

CSE 203A: Recent Developments in Algorithms

Lecture Notes: Schwartz-Zippel Polynomial Identity Testing and Max Cut

Lecturer: Russell Impagliazzo

Scribe: Stefan Schneider

January 24, 2013

1 Schwartz-Zippel Polynomial Identity Testing

The Schwartz-Zippel algorithm for polynomial identity testing is an example for an “If any, then many” randomized algorithm. Given a polynomial, we want to determine if it is 0 for all assignments, e.g.

$$(x - 5y)^{1012} + (x^2 + xy)^{506} - (y + x)^{1012} \stackrel{?}{=} 0$$

The trivial solution, multiplying the polynomial out, can take exponential time. Another simple way to prove that it is not 0 would be to provide a counterexample. Assigning $x = 1$ and $y = 1$ yields

$$(-4)^{1012} + 2^{506} - 2^{1012} \stackrel{?}{=} 0$$

While it is easy to see in this case that the left-hand side is not 0, in general the size of the left-hand side can grow exponentially in the input size, which makes this calculation too expensive. However, doing the calculations modulo a small prime number bounds the size of the left hand side. In our case, taking everything mod 3 yields

$$1^{1012} + (-1)^{506} - (-1)^{1012} \stackrel{?}{=} 0 \pmod{3}$$

which is not true.

More general, we consider arithmetic circuits. An arithmetic circuit with input variables $x_1, \dots, x_n$ is a sequence $g_1, \dots, g_m$ such that g_1, g_2 , and g_3 are the constants 0, 1 and -1 , $g_4, \dots, g_{n+3}$ are the input variables $x_1, \dots, x_n$ and for every $i \geq n + 4$ we either have $g_i = g_j + g_k$ or $g_i = g_j g_k$, where $j, k < i$. The output of the circuit is defined as the value of g_m .

Example 1. The polynomial x^{512} can be represented by the arithmetic circuit $g_1 = 0$, $g_2 = 1$, $g_3 = -1$, $g_4 = x$, $g_5 = g_4^2$, $g_6 = g_5^2, \dots, g_{12} = g_{11}^2$.

We can represent g_i as a polynomial $p_i(x_1, \dots, x_n)$ and then ask the question if $g_m = p_m(x_1, \dots, x_n) = 0$ for all $x_1, \dots, x_n$.

The Schwartz-Zippel algorithm is now as follows: Let R and l be parameters,

- $x_i \leftarrow_u \{0, 1, \dots, R\}$ for all $i = 1, \dots, n$
- Pick a random prime p of length l as described in last class
- Compute each $g_i \pmod{p}$ for $i = 1, \dots, m$

- If $g_m = 0 \pmod p$, output “yes”, otherwise output “no”

Note: Non-primes also work, but the proof is more complicated.

Lemma 1. *Let $\deg$ denote the maximum degree in any one variable, then*

$$\deg(p_i) < 2^i \leq 2^m$$

Proof. The degree can at most double with every operation. $\square$

Lemma 2. *If $p(x_1, \dots, x_n)$ is of degree d and $p(x_1, \dots, x_n) \neq 0$, then choosing a_i uniformly from $\{0, 1, \dots, 2dn\}$ gives $P(p(a_1, \dots, a_n) = 0) \leq \frac{1}{2}$.*

Proof. Let

$$\begin{aligned} q_n &= p(x_1, \dots, x_n) \\ q_{n-1} &= p(x_1, \dots, x_{n-1}, a_n) \\ &\vdots \\ q_0 &= p(a_1, \dots, a_n) \end{aligned}$$

If $q_0 = 0$, but $q_n \neq 0$, then there is an i such that $q_i \neq 0$ and $q_{i-1} = 0$.

We can write

$$q_i(x_1, \dots, x_i, a_{i+1}, \dots, a_n) = \sum_{d_1, \dots, d_{i-1}=0}^d x_1^{d_1} \cdots x_{i-1}^{d_{i-1}} (r_{d_1, \dots, d_{i-1}}(x_i))$$

where $r_{d_1, \dots, d_{i-1}}$ is a degree d polynomial in one variable.

The polynomial q_i is nonzero if and only if there are $d_1, \dots, d_{i-1}$, such that $r_{d_1, \dots, d_{i-1}}$ is a nonzero polynomial. A nonzero polynomial $r_{d_1, \dots, d_{i-1}}$ with degree d has at most d roots. Hence, choosing a_i uniformly at random from $\{0, \dots, 2dn\}$ gives

$$P(r_{d_1, \dots, d_{i-1}}(a_i) = 0) \leq \frac{1}{2n}$$

or $P(q_{i-1} = 0 \mid q_i \neq 0) \leq \frac{1}{2n}$. Therefore, we get $P(q_0 = 0) \leq \frac{1}{2}$. $\square$

Applying lemma 1 to lemma 2, we get that we need to choose $R = 2^{m+1}n$.

Lemma 3. *If $0 \leq a_i \leq R$ for all i , then $p_i(a_1, \dots, a_n) \leq R^{(2^i)}$.*

Proof. The value can at most double in every step. $\square$

To bound the probability that we pick a prime p such that $p_m(a_1, \dots, a_n) \neq 0$, but $p_m(a_1, \dots, a_n) = 0 \pmod p$, let $V = R^{(2^m)} \leq 2^{(m+1+\log n)2^m}$ and let k be the number of distinct prime factors of $p_m(a_1, \dots, a_n)$. We have

$$k \leq \log p_m(a_1, \dots, a_n) \leq \log V \leq 2^m(m + \log n + 1)$$

We use that the number of prime numbers of length l is $\Theta\left(\frac{2^l}{l}\right)$. Choosing $l = 3m$, the algorithm picks one of $\frac{2^{3m}}{c3m}$ prime numbers uniformly at random for some c , at most $2^m(m + \log n + 1)$ of which divide $p_m(a_1, \dots, a_n)$. Hence the probability that we pick a divisor of $p_m(a_1, \dots, a_n)$ is bounded by $\text{poly}(m)2^{-2m}$.

We can conclude that the probability of outputting “yes”, if the correct answer is “no” is bounded by $\frac{1}{2} + \text{poly}(m)2^{-2m}$.

2 Max-Cut Problem

Given an undirected graph $G = (V, E)$, the goal is to find a partition of V into two sets L and R such that the number of edges that cross from one set to the other is maximized.

Consider the following simple randomized algorithm: For each node, add it to L or R with probability $\frac{1}{2}$ each. Let Cut denote the size of the resulting cut. We get

$$E[\text{Cut}] = \sum_{e \in E} P(e \text{ is a cut edge}) = \frac{1}{2}|E|$$

Since the max cut is at most $|E|$, this is a $\frac{1}{2}$ -approximation in expectation. Let p be the probability that the cut is indeed at least $\frac{1}{2}|E|$. Then

$$\begin{aligned} \frac{1}{2}|E| = E[\text{Cut}] &\leq (1-p) \left(\frac{1}{2}|E| - 1 \right) + p|E| \\ &= \frac{1}{2}|E| - 1 + p \left(\frac{1}{2}|E| + 1 \right) \end{aligned}$$

Hence $p \geq \frac{1}{1/2|E|+1} = O\left(\frac{1}{|E|}\right)$. Therefore after repeating the algorithm $O(|E|)$ times, with high probability we find a cut which $\frac{1}{2}$ -approximates the max cut.