

Homework: 5.1; 5.6, 5.10; 6.3, 6.5; 6.18; 7.3; 7.7;7.8.