6.885 Algebra and Computation September 7, 2005

Lecture 1

Lecturer: Madhu Sudan *Scribe: Swastik Kopparty*

1 Motivation

1.1 Historical

Historically, the first algorithms were for manipulations of numbers. Take a simple example like that of multiplying two n bit numbers. We know 3 algorithms:

- Repeated addition - $exp(n)$ time
- Long multiplication - n^2 time
- FFT based - $n\text{polylog}(n)$ time

Increasing algebraic sophistication got us better running times.

The roots of both words "algebra" and "algorithm" are related. "Algorithm" comes from Al-Khwarizmi, author of "Al Jabr", from which "algebra" comes.

1.2 Problems

Algebra is a source of many interesting problems, very related to important computational problems. They all fall under the theme "Find a solution subject to many constraints".

Consider the following problems:

- Sorting
- SAT: find $x_1, ...x_n$ s.t. $\forall$ clause $C \in C_1, \ldots, C_m$, $C_i(x) = true$.
- Linear equations: find $x_1, ...x_n$ s.t. $\forall$ linear functions $l \in l_1, \ldots l_m$, $l_i(x) = b_i$. Notice the similarity.
- Linear programming is similar
- Root finding: given $a_0, \ldots a_n$, find x s.t. $\sum_i a_i x^i = 0$
- Primality testing: test if N is a prime number.

1.3 Many intriguing *hard problems*

Consider the following natural problem on finding solutions to a system of polynomial equations:

- Given $P_1, \ldots P_m$ polynomials in variables $x_1, \ldots, x_n$ with $0, 1$ coefficients
- Find $x_1, \ldots, x_n$ real numbers satisfying

This problem is NP hard and is solvable in PSPACE. Assuming the Extended Riemann Hypothesis, it is in the Polynomial Hierarchy! This is representative of the very intriguing algorithmic nature of algebraic problems.

Factoring integers, of course, is another such question.

The permanent of a matrix is an important function. Recall that

$$det(A) = \sum_{\pi \in S_n} (-1)^{sign(\pi)} \prod_{i=1}^{n} A_{i\pi(i)}.$$

Analogously define the permanent

$$per(A) = \sum_{\pi \in S_n} \prod_{i=1}^{n} A_{i\pi(i)}.$$

We don't know a polynomial time algorithm to compute the permanent of a matrix. The fact that the permanent is computationally difficult and also a low degree polynomial has many ramifications for complexity theory.

2 Contents of the course

2.0.1 Review

Groups, Rings, Fields. Notable is Gauss' lemma, a suffucient condition for factorization to be well defined

2.0.2 Sample of Algorithms

FFT, Primality Testing, Matrix Multiplication

2.0.3 Factorization of polynomials

Over finite fields, square roots over finite fields. Over rationals, Lenstra-Lenstra-Lovasz algorithm. Polynomials in several variables.

2.0.4 Ideals, Varieties, Algorithms

After a book by the same title by Cox, Little, O'Shea. Complexity of solving a system of polynomial equations in several variables.

2.0.5 Complexity Results

This will be of a survey nature.

2.1 Assignment

Email madhu@mit.edu with

- Who you are
- Your background
- Why you are in the course