Lecture 17

In this lecture we will look at the notion of *Interactive Proofs* and the induced class **IP**. Intuitively, this class is obtained by adding the ingredients Interaction and Randomness to the class **NP**. As we shall see, adding either one does not make much difference but the two put together make **IP** a very powerful class.

1 Interactive Proofs

We begin be recalling the definition of **NP**. Given a language L, we say that: $L \in \mathbf{NP}$ iff there is an algorithm $V(\cdot,\cdot)$ running in polynomial time $T(\cdot)$ and a polynomial $p(\cdot)$ such that

$$x \in L \Leftrightarrow \exists y, |y| \leq p(|x|) \text{ and } V(x,y) \text{ accepts}$$

We can rewrite the above as two separate conditions, using elementary ideas from logic.

$$x \in L \Rightarrow \exists y, |y| \leq p(|x|) \text{ and } V(x,y) \text{ accepts} \qquad \text{(Completeness)}$$
$$x \notin L \Rightarrow \forall y, |y| \leq p(|x|) \text{ and } V(x,y) \text{ rejects} \qquad \text{(Soundness)}$$

We may generalise this to the figure below 1 where the process of recognizing if $x \in L$ is carried out by a *Verifier* V that must run in polynomial time, and a *Prover* P, who has unbounded time and space. The task of the prover is to convince the verifier that $x \in L$ and the verifier is a skeptic who's task is to be convinced if and only if the string actually belongs in the language.

Now the class **NP** is the set of languages L such that $L \in \mathbf{NP}$ iff there is a Prover P and a polynomial time verifier V such that:

$$x \in L \Rightarrow P \text{ has a strategy to convince V} \qquad \text{(Completeness)}$$
$$x \notin L \Rightarrow P \text{ has no strategy to convince } V \qquad \text{(Soundness)}$$

By *strategy* we mean in this case the certificate or proof that is polynomially small, (corresponding to y in the figure 1) that P supplies to V, while later we will generalise it to interaction *i.e.* where a sequence of messages is exchanged between the two and a strategy for P is a function from the sequence of messages seen to the next message that P must send to V.

Thus, we generalise the above definition to get the class **IP**. The class **IP** differs in the following two ways:

1. There is randomness, the verifier may be a randomized machine.
2. There is interaction: unlike the above where a single static proof is sent across, there are rounds of communication, where the verifier may "ask" further questions of the prover based on the messages that have been sent to it.

First, we shall see that both of the above are required.

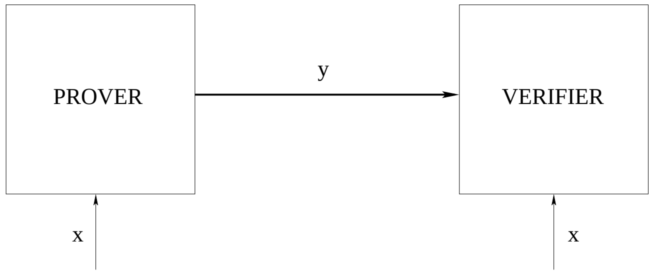

Figure 1: **NP** with a Prover-Verifier mechanism

1.1 NP + Interaction = NP

Let the class $\overline{\mathbf{NP}}$ be the class corresponding to $\mathbf{NP}$ + interaction. That is we allow the verifier to ask questions and thus the strategy is a function from sequences of messages (the past communication) to messages (the next message that the verifier must send). We say that the class $\overline{\mathbf{NP}}$ is the set of languages L such that $L \in \overline{\mathbf{NP}}$ iff there is a Prover P and a polynomial time verifier V such that:

$$x \in L \Rightarrow P \text{ has a strategy to convince } V$$
$$x \notin L \Rightarrow P \text{ has no strategy to convince } V$$

It is easy to check that $\mathbf{NP} = \overline{\mathbf{NP}}$. We only need to show $\overline{\mathbf{NP}} \subseteq \mathbf{NP}$, as the other inclusion is obvious. Say $L \in \overline{\mathbf{NP}}$, let P, V be the prover and verifier for L. Define P', V' the one-round prover and verifier as follows: P' sends to V' the entire transcript of the interaction between P, V as the certificate. Suppose that interaction was the set of messages $y_1, \ldots, y_n$ as in the figure 2 below. V' is essentially like V only it has been given all the answers to its questions at once. Thus it takes the first message of P, y_1, its response (behaving as V) is exactly y_2, but y_3 is the reply to that, and so on, it "asks" the questions and is supplied all the answers on the same transcript, and it accepts iff after this exchange V accepted. Thus the language accepted by P', V' is the same as L and so $L \in \mathbf{NP}$. Note that the entire set of messages is polynomially long as there are polynomially many messages and each is polynomially long.

Note that for the above to work, the prover must at the very beginning know what responses V may make, which is possible only if V is deterministic. If V was randomized then to cover all possible questions in a polynomially long interaction P' would have to send an exponentially large transcript.

1.2 NP + Randomness

By adding randomness to the class **NP** we get the class **MA**.

