**CS 6810: Theory of Computing** **Fall 2023**

Lecture 24: Nov 14, 2021

*Lecturer: Eshan Chattopadhyay* *Scribe: Daniel Brous*

# 1 Relation between AM and MA

**Theorem 1.1.** MA $\subseteq$ AM.

*Proof.* Take some language $L \in$ MA. Due to the perfect completeness of Merlin Arthur protocols as we proved in the last lecture, this means that there's some Merlin Arthur protocol consisting of a verifier $V$ and polynomials $f, g \in \mathbb{Z}[|x|]$ such that

1. If $x \in L$, then there exists some $m \in \{0,1\}^{g(|x|)}$ such that for all $r \in \{0,1\}^{f(|x|)}$, $V(x,m,r) = 1$.
2. If $x \notin L$, then for all $m \in \{0,1\}^{g(|x|)}$,

$$\Pr[V(x,m,r) = 1] < \frac{1}{3}.$$

We can first reduce the error bound of our verifier from $\frac{1}{3}$ to $\frac{1}{2^{g(|x|)+2}}$ in the case where $x \notin L$ without affecting perfect completeness: Let's define a verifier $\hat{V}(x,m,r)$ which produces a random string $r \in \{0,1\}^{h(|x|)}$ which encodes $g(|x|)+2$ random strings $r_i$ of length $f(|x|)$ each, i.e. $r = \langle r_1, \ldots, r_{|m|+2} \rangle$ where $r_i \sim \text{Uniform}[\{0,1\}^{f(|x|)}]$ and $h(|x|)$ is the polynomial size of this encoding, and runs $V(x,m,r_i)$ for each $i$ and takes the and of the outputs, i.e.

$$\hat{V}(x,m,r) = \bigwedge_{i=1}^{g(|x|)+2} V(x,m,r_i).$$

1. If $x \in L$, then we already know that there exists some $m \in \{0,1\}^{g(|x|)}$ such that for all $r \in \{0,1\}^f(|x|)$, $V(x,m,r) = 1$. Therefore, with this same $m$, for all sets of $g(|x|)+2$ strings $r_i \in \{0,1\}^{f(|x|)}$,
$$\hat{V}(x,m,r) = \bigwedge_{i=1}^{g(|x|)+2} V(x,m,r_i) = 1.$$
2. If $x \notin L$, then for any $m \in \{0,1\}^{g(|x|)}$,

$$\Pr[\hat{V}(x,m,r) = 1] = \prod_{i=1}^{g(|x|)+2} \Pr[V(x,m,r_i) = 1] < \prod_{i=1}^{g(|x|)+2} \frac{1}{2} = \frac{1}{2^{g(|x|)+2}}.$$

Now let's show that $L \in$ AM. Let's define a verifier $\hat{V}'$ which does everything exactly as $\hat{V}$, but it sends the random string $r$ to Merlin so that we have an Arthur-Merlin protocol.

1. If $x \in L$, then we already know that there exists some $m \in \{0,1\}^{g(|x|)}$ such that for all $r \in \{0,1\}^{h(|x|)}$, $\hat{V}(x,m,r)$. Thus, if we define a Merlin which just outputs this $m$ regardless of the message $r$ they receive, then we have that $\hat{V}'(x, m(x,r), r) = 1$ for all $r \in \{0,1\}^{f(|x|)}$.

2. If $x \notin L$, then we already know that for all $m \in \{0,1\}^{g(|x|)}$, $\Pr[\hat{V}(x,m,r)=1] < \frac{1}{2^{g(|x|)+2}}$. Therefore, using this, we can see that

$$\begin{aligned}
\Pr[\hat{V}'(x,m(x,r),r)=1] &= \frac{1}{2^{h(|x|)}} \sum_{r \in \{0,1\}^{h(|x|)}} \mathbb{1}[\hat{V}'(x,m(x,r),r)=1] \\
&\leq \frac{1}{2^{h(|x|)}} \sum_{m \in \{0,1\}^{g(|x|)}} \sum_{r \in \{0,1\}^{h(|x|)}} \mathbb{1}[\hat{V}(x,m,r)=1] \\
&= \sum_{m \in \{0,1\}^{g(|x|)}} \Pr[\hat{V}(x,m,r)=1] \\
&< \sum_{m \in \{0,1\}^{g(|x|)}} \frac{1}{2^{g(|x|)+2}} \\
&= \frac{1}{2^2} \\
&< \frac{1}{3},
\end{aligned}$$

where $\mathbb{1}[E]$ is the indicator function of the event $E$, i.e. it's 1 when $E$ is true and 0 otherwise.

□

An immediate corollary of this is the following:

**Corollary 1.2.** $\mathrm{AM}[O(1)] = \mathrm{AM}$.

*Proof.* The proof is that if we have a signal sent from $A$ to $M$, then $M$ to $A$, then $A$ to $M$, i.e. an AMAM protocol, then the middle signal along with the random string that Arthur sends back to Merlin is itself an MA protocol, and so we can apply the theorem we just proved to get an AM protocol by repeating the original MA verifier sufficiently many times, and so our total protocol is an AAMM protocol. But since sending two strings of length $\mathrm{poly}(|x|)$ is the same as sending one string of length $\mathrm{poly}(|x|)$, this is really just an AM protocol. We can repeat this step $O(1)$ times to get that $\mathrm{AM}[O(1)] = \mathrm{AM}$. □

**Observation 1.3.** *We can't say that* $\mathrm{AM}[\mathrm{poly}(n)] = \mathrm{AM}$ *because the size of the messages we have to sum over to get our upper bound for the case with* $x \notin L$ *in the proof of Theorem 1.1 increases with each switch from one of the* MA *subprotocols to an* AM *subprotocol, and this increase happens too quickly.*

## 2 Perfect Completeness of AM

We saw in the last lecture that we could get perfect completeness of MA, i.e. for any language $L \in \mathrm{MA}$, we can construct a verifier $V$ that always get's the right answer for some $m$ (where $f(|x|), g(|x|) \in \mathbb{Z}[|x|]$):

1. If $x \in L$, then there exists an $m \in \{0,1\}^{f(|x|)}$ such that for all $r \in \{0,1\}^{g(|x|)}$, $V(x,m,r) = 1$
2. If $x \notin L$, then for all $m \in \{0,1\}^{f(|x|)}$, $\Pr[V(x,m,r)=1] < \frac{1}{3}$.

It turns out that we can also get perfect completeness of AM. The way we'll show this is by constructing a perfectly complete MAM protocol for a language in AM and then using the theorem we proved earlier to switch the perfectly complete MA subprotocol to a perfectly complete AM subprotocol.

So take some language $L \in \text{AM}$. We haven't proved it, but one can show that there exists some $\varepsilon > 0$ on the order of $\frac{1}{\text{poly}(n)}$ and a verifier $V$ such that if $r \sim \text{Uniform}[\{0,1\}^l]$, then

1. If $x \in L$, then there exists an $m(x,r)$ such that $\Pr[V(x,r,m(x,r)) = 1] \geq 1 - \varepsilon$.

2. If $x \notin L$, then for all $m(x,r)$, $\Pr[V(x,r,m(x,r)) = 1] \leq \varepsilon$.

Equivalently, if we define

$$\mathbf{1}_{x,m(x,r)} := \{r \in \{0,1\}^{f(|x|)} \mid V(x,m(x,r),r) = 1\}$$

and let $L := 2^{f(|x|)}$, then these two conditions can be written as

1. If $x \in L$, then there exists an $m(x,r)$ such that $|\mathbf{1}_{x,m(x,r)}| \geq (1-\varepsilon)L$.

2. If $x \notin L$, then for all $m(x,r)$, $|\mathbf{1}_{x,m(x,r)}| \leq \varepsilon L$.

We know from the previous lecture that there exists $v_1, \ldots, v_t \in \{0,1\}^{f(|x|)}$ with $t = \text{poly}(n)$ such that for all $S \subseteq \{0,1\}^{f(|x|)}$ with $|S| \geq (1-\varepsilon)L$, $\bigcup_{i=1}^{t}(v_i + S) = \{0,1\}^{f(|x|)}$. This gives us the following:

1. If $x \in L$, then there exists an $m(x,r)$ such that $|\mathbf{1}_{x,m(x,r)}| \geq (1-\varepsilon)L$. Therefore, there exists an $m(x,r)$ such that for all $r \in \{0,1\}^{f(|x|)}$, there exists an $i \in [t]$ such that $V(x, m(x, r + v_i), r + v_i) = 1$.

