CS 744: Pseudorandomness Generators

Lecture 1: Randomness and its benefits

December 31, 2015

Often, when deterministic solutions have eluded us, there is a very efficient randomized solution at hand.

Since it is often easier to design a simple randomized strategy, it is conceivable that a general strategy to obtain deterministic solutions is the following — first, think of a randomized strategy, and then *derandomize* the algorithm by making the random source used in the algorithm, redundant. Thus in this course, we will adopt a new perspective — in addition to the time and space required for an algorithm, we will think of *the number of independent random bits used by the algorithm* as a resource bound. *Derandomization* is the process of eliminating, or at the very least, reducing, the number of independent random bits used in an algorithm. This course is about one of these methods of derandomization. Before this, we take a look at where randomness helps.

Randomized algorithms is a well-studied area in computer science. Even though it is interesting to examine randomized algorithms for isolated problems, in computational complexity theory, however, we will be judicious in our choice of problems for which we seek randomized algorithms. The problems we examine will tend to be *complete* in some sense, for some complexity class.

We will illustrate the power of randomness over with two examples — in the first case, we do not yet know of a deterministic efficient solution. In the second, the randomized solution is the basis for a deterministic one.

1 Polynomial Identity Testing

A polynomial in n variables is a function of the form

$$P(x_1, \ldots, x_n) = \sum_{i_1, \ldots, i_n \in \mathbb{N}} c_{i_1, \ldots, i_n} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n},$$

where the coefficients come from some *field*, like the field $\mathbb{Q}$ of rationals, $\mathbb{R}$ of reals, $\mathbb{Z}_p$ of integers modulo a prime.[1] The individual terms $x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}$ etc. are called *monomials*. The *degree of a multivariate polynomial* is the sum of the exponents over its monomials with non-zero coefficients.

Suppose we are given two degree d *formal* polynomials in n variables, $P(x_1, \ldots, x_n)$ and $Q(x_1, \ldots, x_n)$. We would like to know if $P = Q$, *i.e.* whether the corresponding coefficients in P and Q of like-degree terms are identical.[2] This problem is equivalent to deciding whether a given polynomial $P'(x_1, \ldots, x_n)$ of degree d is identically zero.

[1] The set $\mathbb{Z}$ with addition and multiplication is not a field, but is an *integral domain*.

[2] We are dealing with formal polynomials, hence we are not concerned with whether P evaluated at a point is equal to Q evaluated at that point.

The problem is trivial if all the coefficients of P and Q are explicitly given. There are however, three interesting and realistic versions where the problem is non-trivial.

1. If P is given as a "black-box" so that we can evaluate it at any point, but we do not know its explicit algebraic form. We can assume that we know the field $\mathbb{F}$ of the coefficients, the number n of variables and the degree d of the polynomial, but nothing further.

2. We are given the polynomial as an arithmetic formula, but possibly different from the "sum of monomials" form above. We can try expanding the given form to the "sum of monomials" form and then compare for equality, but this expansion process may take time exponential in the length of the original expression.

3. We may be given the polynomial as an *arithmetic circuit*.

We would like to have a randomized efficient technique to test whether a given multivariate polynomial is identically zero. For this, we will use the following useful fact which is easily established through induction.

Lemma 1. *(Schwartz-Zippel Lemma) If f is a non-zero polynomial of degree d over a field $\mathbb{F}$ and $S \subseteq \mathbb{F}$, then*

$$Pr\left[(a_1, \ldots, a_n) \mid a_1, \ldots, a_n \in S \quad \textit{and} \quad f(a_1, \ldots, a_n) = 0\right] \leq \frac{d}{|S|}.$$

A univariate polynomial of degree d has at most d roots, hence the above probability is the fraction of roots which are in S, which is clearly at most $d/|S|$. The Schwartz-Zippel Lemma is a generalization of this observation to multivariate polynomials. We can prove this by induction on n, the number of variables in the polynomial [3]

Proof. The base case, that of univariate polynomials, is clear by the above discussion. Assume that the hypothesis holds for all non-zero polynomials (of all degrees) in at most $n-1$ variables.

Let

$$f(x_1, \ldots, x_n) = \sum_{i_1, \ldots, i_n \in \mathbb{N}} c_{i_1, \ldots, i_n} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}.$$

Then we may rewrite f as follows.

$$f(x_1, \ldots, x_n) = \sum_{j=0}^{d} x_1^j g_j(x_2, \ldots, x_n),$$

where g_js are polynomials of degree $d-j$ in $n-1$ variables. Since f is non-zero, at least one of these g_js is also non-zero. By the induction hypothesis, for each non-zero g_j,

$$Pr\left[(a_2, \ldots, a_n) \mid a_2, \ldots, a_n \in S \quad \text{and} \quad g_j(a_2, \ldots, a_n) = 0\right] \leq \frac{d-j}{|S|}.$$

For a non-zero polynomial g_j, the probability

$$Pr\left[(a_1, a_2, \ldots, a_n) \mid a_1, a_2, \ldots, a_n \in S \quad \text{and} \quad a_1^j g_j(a_2, \ldots, a_n) = 0\right] =$$
$$Pr\left[a_1 \in S \mid a_1^j = 0\right] + Pr\left[(a_2, \ldots, a_n) \mid a_2, \ldots, a_n \in S \quad \text{and} \quad g_j(a_2, \ldots, a_n) = 0\right],$$

and this is at most $\frac{j+d-j}{|S|}$, *i.e.* $\frac{d}{|S|}$ by the induction hypothesis. □

[3] The induction does not proceed on the degree d.

A randomized algorithm for testing whether a given polynomial P with n variables and degree at most d, with coefficients from a finite field $\mathbb{F}$ is as follows. Pick a set S of $2d$ points from the field $\mathbb{F}$. Uniformly at random, pick points $a_1, \ldots, a_n$ from S and evaluate $f(a_1, \ldots, a_n)$. If f is zero, then we say that f is identically zero on $\mathbb{F}$, otherwise, we say that f is not identically zero on $\mathbb{F}$.

This algorithm is an efficient randomized solution for all the three non-trivial variants discussed above. By the Schwartz-Zippel Lemma, the above algorithm will be wrong with probability at most $1/2$, when $f \neq 0$. Is this a one-sided error or a two-sided error?

We can reduce the error by repeating the experiment independently. By independently conducting the experiment k times, it is possible to reduce the error to at most $\frac{1}{2^{k/2}}$. Since we are considering feasible computations broadly, we tend to choose k to be polynomial in the size of the input. This reduces the error to such an extent that the randomized algorithm will practically never be wrong. Hence for all practical purposes, we have a solution to the problem at hand, once we have a randomized algorithm with bounded error probability. We will see the formal justification for this claim in the next class.