Definition 1 (**MA**) $L \in \mathbf{MA}$ *iff there exists a probabilistic polynomial time machine* V

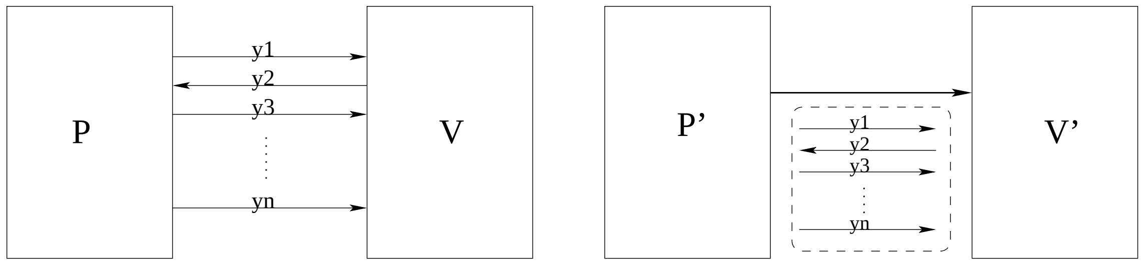

Figure 2: Simulating several rounds of interaction in one round

such that:

$$x \in L \Rightarrow \exists y.\mathbf{Pr}[V(x,y) \ \textit{accepts}\ ] \geq \frac{2}{3}$$
$$x \in L \Rightarrow \forall y.\mathbf{Pr}[V(x,y) \ \textit{accepts}\ ] \leq \frac{1}{3}$$

As in similar classes, the success probabilities may be boosted arbitrarily high. It is conjectured that **MA** = **NP**. What is known is that if **coNP** $\subseteq$ **MA** then the polynomial heirarchy collapses.

Exercise 1 A non-deterministic circuit C has two inputs $x = x_1, \ldots, l_m$ and $y = y_1, \ldots, y_n$. C accepts the string x iff $\exists y.C(x,y) = 1$. Show that **MA** has non-deterministic circuits of polynomial size. (*Hint:* Proof is similar to that of **BPP** has poly sized circuits)

We are now in a position to define formally the class **IP**.

2 IP

Definition 2 (**IP**) *A language L is in* $\mathbf{IP}(r(\cdot))$ *iff there exists a probabilistic polynomial time verifier V such that:*

$$x \in L \Rightarrow \exists P.\mathbf{Pr}[V \ \textit{interacting with}\ P \ \textit{accepts}\ ] \geq \frac{2}{3}$$
$$x \in L \Rightarrow \forall P.\mathbf{Pr}[V \ \textit{interacting with}\ P \ \textit{accepts}\ ] \leq \frac{1}{3}$$

Where V also uses at most $r(|x|)$ rounds of interaction.

Remark 1 By round we really mean a single "message" and not a question-answer sequence. That is, by two rounds we mean that the Verifier asks a question of the Prover and the Prover replies, which is exactly two messages.

A related class is **AM** which is defined as follows:

Definition 3 (**AM**) *A language L is in* $\mathbf{AM}(r(\cdot))$ *iff* $L \in \mathbf{IP}(r(\cdot))$ *and at each round the verifier sends a random message, that is a message that is completely random and independent of all previous communication.*

The above is also known as *Public Coin Proof Systems.*

There are the following surprising theorems about the classes **AM** and **IP**.

THEOREM 1
$\mathbf{IP}(r(n)) \subseteq \mathbf{AM}(r(n)+2)$

THEOREM 2
$\forall r(n) \geq 2$, $\mathbf{AM}(2r(n)) \subseteq \mathbf{AM}(r(n))$.

This yields the following corollaries:

COROLLARY 3
$\mathbf{AM}(O(1)) \subseteq \mathbf{AM}(2)$

That is, all constant round **AM** proofs can be done in just two rounds ! And also:

COROLLARY 4
IP *(O(1))*=**AM** *(O(1))*=**AM** *(2)*

Finally we have the famous theorem of Shamir:

THEOREM 5
$\mathbf{IP}(poly(n)) = \mathbf{PSPACE}$

We know that $\mathbf{AM}(2)$ is good enough for systems with a constant number of rounds, but it would be surprising indeed if polynomially many rounds could be simulated in a constant number of round. Due to the above result, when people say **AM** they mean $\mathbf{AM}(2)$, but when people say **IP** they mean $\mathbf{IP}(\text{poly})$.

The above classes **IP** and **AM** were discovered independently. The first was discovered by Shafi Goldwasser, Silvio Micali and Charles Rackoff. They proposed the class **IP** in order to facilitate the definition of the class Zero Knowledge Proofs which we shall see shortly, the paper was written in 1983 but appeared first in FOCS 85. The class **AM** was discovered by Lazlo Babai (the journal version of the paper is by Babai and Moran), in order to characterize the complexity of a group-theoretic problem. The paper also appeared in FOCS 85, and contained theorem 2.

There is also the following theorem that gives an upper bound on **IP** with constant rounds.

THEOREM 6
If $\mathbf{coNP} \subseteq \mathbf{IP}(O(1))$ *then the polynomial heirarchy collapses.*

Thus for a long time it was believed that **IP** could not capture **coNP** as it was thought that **IP** with a constant number of rounds was roughly as good as **IP** with polynomially many rounds. However, it was shown that in fact **coNP** was contained in $\mathbf{IP}(\text{poly}(n))$, the proof of which we shall shortly see and a few weeks after that result, Shamir proved that in fact **IP** was the same as **PSPACE**.

3 An Example: Graph Non-Isomorphism

To get a feel of the power of interaction and randomness let us look at an example, GRAPH NON-ISOMORPHISM. The problem is the following:

- