If GI is NP-complete then $\Sigma_3^{\mathrm{p}} = \Pi_3^{\mathrm{p}}$
Exposition by William Gasarch

1 Introduction

Our goal is to show that if GI is NP-complete then $\Sigma_3^{\mathrm{p}} = \Pi_3^{\mathrm{p}}$. We will do this in several parts which we explain here informally.

1. If a prover wanted to convince a verifier that $G_1 \equiv G_2$ then this is easy: just give the verifier the isomorphism. What if a prover wanted to convince a verifier that $G_1 \not\equiv G_2$? We show that if we allow the verifier and prover a public source of random coins and we allow two rounds of interaction (that is, verifier sends to prover, and prover responds) then there is an interactive protocol where the following holds.

$$\begin{array}{ll} (G_1, G_2) \in \overline{GI} & \rightarrow \Pr(\text{Protocol returns YES}) = 1 \\ (G_1, G_2) \notin \overline{GI} & \rightarrow \Pr(\text{Protocol returns YES}) \leq \frac{1}{4} \end{array}$$

   This is written as $\overline{GI} \in \mathrm{AM}$. (This will stand for Arthur-Merlin.)

2. We show that

$$\mathrm{AM} \subseteq \mathrm{NP/poly}.$$

3. Combining items 1 and 2 above we get

$$\overline{GI} \in \mathrm{NP/poly}.$$

4. If GI is NP-complete then

$$SAT \leq_{\mathrm{m}}^{\mathrm{p}} GI$$

   so

$$TAUT \leq_m^{\mathrm{p}} \overline{GI}.$$

   Since

$$\overline{GI} \in \mathrm{NP/poly},$$
$$TAUT \in \mathrm{NP/poly}.$$

   We showed in the notes on sparseness that

$$TAUT \in \mathrm{NP/poly} \rightarrow \Sigma_3^{\mathrm{p}} = \Pi_3^{\mathrm{p}}.$$

   Hence we have

$$\Sigma_3^{\mathrm{p}} = \Pi_3^{\mathrm{p}}.$$

5. We showed in the notes on P, NP, and PH that

$$\Sigma_3^{\mathrm{p}} = \Pi_3^{\mathrm{p}} \rightarrow PH == \Sigma_3^{\mathrm{p}} = \Pi_3^{\mathrm{p}}.$$

6. Putting this all together we get

$$GI \text{ NPC } \rightarrow PH == \Sigma_3^{\mathrm{p}} = \Pi_3^{\mathrm{p}}.$$

Two types of interactive proof systems have been defined: (1) Goldwasser, Micali, and Rackoff [3] defined interactive protocols with *private* coins, and (2) Babai [1, 2] defined interactive protocols with *public* coins. It was first shown that with private coins $\overline{GI}$ had an interactive protocol of two rounds. Goldwasser and Sipser [4] showed that public coins were equivalent to private coins, and hence there is an interactive protocol for $\overline{GI}$ with public coins. A more direct proof appears in [6]. We present a version of that proof, from [5], here.

2 The Class Arthur-Merlin

NP can be viewed as an interaction between a prover and a verifier. Both the prover and the verifier have access to the input x. The prover (who is all powerful) sends the verifier a witness y. The verifier then verifies that y is evidence that $x \in A$. In this spirit we give an alternative definition of NP.

Def 2.1 $A \in$ NP if there exists a set $B \in P$ such that the following holds.

$$\begin{aligned} x \in A &\rightarrow (\exists^p y)[(x,y) \in B] \\ x \notin A &\rightarrow (\forall^p y)[(x,y) \notin B] \end{aligned}$$

We want to modify the roles of the verifier (who we will call Arthur) and the prover (who we will call Merlin). Picture the following. On input x Arthur sends a random sequence r to Merlin who then tries to send evidence that $x \in A$. This evidence is a message that depends on x and r. Upon receiving the message, Arthur tries to use this to verify that $x \in A$. He does this in deterministic poly time.

Note that we do not need Arthur to send the random bits—they could come from an independent source. We also do not even need Merlin— all we need is that if $x \in A$ then evidence of this probably exists, and if $x \notin A$

then evidence that $x \in A$ probably does not exist. In the formal definition we take this view. That is, the formal definition does not mention Arthur sending bits or Merlin producing evidence. Our protocol that $\overline{GI} \in$ AM *will* use the intuition of Arthur sending bits and Merlin responding; however, that protocol can easily be translated into the formal framework we define here.