2. If $x \notin L$, then for all $m(x,r)$, $|\mathbf{1}_{x,m(x,r)}| \leq \varepsilon L$. Therefore, for all $m(x,r)$, using the union bound, we have that

$$\begin{aligned}
\Pr[\exists i \in [t] \mid V(x, m(x, r+v_i), r+v_i) = 1] &= \Pr\left[\bigcup_{i \in [t]} \{V(x, m(x, r+v_i), r+v_i) = 1\}\right] \\
&\leq \sum_{i=1}^{t} \Pr[V(x, m(x, r+v_i), r+v_i) = 1] \\
&\leq \varepsilon t.
\end{aligned}$$

This means that for $\varepsilon$ small enough, we have found a perfectly complete MAM protocol for $L$ where $M$ sends $v_1, \ldots, v_t$ to $A$, $A$ sends $r$, and $M$ sends $m(x,r)$ and $i \in [t]$ back to $A$, and the verifier is

$$\hat{V}(x, v_1, \ldots, v_t, m, i, r) := V(x, m, r + v_i).$$

Thus, using Theorem 1.1 to switch the MA subprotocol to an AM subprotocol, we get a perfectly complete AM protocol for $L$.

## 3 Why $GI$ is ***probably*** not $NP$-complete

We use the following fact, that you will prove as part of homework.
**Fact:** AM $\subseteq \Pi_2$.

Here's a pretty good reason that we believe that $GI$ is probably not $NP$-complete:

**Theorem 3.1.** *If $GI$ is $NP$-complete, then $PH$ collapses to level 2.*

*Proof.* It suffices to show that if $GNI$ is $co-NP$ complete, then $\Sigma_2 - SAT \in$ AM (because then $\Sigma_2 = \Pi_2 = PH$ since $\Sigma_2 - SAT$ is $\Sigma_2$-complete and AM $\subseteq \Pi_2$). Therefore, suppose that $GNI$ is $co-NP$ complete. We know that $GNI \in$ AM from the previous lecture. Thus, $co-NP \subseteq$ AM. By definition,

$$\Sigma_2 - SAT = \{\phi \mid \exists x \text{s.t.} \forall y, \phi(x,y) = 1\}.$$

Since $co - NP = \forall P$,

$$S_x := \{\phi \mid \forall y, \phi(x,y) = 1\} \in co - NP$$

for all $x$, and so since $co-NP \subseteq$ AM, there exists an AM protocol for $S_x$. If we first have Merlin send in an $x$ and then run this AM protocol for $S_x$, then it's an MAM protocol for $\Sigma_2 - SAT$. Finally, using Theorem 1.1, we can switch the MA subprotocol to an AM subprotocol, meaning we now have an AM protocol for $\Sigma_2 - SAT$, i.e. $\Sigma_2 - SAT \in$ AM. □

## 4 $co - NP \subseteq IP$

We want to show that $\overline{SAT} \in IP$. Instead, we'll show something stronger: we'll create an $IP$ protocol for

$$S := \{(\phi, r) \mid r \text{ is the number of satisfying assignments of } \phi\}.$$

Notice that $(\phi, 0) \in S$ if and only if $\phi \in \overline{SAT}$–this is why this statement is stronger.

Consider the following IP protocol: Given as input to the system some $(\phi, r)$, the prover $P$ sends $r_0, r_1$ to the verifier $V$, the verifier checks if $r = r_0 + r_1$, and sends back a random bit $b$. We then reduce the formula $\phi$ to $\phi_b$, i.e. setting the first variable in $\phi$ to have value $b$, and $r$ to $r_b$. We then recurse the described interaction until the formula has all variables assigned to some fixed choices, at which point the verifier returns 1 if all the checks were valid. This is $2n$ rounds total (where $\phi$ has $n$ variables). If $(\phi, r) \in S$, then there must exists some $\{r_b\}_{b \in \bigcup \{0,1\}^l, l \in [n]}$ because we can choose $r_b$ to be the number of satisfying assignments for $\phi_b$ (this works because then $r_{b0} + r_{b1} = r_b$ for all $b$). If $(\phi, r) \notin S$ however, then this protocol doesn't quite work because

$$\Pr[V = 1] \geq \frac{1}{2^n}$$

Instead we'll use polynomials. Fix some prime $q \in [2^n, 2^{2n})$ (which we can compute with randomness). Given a boolean 3-CNF formula $\phi(x_1, \ldots, x_n)$, we can make a polynomial $f_\phi(x_1, \ldots, x_n) \in \mathbb{F}_q[x]$ of degree $3m$ that evaluates to 1 on some assignment of $x_1, \ldots, x_n \in \{0, 1\}$ if and only if $\phi$ evaluates to 1 on this assignment. We do this by making a degree 3 poly for each clause and multiplying them. Literal $x_i$ gets mapped to the poly $1 - x_i$, literal $\overline{x_i}$ gets mapped to $x_i$, and the clause gets mapped to 1 minus the product of the literal polys. For example,

