CSE599s: Extremal Combinatorics November 7, 2011

Lecture 11 Isolation Lemma and Polynomial Based Constructions

Lecturer: Anup Rao

1 The Isolation Lemma

Here we show a beautiful lemma of Mulmuley, Vazirani and Vazirani.

Let $\mathcal{F}$ be a family of sets on the universe $[n]$. Suppose we assign each element $x \in [n]$ a random weight $w(x) \in [T]$. For any set S, define the weight of S to be $w(S) = \sum_{x \in S} w(x)$. Then we have

Lemma 1. $\Pr[\textit{the minimum weight set is not unique}] \leq n/T$.

Proof For any x, define

$$\alpha(x) = \min_{S \in \mathcal{F}, x \notin S} w(S) - \min_{S \in \mathcal{F}, x \in S} (w(S) - w(x)).$$

$\alpha(x)$ depends only on the weights of elements other than x, and is the weight on x needed to make the lightest set that contains x the same weight as the lightest set that does not contain x. Once the weights of all other elements have been assigned, $w(x) < \alpha(x)$ means that any minimum weight set must contain x, $w(x) > \alpha(x)$ means any minimum weight set cannot contain x and $w(x) = \alpha(x)$ means that both a set that contains x and a set that does not contain x can be minimum weight. Observe that if two sets achieve the minimum weight, then there must be some element x which is in one but not in the other. Then it must be that $w(x) = \alpha(x)$. However, $\Pr[x = \alpha(x)] \leq 1/T$, so by the union bound, the probability that this happens for any x is at most n/T. ■

1.1 An application: non-uniform NL is contained in parity-L

Given a Turing machine that has $\log n$ bits of space to work with when operating on an n-bit input, we can simulate it using a finite automaton with n states (one for each setting of the work space). We can obtain a directed graph with the following properties:

- Every vertex has fan-out 2, and every edge is labeled 0 or 1.
- There is a designated start vertex s and a designated accept vertex t.
- Each vertex is labeled by an input variable x_i, or is unlabeled.
- The output of the Turing machine is 1 if and only if there is a an accepting path that is consistent with the bits of the input (namely it follows the out-edge labeled x_i if the corresponding vertex is labeled by the variable x_i).

This characterization means that to prove that nondeterminism does not help machines that use $O(\log n)$ space, it is enough to give a deterministic algorithm that takes as input a directed graph, and outputs 1 exactly when the graph has an $s-t$ path (the graph is obtained by first throwing out the edges which are inconsistent with the input, which can be done in $O(\log n)$ space).

Wigderson showed that you can always map the original automaton to a graph where it is sufficient to tell whether the number of accepting path is even or odd (rather than 0 or > 0).

Theorem 2. *If A is an automaton on n vertices that reads n input bits there is an automaton B on $\mathsf{poly}(n)$ vertices reading n input bits such that for each input x, if A has an accepting path consistent with x, then B has an odd number of accepting paths consistent with x, and if A has no accepting path consistent with x, B has an even number of accepting paths consistent with x.*

Here we sketch the proof.

Let $w : E \to [T]$ be a weight function on the edges E of the original automaton. We shall design an automaton that looks for st paths of weight exactly ℓ. We obtain a new automaton G^ℓ_w as follows. Each vertex u in G is replaced by ℓ vertices $u_0, \ldots, u_{\ell-1}$ in G^ℓ_w. If (u, v) is an edge in the original graph, $(u_i, v_{i+w(u,v)})$ is the corresponding edge of G^ℓ_w. Thus a path from a to b of total weight k is replaced by a path from a_0 to b_k. Let s_0 be the start vertex, and $t_{\ell-1}$ be the accept vertex of the automaton. Observe that if we knew the weight of the minimum weight accepting path, then we could set ℓ to be this weight. This would give us an automaton where there is a single accepting path (if any).

