CS 6810: Theory of Computing **Spring 2026**

Lecture 19: April 7, 2026

Lecturer: Eshan Chattopadhyay *Scribe: Matthew Cowan*

Review from Last Time

Interactive Proofs

An interactive proof system has two parties:

- a computationally unbounded prover P,
- a probabilistic polynomial-time verifier V.

Unlike an ordinary proof system, the prover and verifier are allowed to interact.

For a language L:

- *Completeness:* if $x \in L$, then there exists a prover strategy causing V to accept with high probability,
- *Soundness:* if $x \notin L$, then every prover strategy causes V to accept only with small probability.

It's important to contrast this with the usual certificate view:

$$P \xrightarrow{\text{certificate } c} V.$$

Just having the prover send the certificate over is what is intuitively captured by NP, but by allowing interaction we aim to capture NP and more. Also note that in the above definition, we say large probability or small probability because these probabilities can be driven down to be negligibly small simply by running the interaction multiple times, as seen in previous classes.

Arthur–Merlin Protocols(and Merlin-Arthur)

In an Arthur–Merlin protocol, the verifier's randomness is *public*. Arthur sends random coins, Merlin responds, and acceptance is computed by a polynomial-time verifier.

For a two-message protocol, we can think of Arthur and Merlin playing a game where

- Arthur sends randomly sampled $r \sim \{0,1\}^n$ to Merlin
- Merlin creates some kind of message m, and sends it to Arthur
- The output of the interaction is $V(x, r, m)$ where V is the verifier Turing Machine

Once again, we have the ideas of completeness and soundness captured by:

$$x \in L \implies \exists m \text{ such that } \Pr_r\big[V(x,r,m) = 1\big] = 1,$$
$$x \notin L \implies \forall m \text{ we have } \Pr_r\big[V(x,r,m) = 1\big] \leq \tfrac{1}{3}.$$

Note that we are giving without proof that we can have perfect completeness for interactive proofs.

We could also consider a Merlin-Arthur protocol which is the same game, except Merlin sends m first, then Arthur sends r. In this game, Merlin must send his message without any knowledge of the randomness sampled by Arthur. We will not give proof, but this difference means that under the same derandomization assumptions used to prove Nissan-Wigderson, we can conclude that $\mathsf{MA} = \mathsf{NP}$.

Graph Isomorphism and Graph Non-Isomorphism

Definitions

Definition 1. *Two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$ are* isomorphic*, written $G \cong H$, if there exists a bijection*

$$f : V_G \to V_H$$

such that

$$(u, v) \in E_G \iff (f(u), f(v)) \in E_H.$$

Equivalently, G and H are the same graph up to relabeling of vertices.

We define the languages

$$\mathsf{GI} = \{(G, H) : G \cong H\},$$
$$\mathsf{GNI} = \{(G, H) : G \ncong H\}.$$

Graph Isomorphism is an important problem in NP because it is one that is widely believed to be an intermediate problem – one that is not in P but is not NP complete either. Even recent improvements in solving Graph Isomorphism like Babai's result from 2016, only achieve $n^{\mathrm{poly}(\log n)}$ running time, which is quasipolynomial.

The Goldwasser–Sipser Theorem

Theorem 1. $\mathsf{GNI} \in \mathsf{AM}[t]$ *for some $t \geq 2$. Equivalently, the Graph Non-Isomorphism problem can be solved by an Arthur Merlin Protocol that goes for t rounds.*

One reason we find this theorem so interesting is because we know that $\mathsf{GI} \in \mathsf{NP}$, but we don't believe that NP is closed under complement. Nevertheless, we will show that $\mathsf{GNI} \in \mathsf{AM}$, which is surprising since we can morally think of AM as a randomized version of NP.

Set Lower Bound Sketch

An important technique used in the Goldwasser-Sipser proof is a set lower bound protocol.

Set Lower Bound Protocol

We are given a succinctly represented set $S \subseteq \{0,1\}^n$ where membership in S can be easily certified. The goal is to distinguish the case where S is *large* from the case where S is *small*. Formally, there is some parameter k known to Arthur and Merlin and if $|S| > k$, then Merlin should convince

Arthur with high probability that this is the case, and if $|S| < \frac{k}{2}$, ie the set is much smaller than k, then Arthur should reject with high probability. This can be thought of as a promise problem because there are no guarantees for when $\frac{k}{2} \leq |S| \leq k$.

In the context of solving the Graph Non-Isomorphism problem, we will use

$$S := \{H : H \text{ is isomorphic to at least one of } G_1, G_2\}$$

Note that S surely is a subset of $\{0,1\}^n$ (or maybe really n^2) by encoding graphs in their adjacency matrices. Also note that membership is easily checked by giving the relabelings as the certificate.