In the literature these are called "Arthur-Merlin games" where the intuition is that Merlin is trying to convince Arthur that $x \in A$. Merlin is all powerful, Arthur is just poly time and coins. These are abbreviated 'AM games'. The order of the letters matters— Arthur goes first, then Merlin goes.

Def 2.2 A set A is in AM if there exists polynomials p, q, and a set $B \in P$ such that the following hold

1. The domain of B is $\bigcup_{n=0}^{\infty}\{0,1\}^n \times \{0,1\}^{q(n)} \times \{0,1\}^{p(n)}$. (Arthur and Merlin both have x. Arthur challenges Merlin to show that $x \in A$ by sending him a random string $r \in \{0,1\}^{p(n)}$. Merlin responds to the challenge with $y \in \{0,1\}^{q(n)}$. Upon receiving y Arthur computes if $(x,y,r) \in B$. If this yields YES then Arthur is convinced that $x \in A$. If this yields NO then Arthur is not convinced. He may err with a small probability.)

2. Let x be of length n.

   $x \in A \rightarrow \Pr_{|r|=p(n)}((\exists y, |y| = q(n))[(x,y,r) \in B]) \geq \frac{3}{4}$.

   $x \notin A \rightarrow \Pr_{|r|=p(n)}((\exists y, |y| = q(n))[(x,y,r) \in B]) \leq \frac{1}{4}$.

Exercise 1

1. Show that if you replace the $\frac{3}{4}$ with $\frac{99}{100}$ and the $\frac{1}{4}$ with $\frac{1}{100}$ in the above definition, you still get AM.

2. Show that if you replace the $\frac{3}{4}$ with $1 - \frac{1}{2^{|x|}}$ and the $\frac{1}{4}$ with $\frac{1}{2^{|x|}}$ in the above definition, you still get AM.

3 Some Graph Theory

From now on n is the number of vertices of the graph in question!!!

Def 3.1 Graphs $G = (V_1, E_1)$ and $H = (V_2, E_2)$ are *isomorphic* if there exists a bijection $f : V_1 \to V_2$ such that $(a, b) \in E_1$ iff $(f(a), f(b)) \in E_2$.

Notation 3.2 If G_1 and G_2 are graphs then $G_1 \equiv G_2$ means that G_1 and G_2 are isomorphic.

Let $G = (V, E)$ be a graph. We want to look at what happens when we relabel the graph by permuting the vertices. This leads to $n!$ graphs. How many of them are really different?

Def 3.3 Let $G = (V, E)$ be a graph. An *automorphism of* G is a bijection $\sigma : V \to V$ such that, for all $x, y \in V$, if $(x, y) \in E$ then $(\sigma(x), \sigma(y)) \in E$.

Def 3.4 A *permutation of* n *elements* is a bijection of the elements to themselves. S_n is the set of all such permutations. $AUT(G)$ is the set of all automorphisms of G. If $V = \{1, \ldots, n\}$ then $AUT(G) \subseteq S_n$. Both $AUT(G)$ and S_n are groups. (We do not use this fact.)

Def 3.5 Let $G = (V, E)$ and $V = \{1, \ldots, n\}$. Let σ be a permutation of V. (We view σ as a bijection from V to V.) The graph $\sigma(G)$ is (V, E') where

$$E' = \{(\sigma(x), \sigma(y) : (x, y) \in E\}.$$

Fact 3.6 *If G is a graph and σ is an automorphism of that graph then G and $\sigma(G)$ are the same graph (not just isomorphic, they are the exact same).*

Lemma 3.7 *If G is a graph on n vertices then*

$$|\{\tau(G) : \tau \in S_n\}| = \frac{n!}{|AUT(G)|}.$$

(HOW TO VIEW THIS: the set is a SET not a MULTISET. Sometimes $\tau(G)$ will really EQUAL G (not isom, actually EQUAL). We DO NOT count this twice.)

Proof: Let $AUT(G) = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}$. The multiset
$B = \{\sigma(G) : \sigma \in S_n\}$
has $n!$ elements in it. Let the elements be $G_1, \ldots, G_{n!}$. Note that if G_i and G_j are the same graph iff there exists an automorphism σ such that $\sigma(G_i) = G_j$. Hence the only graphs that are the same as G_i are

$$\{\sigma_1(G_i), \ldots, \sigma_m(G_i)\}.$$

Therefore for every graph in B there are $|AUT(G)|$ graphs that are the same as it. This results in a partition of the multiset B into parts of size $|AUT(G)|$. Hence the number of parts, which is the number of distinct graphs, is $\frac{n!}{|AUT(G)|}$. ∎

The following sets will be useful.

