Notes on Complexity Theory Last updated: October, 2011

Lecture 17

Jonathan Katz

1 Graph Non-Isomorphism is in AM

The proof system we showed earlier for graph non-isomorphism relied on the fact that the verifier's coins are kept hidden from the prover. Is this inherent? Somewhat surprisingly, we now show a *public-coin* proof for graph non-isomorphism. Before doing so, we take a brief detour to discuss *pairwise-independent hash functions* (which are useful in many other contexts as well).

1.1 Pairwise-Independent Hash Functions

Fix some domain D and range R. Let $\mathcal{H} = \{h_k\}_{k \in K}$ be a family of functions, where each $k \in K$ defines a function $h_k : D \to R$. We say that $\mathcal{H}$ is[1] *pairwise independent family* if for all distinct $x, x' \in D$ and all (not necessarily distinct) $y, y' \in R$ we have

$$\Pr_{k \leftarrow K}\left[h_k(x) = y \bigwedge h_k(x') = y'\right] = 1/|R|^2 .$$

Put differently, let $D = \{x_1, \ldots, x_\ell\}$ and consider the random variables $Y_i = h_K(x_i)$ (where K is uniform). If $\mathcal{H}$ is pairwise independent then each Y_i is uniformly distributed, and moreover the random variables $Y_1, \ldots, Y_\ell$ are pairwise independent; i.e., for any $i \neq j$ the random variables Y_i and Y_j are independent.

We show a simple construction of a pairwise-independent family for $D = R = \mathbb{F}$, where $\mathbb{F}$ is any finite field. Setting $\mathbb{F} = GF(2^n)$, and viewing strings of length n as field elements, we obtain a construction with $D = R = \{0,1\}^n$. By truncating the output, we obtain a construction with $D = \{0,1\}^n$ and $R = \{0,1\}^\ell$ for any $n \geq \ell$. By padding the input with 0s, we obtain a construction for any $\ell \geq n$.

Fix $D = R = \mathbb{F}$ and let $\mathcal{H} = \{h_{a,b}\}_{a,b \in \mathbb{F}}$ where $h_{a,b}(x) = ax + b$. We claim that $\mathcal{H}$ is pairwise independent. Indeed, fix any distinct $x, x' \in \mathbb{F}$ and any $y, y' \in \mathbb{F}$, and consider the probability (over choice of a, b) that

$$\begin{aligned} y &= ax + b \\ y' &= ax' + b. \end{aligned}$$

Using some basic algebra, we see that the above equations are true iff

$$\begin{aligned} a &= (y - y') \cdot (x - x')^{-1} \\ b &= y - (y - y') \cdot (x - x')^{-1} \cdot x. \end{aligned}$$

(Note that the above rely on the fact that $x \neq x'$.) Since x, x', y, y' are fixed, the right-hand sides of the above equations are some fixed elements in $\mathbb{F}$; hence, the probability that a, b satisfy both equations is exactly $1/|\mathbb{F}|^2$ as required.

[1] Frequently, terminology is abused and $h_k \in \mathcal{H}$ is called a pairwise-independent hash function. Formally, it only makes sense to speak about pairwise independent *families* of functions.

For applications, what we actually need are *ways to construct* pairwise-independent families on, say, $\{0,1\}^n$ for some given n. In that case we actually want an efficient probabilistic algorithm that, given n, outputs a key k that, in turn, defines a function $h_k : \{0,1\}^n \to \{0,1\}^n$ that is efficiently computable. The construction given above satisfies this, though it is not entirely trivial to show this. (In particular, we need to use the fact that we can efficiently generate, and manipulate elements of, $GF(2^n)$.)

1.2 An AM Protocol for Graph Non-Isomorphism

We begin by introducing some more notation. For an n-vertex graph G (represented as an adjacency matrix), consider the (multi-)set $\mathsf{all}(G) = \{\pi_1(G), \ldots, \pi_{n!}(G)\}$ of all permuted versions of G. This is indeed a multi-set (in general) since it is possible that $\pi_i(G) = \pi_j(G)$ even when $\pi_i \neq \pi_j$. For example, consider the 3-vertex graph G in which there is a single edge $(1,2)$. Considering the 6 possible permutations on the labels of the vertices, we see that $\pi = (12)(3)$ maps G to itself, even though π is not the identity permutation. On the other hand, $\pi' = (13)(2)$ maps G to a graph isomorphic, but not identical, to G.

Let $\mathsf{aut}(G) = \{\pi \mid \pi(G) = G\}$; these are the *automorphisms* of G. (Note that $\mathsf{aut}(G)$ is never empty, since the identity permutation is always in $\mathsf{aut}(G)$.) Let $\mathsf{iso}(G)$ be the *set* (not multi-set) $\{\pi(G) \mid \pi \text{ is a permutation}\}$. We claim that for any n-vertex graph G we have:

$$|\mathsf{aut}(G)| \cdot |\mathsf{iso}(G)| = n! \; .$$

The reason is that our original multi-set $\mathsf{all}(G)$ has exactly $n!$ elements in it, but each graph in $\mathsf{iso}(G)$ appears exactly $\mathsf{aut}(G)$ times in $\mathsf{all}(G)$ (because $|\mathsf{aut}(G)| = |\mathsf{aut}(\pi(G))|$ for any permutation π).

We now have the ideas we need to describe the proof system. Given graphs (G_0, G_1), define the set W as follows:

$$W = \left\{ (H,\sigma) \mid \begin{array}{c} H \text{ is isomorphic to either } G_0 \text{ or } G_1 \\ \text{and } \sigma \in \mathsf{aut}(H) \end{array} \right\}.$$

Note that if $G_0 \cong G_1$, then H is isomorphic to G_0 iff it is isomorphic to G_1; also, the number of automorphisms of any such H is exactly $|\mathsf{aut}(G_0)|$. So the size of W is exactly $|\mathsf{iso}(G_0)| \cdot |\mathsf{aut}(G_0)| = n!$. On the other hand, if $G_0 \ncong G_1$ then the graphs isomorphic to G_0 are distinct from those graphs isomorphic to G_1. So the size of W in this case is

$$|\mathsf{iso}(G_0)| \cdot |\mathsf{aut}(G_0)| + |\mathsf{iso}(G_1)| \cdot |\mathsf{aut}(G_1)| = 2n! \; .$$

So, $|W \times W| = (n!)^2$ if $G_0 \cong G_1$ and $|W \times W| = 4 \cdot (n!)^2$ if $G_0 \ncong G_1$. Furthermore, it is possible to prove membership in W by giving an isomorphism to either G_0 or G_1 (the automorphism can be verified in polynomial time).

The above suggests the following proof system:

1. On common input (G_0, G_1), define $W \times W$ as above. (Arthur obviously cannot construct $W \times W$, but all it needs to do is compute the upper bound $4(n!)^2$ on its size.) Let $m = \log 4(n!)^2$, and note that m is polynomial in the input size n.

2. Arthur selects a random h from a pairwise-independent family, where h maps strings of the appropriate length (which will become obvious in a minute) to $\{0,1\}^m$. It sends h to Merlin.

3. Merlin finds an $x \in W \times W$ such that $h(x) = 0^m$ (if one exists). It sends this x to Arthur, along with a proof that $x \in W \times W$.

4. Arthur outputs 1 if $x \in W \times W$ and $h(x) = 0^m$.

We now analyze the above. Say (G_0, G_1) are isomorphic. Then $|W \times W| = (n!)^2$ and so

$$\begin{aligned} \Pr_h[\exists x \in W \times W : h(x) = 0^m] &\leq \sum_{x \in W \times W} \Pr_h[h(x) = 0^m] \\ &= (n!)^2 \cdot 2^{-m} = 1/4, \end{aligned}$$

and so Merlin convinces Arthur only with probability at most 1/4. On the other hand, if $G_0 \not\cong G_1$ then $|W \times W| = 4(n!)^2$ and we can bound the desired probability as follows:

$$\begin{aligned} \Pr_h[\exists x \in W \times W : h(x) = 0^m] &\geq \sum_{x \in W \times W} \Pr_h[h(x) = 0^m] \\ &\quad - \frac{1}{2} \cdot \sum_{\substack{x,y \in W \times W \\ x \neq y}} \Pr_h[h(x) = 0^m \wedge h(y) = 0^m] \\ &> 1 - \frac{1}{2} \cdot (4(n!)^2)^2 \cdot (2^{-m})^2 = 1/2\,, \end{aligned}$$

using the inclusion-exclusion principle for the first inequality, and relying on pairwise independence in the second step. (A better bound can be obtained using Chebyshev's inequality.)

The above does not have perfect completeness, but we have seen before that this can be fixed.

1.3 Evidence that Graph Isomorphism is not $\mathcal{NP}$-Complete

Let GI be the language of graph isomorphism, and GNI be the language of graph non-isomorphism. In the previous section we showed $GNI \in \mathbf{AM}$. This gives evidence that GI is *not* $\mathcal{NP}$-complete.

Theorem 1 *If GI is $\mathcal{NP}$-complete, then the polynomial hierarchy collapses (specifically, $\mathsf{PH} = \Sigma_2$).*

Proof We first observe that $\mathbf{AM} \subseteq \Pi_2$ (why?). Now, assume GI is $\mathcal{NP}$-complete. Then GNI is $\mathsf{co}\mathcal{NP}$-complete and hence (since $GNI \in \mathbf{AM}$) we have $\mathsf{co}\mathcal{NP} \subseteq \mathbf{AM}$. We show that this implies $\Sigma_2 \subseteq \mathbf{AM} \subseteq \Pi_2$ and hence $\mathsf{PH} = \Sigma_2$.

Say $L \in \Sigma_2$. Then by definition of Σ_2, there is a language $L' \in \Pi_1 = \mathsf{co}\mathcal{NP}$ such that: (1) if $x \in L$ then there exists a y such that $(x, y) \in L'$, but (2) if $x \notin L$ then for all y we have $(x, y) \notin L'$. This immediately suggests the following proof system for L:

1. Merlin sends y to Arthur.

2. Arthur and Merlin then run an **AM** protocol that $(x, y) \in L'$ (this is possible precisely because $L' \in \mathsf{co}\mathcal{NP} \subseteq \mathbf{AM}$).

The above is an **MAM** proof system for L. But, as we have seen, this means there is an **AM** proof system for L. Since $L \in \Sigma_2$ was arbitrary this means $\Sigma_2 \subseteq \mathbf{AM}$, completing the proof. ■