The high-level idea of the protocol is to use hashing:

- choose a pairwise independent family of hash functions,
- hash elements of S into a smaller range,
- ask Merlin to produce an element of S that lands in a designated bucket,
- use the acceptance probability to estimate whether $|S|$ is above or below the target threshold.

Hashing Setup

Let $\mathcal{H}_i$ be a pairwise independent(or equivalently 2-universal) family of functions

$$h : \{0,1\}^n \to \{0,1\}^\ell.$$

Then for distinct x, x',

$$\Pr_h[h(x) = h(x')] \leq 2^{-\ell}.$$

For a fixed set $S \subseteq \{0,1\}^n$ and $y \in \{0,1\}^\ell$, define the random variable

$$N(S) = \big|\{x \in S : h(x) = y\}\big|.$$

By linearity of expectation,

$$\mathbb{E}_h[N(S)] = \frac{|S|}{2^\ell}.$$

Arthur–Merlin Protocol for Set Lower Bound

Now that we've defined some important properties of 2-universal Hash Families, let's walk through the protocol step by step.

1. Arthur samples a 2-universal hash function $h \sim \mathcal{H}_i$ and a codomain element $y \sim \{0,1\}^\ell$ and sends h, y to Merlin.
2. Merlin returns an element $x \in \{0,1\}^n$ and a circuit $C \in \{0,1\}^{\text{poly}(n)}$, where C is the circuit checking membership in S.
3. Arthur verifies both:
   - $x \in S$ (by using C),
   - $h(x) = y$.

4. Arthur accepts iff both checks pass.

Now, we must analyze the probabilities for accepting in this protocol, and then later connect this to the size of S and the Graph Isomorphism problem. Assume that the codomain of the hash function $\{0,1\}^\ell$ is such that $2^{\ell-2} \leq k \leq 2^{\ell-1}$.

The probability that Arthur accepts for an optimal Merlin

$$\begin{aligned} &= \Pr_{h,y}[\exists x \in S.h(x) = y] \\ &= \frac{1}{2^\ell}\mathbb{E}[|h(S)|] \end{aligned}$$

because the probability of there existing a preimage is the same as the probability of y landing in the image of h. Now, let $\frac{|S|}{2^\ell} := p^*$, and note that $\frac{1}{4} \leq p^* \leq \frac{1}{2}$ by how ℓ is picked above. First, by simply noting that an image can be no larger than its domain, we'll create the nice upper bound

$$\frac{1}{2^\ell}\mathbb{E}_h\left[|h(x)|\right] \leq \frac{|S|}{2^\ell} = p^*$$

Next, we will make a lower bound by using the same logic that the largest image is size $|S|$ and that any image that is smaller than S is due to hash collisions. Formally,

$$\begin{aligned} \frac{1}{2^\ell}\mathbb{E}_h\left[|h(s)|\right] &\geq \frac{|S|}{2^\ell} - \frac{1}{2^\ell}\sum_{x \neq x'} \Pr[h(x) = h(x')] \\ &\geq \frac{|S|}{2^\ell} - \frac{1}{L} \cdot \frac{1}{L}\binom{|S|}{2} \end{aligned}$$

We can conclude that $\frac{|S|}{2^\ell} - \frac{|S|^2}{2^{2\ell+1}} \leq \Pr[\exists x \in S.h(x) = y] \leq \frac{|S|}{2^\ell}$, or equivalently

$$p^* - \frac{1}{2}(p^*)^2 \leq \Pr[\exists x \in S.h(x) = y] \leq p^*$$

In the case where $|S| \geq k$, it follows that the probability of success is therefore

$$\begin{aligned} &\geq \frac{k}{2} - \frac{k^2}{2 \cdot 2^{2\ell}} \\ &= p^*(1 - \frac{1}{2}p^*) \\ &\geq \frac{3}{4}p^* \end{aligned}$$

In the case where $|S| \leq \frac{k}{2}$, then it's also at most $p^*/2$ by definition. Therefore, if we run this for t rounds, and accept iff Arthur accepts at least $0.6p^*t$ times, then the Chernoff bound will give us the necessary soundness and completeness errors.

Connection Back to Graph Non-Isomorphism

For graph non-isomorphism, the relevant set is the collection of graphs isomorphic to one of the given graphs. It should be clear that hashing to estimate the size of the set S solves the GNI by looking at the simple case where each graph G_1, G_2 has $n!$ isomorphic graphs(we could change the

definition of S slightly to accomodate the case that G_i has fewer than $n!$ isomorphisms). If each $G_1 \ncong G_2$, then each has distinct isomorphisms so $|S| = 2 \cdot n!$, but if they are equal then all of the re-labelings are the same so $|S| = n!$. Clearly, the set lower bound protocol can be used to differentiate the first case from the second.

Summary.

- Interactive proofs generalize certificate verification by allowing interaction.
- Arthur–Merlin protocols are public-coin interactive proofs.
- Hashing and set lower bound arguments explain how public-coin protocols can certify largeness of suitably defined sets.