Def 3.8 Let G, G_1, G_2 be graphs on n vertices.

1. $Y(G) = \{(H, \sigma) : G \equiv H \wedge \sigma \in AUT(G)\}$. Note that there are $\frac{n!}{|AUT(G)|}$ choices for H and $|AUT(G)|$ choices for σ. Hence $|Y(G)| = n!$.

2. $Y(G_1, G_2) = Y(G_1) \cup Y(G_2)$. Note that

$$|Y(G_1, G_2)| = \begin{cases} n! & \text{if } G_1 \equiv G_2; \\ 2n! & \text{if } G_1 \not\equiv G_2. \end{cases}$$

   Note that if $G_1 \equiv G_2$ then $Y(G_1, G_2)$ is small, whereas if $G_1 \not\equiv G_2$ then $Y(G_1, G_2)$ is large. This is silly- in the first case the size is $n!$ and in the second it is $2n!$. However we want to make it non-silly. We want to increase the difference. Hence in the next definition we do this formally.

3. Let $X(G_1, G_2) = Y(G_1, G_2) \times \cdots \times Y(G_1, G_2)$ (there are n copies of $Y(G_1, G_2)$. We use n copies to help us in the calculation in Theorem 5.1.) Note that

$$|X(G_1, G_2)| = \begin{cases} (n!)^n & \text{if } G_1 \equiv G_2; \\ 2^n (n!)^n & \text{if } G_1 \not\equiv G_2. \end{cases}$$

We can now rethink the $\overline{GI}$ problem—Merlin has to convince Arthur that the set $X(G_1, G_2)$ is 'large'- that is- size $2^n(n!)^n$ rather than $(n!)^n$. How can Merlin convince Arthur that X is large? Remember- we are computer

scientists! We can use Hash Functions! If a hash function on X has lots of collisions then X must be large! We will refine this notion.

We will want to represent the elements in $X(G_1, G_2)$. How long is that representation?

1. A graph takes $\Theta(n^2)$ bits to represent.
2. An automorphism takes $\Theta(n \log n)$ bits to represent.
3. Every element in $Y(G_1, G_2)$ takes $\Theta(n^2 + n \log n) = \Theta(n^2)$ bits to represent.
4. Every element in $X(G_1, G_2)$ takes $\Theta(n(n^2)) = \Theta(n^3)$ bits to represent.

We leave the following easy lemma to the reader.

Lemma 3.9 *There exists a polynomial predicate R and a polynomial p such that, for all n, for all graphs G, H on n vertices, for all σ*

$z \in X(G, H) \rightarrow (\exists y)[|y| = p(n) \wedge R(G, H, z, y)]$.

(In essence, if Arthur has G, H, z and $z \in X(G, H)$, then there is a short proof that $z \in X(G, H)$.)

4 Hash Functions

Say a set $X \subseteq \{0,1\}^N$ is 'large'. If we pick a *random* hash function $h : \{0,1\}^N \rightarrow \{0,1\}^k$ (where $k << N$) then it is 'likely' that **many** elements of X will map to 0^k. Say a set $X \subseteq \{0,1\}^N$ is 'small'. If we pick a random hash function $h : \{0,1\}^N \rightarrow \{0,1\}^k$ (where $k << N$) then it is 'likely' that **few** elements of X will map to 0^k.

For Merlin to prove to Arthur that X is large they will do the following: Arthur will pick a random hash function $h : \{0,1\}^N \rightarrow \{0,1\}^k$ (we will specify N and k later) and Merlin will try to find several elements $x \in X$ such that $h(x) = 0^k$.

Arthur will pick a random hash function by picking kN random bits and putting them into an $k \times N$ matrix. Call this matrix C. Let $h : \{0,1\}^N \rightarrow \{0,1\}^k$ be the function $h(x) = Cx$ (multiplying the matrix C by the vector x). All of the arithmetic is done mod 2.

Def 4.1

1. A *sample space* is the set of things that could happen. In our case it will be the set of possible hash functions that could be produced.

2. A *random variable* is a mapping from the sample space to numbers. In our case it will be mapping the hash function h to the number $|\{x : h(x) = 0^k\}|$.

3. If S is a random variable then $E(S)$ is its expected value and $Var(S)$ is its variance, which is defined as $E((S - E(S))^2)$. This is known to equal $E(S^2) - E(S)^2$.

Lemma 4.2 *Let $k, N \in \mathsf{N}$. Let $X \subseteq \{0,1\}^n$. Assume $0^N \notin X$. Consider the following random variable: Pick a random $k \times n$ 0-1 valued matrix M.*

$$S = |\{x \in X : M(x) = 0^k\}|.$$

Output S. Then $E(S) = 2^{-k}|X|$ and $Var(S) \leq 2^{-k}|X|$. (Note that neither $E(S)$ nor $Var(S)$ depends on n, just on k and $|X|$.)

Proof: Before looking at $E(S)$ and $Var(S)$ we will need to look at E of some easier random variables

Let $x, y \in X$. Let R_x be the random variable

$R_x = \begin{cases} 1 & \text{if } h(x) = 0^k; \\ 0 & \text{if } h(x) \neq 0^k. \end{cases}$

Let R_y be similar.

Let $h_i(x)$ be the ith bit of the vector $h(x)$.

$$\begin{aligned} E(R_x) &= \Pr(h(x) = 0^k) \cdot 1 + \Pr(h(x) \neq 0^k) \cdot 0; \\ E(R_x) &= \Pr(h(x) = 0^k); \\ E(R_x) &= \textstyle\prod_{i=1}^{k} \Pr(h_i(x) = 0). \end{aligned}$$

Recall that x is fixed and that $x \neq 0^N$. The probability that $h_i(x) = 0$ can be phrased as follows: What is the probability that a randomly chosen y will make $x \cdot y \equiv 0 \pmod 2$? An easy exercise shows that this is $\frac{1}{2}$. Hence $E(R_x) = \prod_{i=1}^{k} \Pr(h_i(x) = 0) = \frac{1}{2^k}$.

The exact same calculation shows that $E(R_x^2) = \frac{1}{2^k}$. (For any 0-1 valued random variable Z, $Z = Z^2$, hence $E(Z) = E(Z^2)$.)

We now compute $E(R_x R_y)$.

$$\begin{aligned} E(R_xR_y) = \; & \Pr(h(x)=1 \wedge h(y)=1)\cdot 1 + \\ & \Pr(h(x)=0)\Pr(h(y)=1)\cdot 0 + \\ & \Pr(h(x)=1)\Pr(h(x)=0)\cdot 0 + \\ & \Pr(h(x)=0)\Pr(h(x)=0)\cdot 0 \\ = \; & \Pr(h(x)=1)\Pr(h(y)=1) \\ = \; & \tfrac{1}{2^k}\tfrac{1}{2^k} = \tfrac{1}{4^k} \end{aligned}$$

We are now ready to tackle $E(S)$ and $Var(S)$. Note that $S = \sum_{x\in X} R_x$.
$E(S) = E(\sum_{x\in X} R_x) = \sum_{x\in X} E(R_x) = \frac{1}{2^k}|X|$.
We now look at $Var(S)$. Recall that $Var(S) = E(S^2) - (E(S))^2$.

$$\begin{aligned} E(S^2) = \; & E((\textstyle\sum_{x\in X} R_x)(\sum_{y\in X} R_y)); \\ = \; & \textstyle\sum_{x\in X}\sum_{y\in X} E(R_xR_y); \\ = \; & \textstyle\sum_{x\in X} E(R_x^2) + \sum_{x\neq y} E(R_xR_y); \\ = \; & \textstyle\sum_{x\in X} \frac{1}{2^k} + \sum_{x\neq y} \frac{1}{4^k}; \\ = \; & \tfrac{1}{2^k}|X| + \tfrac{1}{4^k}|X|(|X|-1); \end{aligned}$$

$$\begin{aligned} Var(S) = \; & E(S^2) - (E(S))^2 \\ = \; & \tfrac{1}{2^k}|X| + \tfrac{1}{4^k}|X|(|X|-1) - \tfrac{1}{4^k}|X|^2 \\ = \; & \tfrac{1}{2^k}|X| + \tfrac{1}{4^k}|X|^2 - \tfrac{1}{4^k}|X| - \tfrac{1}{4^k}|X|^2 \\ = \; & \tfrac{1}{2^k}|X| - \tfrac{1}{4^k}|X| \\ \leq \; & \tfrac{1}{2^k}|X| \end{aligned}$$

∎

5 $\overline{GI} \in$ AM

We code elements of $X(G_1, G_2)$ as strings of length N. By the note at the end of Section 3, $N = \Theta(n^3)$. We leave it to the reader to devise a coding system. Make sure that $0^N \notin X(G_1, G_2)$. We will let $k = \lceil \log((n!)^n) \rceil$ throughout. We will use that $2^k = \Theta((n!)^n)$. Let S be the random variable from Lemma 4.2 with $X = X(G_1, G_2)$. Note the following:

1. If $G_1 \equiv G_2$ then $|X(G_1, G_2)| = (n!)^n$. Hence

$$E(S) = (n!)^n/2^k = \Theta(1).$$

and

$$Var(S) \leq \Theta(1).$$

2. If $G_1 \not\equiv G_2$ then $|X(G_1, G_2)| = 2^n(n!)^n$. Hence $E(S) = 2^n(n!)^n/2^k = \Theta(2^n)$ and $Var(S) \leq \Theta(2^n)$.

Theorem 5.1 $\overline{GI} \in$ AM.

Proof:

We present the protocol and show that it works.

1. Input(G_1, G_2). (Both Merlin and Arthur see this.) Let N be the exact length of an element of $X(G_1, G_2)$. Recall that N is $\Theta(n^3)$.

2. Arthur sends Merlin an (randomly chosen) $N \times k$ matrix of 0's and 1' M. Since $X(G_1, G_2) \subseteq \{0,1\}^N$ we use M as a function from $X(G_1, G_2)$ to $\{0,1\}^k$.

3. Merlin sends Arthur $z_1, \ldots, z_n \in \{0,1\}^N$. For each z_i he also sends back proof that $z_i \in X(G_1, G_2)$ in the form of a string y_i of length $p(n)$ such that $R(z_i, y_i, G_1, G_2)$. (The p and R are from Lemma 3.9.) Merlin's intent is that all the z_i are in $X(G_1, G_2)$ and they all map to 0^k.

4. For each i, Arthur checks if that $R(z_i, y_i, G_1, G_2)$ is true and that $M(z_i) = 0^k$. If for any i either of these fails then output NO. Else output YES.

We show that if $G_1 \equiv G_2$ then it is unlikely that Merlin can come up with n elements that map to 0^k, whereas if $G_1 \not\equiv G_2$ then it will be quite likely.

We use the following lemma which is Chebyshev's inequality. We do not present a proof.

Lemma 5.2 *If S is any random variable and $a > 0$ then*
$\Pr(|S - E(S)| \geq a) < \frac{Var(S)}{a^2}$.

Intuitively this is saying that the probability that S is far away from $E(S)$ is small, and how small depends on $Var(S)$.

There are two cases of the protocol to consider.

Case 1: $G_1 \equiv G_2$. We need to show that Merlin will have a hard time fooling us. Let

$$S = \{x : M(x) = 0^k\}.$$

We show that for most choices of M there do not exist n elements in S (that is, there do not exists n elements of $X(G_1, G_2)$ that map to 0^k.) Since $G_1 \equiv G_2$ we have that $E(S) = \Theta(1)$ and $Var(S) \leq \Theta(1)$. We want to bound $\Pr(S \geq n)$. We want to phrase this as $\Pr(|S - E(S)| \geq a)$ for some a so that we can use Chebyshev's inequality.

$$\begin{aligned} S &\geq n \\ S - E(S) &\geq n - E(S) \\ |S - E(S)| &\geq |n - E(S)| \\ |S - \Theta(1)| &\geq |n - \Theta(1)| \end{aligned}$$

We now apply Chebyshev's inequality on S with $a = n - \Theta(1)$.

$$\begin{aligned} \Pr(S \geq n) \leq &\ \Pr(|S - E(S)| \geq n - \Theta(1)); \\ \leq &\ \tfrac{Var(S)}{(n-\Theta(1))^2}; \\ \leq &\ \tfrac{\Theta(1)}{(n-\Theta(1))^2}; \end{aligned}$$