Claim 3. *G^ℓ_w accepts x if and only if there is an $s-t$ path of weight exactly ℓ in the original automaton on input x.*

Pick n weight functions $w_1, \ldots, w_n$ at random. By the isolation lemma, given any fixed input, the shortest $s-t$ path (if any) is unique except with probability n^2/T, since the number of edges is at most n^2. The probability that no weight function gives a unique shortest path is $(n^2/T)^n$. By the union bound, except with probability $2^n(n^2/T)^n$, for every input x, there is some weight function w_i that gives a unique minimum weight $s-t$ path (if one exists).

Claim 4. *If $T > 2n^2$, there is some choice of weight functions $w_1, \ldots, w_n$ such that if x is not accepted by the original automaton, then none of the the automatons $G^\ell_{w_i}$ for $\ell \leq nT$ accepts x, and if x is accepted by the original automaton, then one of the automatons $G^\ell_{w_i}$ has a unique accepting path for x.*

We fix a choice of $w_1, \ldots, w_n$ that satisfies the above claim. Let s^i, t^i be the start and end vertices for each of these n^2T graphs. We generate the final automaton as follows.

1. The start vertex is s^1 and the accept vertex is t^{n^2T}.
2. Identify t^i with s^{i+1} for each i.
3. Add the edges (s^i, t^i) and (s^1, t^{n^2T}) (formally we will need to add dummy vertices to ensure that the fan-out is still 2, but we gloss over these details here).

By construction, if x is not accepted by the original automaton, then there are exactly two accepting paths in the new automaton. One is obtained by following the edge from the start vertex to the accept vertex, and the other is obtained by using the edges (s^i, t^i). On the other hand,

if x is accepted by the original automaton, we get an odd number of accepting paths in the new automaton. This is because for some w_i, ℓ we have exactly one accepting path in $G^{\ell}_{w_i}$, which gives two accepting paths in the corresponding subgraph. This ensures the total number of paths that go through each automaton is even, and the total number of accepting paths is odd.

2 Finite Fields Refresher

We recalled some basic facts about finite fields, most of which I do not reproduce here.

Let $\mathbb{F}$ be any field.

Fact 5. *Given any $d+1$ points, evaluating univariate degree d polynomials on those points defines a linear bijection between the coefficients of the polynomial and the evaluations.*

Proof Since the space of evaluations and the space of polynomials have the same dimension, it will suffice to show that the evaluation map is onto. Given $a_1, \ldots, a_{d+1}$, define the i'th polynomial $p_i = \prod_{j \neq i}(X - a_j)/(a_i - a_j)$. This is a degree d polynomial, and further

$$p_i(a_j) = \begin{cases} 1 & i = j \\ 0 & \text{else.} \end{cases}$$

Thus any target evaluations can be obtained by taking linear combinations of the p_i's. ■

A polynomial is *irreducible* if it cannot be factored.

Fact 6. *If $p(X)$ is an irreducible polynomial, $\mathbb{F}[X]/p$ (namely polynomials mod p) is a finite field of size $|\mathbb{F}|^d$.*

Fact 7. *For a prime p, let $\mathbb{F}_p$ denote the integers mod p. Up to isomorphism the only finite fields are the ones of size p^d obtained as $\mathbb{F}_p[X]/q(X)$ for some irreducible degree d polynomial $q(X)$.*

3 Constructions Based on Polynomials

3.1 Pairwise Independent Bits

Goal: A small set $S \subset \{0,1\}^n$ such that for a uniformly random $X \in S$, for each $i \neq j$, X_i is independent of X_j and uniform.

We work with a finite field of size $2^t \geq n$. Each element of S is obtained by picking a degree one polynomial (i.e. line) $p(X) = a + bX$. We evaluate this polynomial on n distinct field elements to obtain $p(\alpha_1), \ldots, p(\alpha_n)$. By Fact 5, if a, b are uniform, then any two of these evaluations are uniform and independent. To obtain pairwise independent bits, simply encode each field element with a bit string.

This construction gives a set of size n^2.