$$x_1 \vee x_2 \vee \overline{x_5} \mapsto 1 - (1 - x_1)(1 - x_2)x_5.$$

Now, since our polynomial has this property, it holds that

$$\sum_{(x_1,\ldots,x_n)\in\{0,1\}^n} f_\phi(x_1,\ldots,x_n) = r$$

over the real numbers if and only if $(\phi, r) \in S$. Since our prime $q > 2^n$ and $r \le 2^n$, the above equation holds over the real numbers if and only if it holds over $\mathbb{F}_q$. Therefore, it suffices to create an IP protocol for SUMCHECK, i.e: Given a degree $d$ polynomial $g(x_1,\ldots,x_n)$, a prime $p$, and an integer $z$, verify whether or not

$$z = \sum_{(x_1,\ldots,x_n)\in\{0,1\}^n} g(x_1,\ldots,x_n).$$

Here is the protocol we'll use:

1. If $n = 1$, then accept if and only if $g(0) + g(1) = z$.
2. If $n \ge 2$, prover sends univariate polynomial $h(x_1) \in \mathbb{F}_p[x_1]$.
3. If $h(0) + h(1) \ne z$, then reject. Otherwise, pick a random bit $b$ and recursively check whether

$$b = \sum_{(x_2,\ldots,x_n)\in\{0,1\}^{n-1}} g(b, x_2,\ldots,x_n).$$

If $z = \sum_{(x_1,\ldots,x_n)\in\{0,1\}^n} g(x_1,\ldots,x_n)$, then the prover can just send

$$h(x_1) = \sum_{(x_2,\ldots,x_n)\in\{0,1\}^{n-1}} g(x_1, x_2,\ldots,x_n)$$

and the verifier will accept. If $z \ne \sum_{(x_1,\ldots,x_n)\in\{0,1\}^n} g(x_1,\ldots,x_n)$, then we claim that

$$\Pr[V = 0] \ge \left(1 - \frac{d}{p}\right)^n.$$

We can prove this by induction. It's true for $n = 1$ since the verifier will reject with probability 1. Suppose it's true for $n - 1$ for some $n \in \mathbb{N}$. If the prover were to send the correct $h(x_1) = \sum_{(x_2,\ldots,x_n)\in\{0,1\}^{n-1}} g(x_1, x_2,\ldots,x_n)$ in the first round, then since $h(0) + h(1) \ne z$, the verifier would reject with probability 1, so without loss we can assume that's not the polynomial they send. Therefore, since $h(x_1) - \sum_{(x_2,\ldots,x_n)\in\{0,1\}^{n-1}} g(x_1, x_2,\ldots,x_n) \ne 0$ and is a degree $d$ polynomial, it has at most $d$ roots, and so there are at most $d$ values of $x_1$ for which $h(x_1) = \sum_{(x_2,\ldots,x_n)\in\{0,1\}^{n-1}} g(x_1, x_2,\ldots,x_n)$. Thus,

$$\Pr_b[h(b) \ne \sum_{(x_2,\ldots,x_n)\in\{0,1\}^{n-1}} g(b, x_2,\ldots,x_n)] \ge 1 - \frac{d}{p}$$

The probability that $V$ rejects the initial claim is at most the probability that

$$h(b) \ne \sum_{(x_2,\ldots,x_n)\in\{0,1\}^{n-1}} g(b, x_2,\ldots,x_n)$$

and $V$ rejects the recursive claim, i.e. whether $h(b) = \sum_{(x_2,\ldots,x_n)\in\{0,1\}^{n-1}} g(b, x_2, \ldots, x_n)$. Since the recursive claim is false, by the inductive hypothesis, $V$ rejects this with probability at most $\left(1 - \frac{d}{p}\right)^{n-1}$, and so $V$ rejects the initial claim with probability at most

$$\left(1 - \frac{d}{p}\right)\left(1 - \frac{d}{p}\right)^{n-1} = \left(1 - \frac{d}{p}\right)^{n}.$$

Finally, now that we have our claim, we can use the fact that $d = 3m$ and $p = q$ in our special case of SUMCHECK with boolean formula polynomials, and so our IP protocol accepts with $(\phi, r) \notin S$ with probability at most

$$\left(1 - \frac{3m}{q}\right)^{n}.$$

Sine $q > 2^n$, we get a desired bound on the case when $(\phi, r) \notin S$ because $m$ is polynomial in $n$.