For large enough n, $\Pr(S \geq n) \leq \frac{1}{4}$. Hence we have

$$G_1 \equiv G_2 \rightarrow \Pr(S \geq n) \leq \frac{1}{4}.$$

Case 2: $G_1 \not\equiv G_2$. We need to show that Merlin can convince us that $X(G_1, G_2)$ is large. More precisely, we need to show that for most choices of M there do exist n elements in S (that is, there do exists n elements of $X(G_1, G_2)$ that map to 0^k.) Since $G_1 \not\equiv G_2$ we have that $E(S) = 2^n$ and $Var(S) \leq 2^n$. We want to phrase $\Pr(S \leq n-1)$ in terms of $\Pr(|S-E(S)| \geq a)$ for some a.

$$\begin{aligned} S \leq &\ n - 1 \\ S - E(S) \leq &\ n - 1 - E(S) \\ E(S) - S \geq &\ E(S) - n + 1 \\ |E(S) - S| \geq &\ 2^n - n + 1 \\ |E(S) - S| \geq &\ 2^n - n \end{aligned}$$

$$\begin{aligned} \Pr(S \leq n-1) \leq &\ \Pr(|S - E(S)| \geq 2^n - n); \\ \leq &\ \tfrac{Var(S)}{(2^n-n)^2}; \\ \leq &\ \tfrac{2^n}{(2^n-n)^2}; \end{aligned}$$

For large n this is $\leq \frac{1}{4}$. Hence the probability that there are $\leq n-1$ elements in S is $\leq \frac{1}{4}$. Therefore the probability that there are n elements of S is $\geq 1 - \frac{1}{4} = \frac{3}{4}$. Hence we have

$$G_1 \not\equiv G_2 \rightarrow \Pr(S \leq n-1) \geq \frac{3}{4}.$$

∎

6 AM $\subseteq$ NP/poly

We need the following which we leave as an exercise.

Notation 6.1 If X and Y are sets then $X \Delta Y$ is $(X - Y) \cup (Y - X)$. Note that $X \Delta Y$ is the set of elements where X and Y differ.

Lemma 6.2 *Let $A, A' \subseteq \{0,1\}^*$. If there exists a polynomial s such that, for all n,*

$$|(A \cap \{0,1\}^{\leq n}) \Delta (A' \cap \{0,1\}^{\leq n}| \leq s(n)$$

then $A \in \Sigma_i^{\mathrm{p,SPARSE}}$ iff $A' \in \Sigma_i^{\mathrm{p,SPARSE}}$. (We are saying that if A and A' only differ by a polynomial amount on each length n, then A and A' are similar enough that they are either both in $\Sigma_i^{\mathrm{p,SPARSE}}$ or both not in $\Sigma_i^{\mathrm{p,SPARSE}}$.)

We can now prove our main theorem.

Theorem 6.3 AM $\subseteq$ NP/poly.

Proof: Let $A \in$ AM via p, q, B.

By Exercise 1 there exists polynomials p, q,and a poly predicate V such that the following hold

1. The domain of V is $\{0,1\}^n \times \{0,1\}^{q(n)} \times \{0,1\}^{p(n)}$. The range is $\{0,1\}$.
2. Let x be of length n.
   $x \in A \rightarrow \Pr_{|r|=p(n)}((\exists y, |y| = q(n))[V(x,y,r) = 1]) \geq 1 - \frac{1}{2^n}$.
   $x \notin A \rightarrow \Pr_{|r|=p(n)}((\forall y, |y| = q(n))[V(x,y,r) = 1]) \leq \frac{1}{2^n}$.

For each $x \in A$, $|x| = n$, there are at least $(1 - \frac{1}{2^n})2^{p(n)}$ strings $r \in \{0,1\}^{p(n)}$ such that

$(\exists y, |y| = q(n))[V(x, y, r)]$.

For each $x \notin A$, $|x| = n$, there are at least $(1 - \frac{1}{2^n})2^{p(n)}$ strings $r \in \{0,1\}^{p(n)}$ such that

$(\forall y, |y| = q(n))[\neg V(x, y, r)]$.

In either case we say that the string r is GOOD FOR x in that it yields CORRECT information about x.

Consider the following thought experiment. Picture a $2^n \times 2^{p(n)}$ array such that the following hold.

1. The rows are indexed by $x \in \{0,1\}^n$.
2. The columns are indexed by $r \in \{0,1\}^{p(n)}$.
3. We put GOOD in the (x, r) entry if r is GOOD for x.
4. We put BAD in the (x, r) entry if r is NOT GOOD for x.

Here is a sketch of what it might look like

	$0^{p(n)}$	$0^{p(n)-1}1$	$\cdots$	$1^{p(n)}$
0^n	$GOOD$	BAD	$\cdots$	$GOOD$
$0^{n-1}1$	$GOOD$	$GOOD$	$\cdots$	BAD
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1^n	BAD	$GOOD$	BAD	$GOOD$

(The $\cdots$ and $\vdots$ are not meant to indicate any pattern.)

How many GOOD's are in this array? For each $x \in \{0,1\}^n$ there are at least $(1 - \frac{1}{2^n})2^{p(n)}$ GOODS. Hence there are at least $2^n(1 - \frac{1}{2^n})2^{p(n)} = (2^n - 1)2^{p(n)}$ GOODS. Since there are $2^{p(n)}$ columns there must be at least one column with at least $\frac{(2^n-1)2^{p(n)}}{2^{p(n)}} = 2^n - 1$ GOODS. Let r_0 be the index of the column that has $2^n - 1$ GOODS. Let x_0 be the one (if it exists) row such that (x_0, r_0) is labeled BAD.

Note that for $x \in \{0,1\}^n$, $x \neq x_0$, we have

$x \in A \rightarrow (\exists y, |y| = q(n))[V(x, y, r_0) = 1]$

$x \notin A \rightarrow (\forall y, |y| = q(n))[V(x, y, r_0) = 0]$

Let $\text{ADV}(0^n)$ be the r_0 that you obtain. Note that (aside from x_0)

$A = \{x : (\exists y, |y| = q(|x|)[V(x, y, \text{ADV}(0^{|x|}) = 1]\}$.

Hence $A \in$ NP/poly if you ignore the status of x_0. By Lemma 6.2 we have $A \in$ NP/poly. ∎

7 If GI is NPC then PH collapses

We recall two theorems from other sets of notes. From the notes on Sparse Sets we recall the following.

Theorem 7.1 *If $TAUT \in$ NP/poly then $\Sigma_3^{\mathrm{p}} = \Pi_3^{\mathrm{p}}$.*

From the notes on P, NP, and PH we recall the following

Theorem 7.2 *If $\Sigma_i^{\mathrm{p}} = \Pi_i^{\mathrm{p}}$ then $PH = \Sigma_i^{\mathrm{p}} = \Pi_i^{\mathrm{p}}$.*

We will need the following which is an easy exercise.

Theorem 7.3 *If $A \in$ AM and $B \leq_{\mathrm{m}}^{\mathrm{p}} A$ then $B \in$ AM.*

We now prove the main theorem.

Theorem 7.4 *If GI is NPC then $\Sigma_3^{\mathrm{p}} = \Pi_3^{\mathrm{p}}$.*

Proof: If GI is NPC then $SAT \leq_{\mathrm{m}}^{\mathrm{p}} GI$, so

$$TAUT \leq_{\mathrm{m}}^{\mathrm{p}} \overline{GI}.$$

By Theorem 5.1 $\overline{GI} \in$ AM. By Theorem 7.3

$$TAUT \in \text{AM}.$$

By Theorem 6.3 AM $\subseteq$ NP/poly. Hence

$$TAUT \in \text{NP/poly}.$$

By Theorem 7.1

$$\text{PH} = \Sigma_3^{\mathrm{p}} = \Pi_3^{\mathrm{p}}.$$

∎

References

[1] L. Babai. Trading group theory for randomness. In *Proceedings of the Seventeenth Annual ACM Symposium on the Theory of Computing,* Providence RI, 1985.

[2] L. Babai and S. Moran. Arthur-Merlin games: a randomized proof system and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, pages 254–276, 1988. Prior version in STOC85.

[3] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, pages 186–208, 1989.

[4] S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. In S. Micali, editor, *Randomness and Computation*, pages 73–90, Greenwich, CT, 1989. JAI Press. Earlier version in STOC86.

[5] J. Köbler, U. Schöning, and J. Torán. *The Graph Isomorphism Problem: Its Structural Complexity.* Progress in Theoretical Computer Science. Birkhauser, Boston, 1993.

[6] U. Schnoing. Graph isomorphism is in the low hierarchy. *Journal of Computer and System Sciences*, 59:312–323, 1988.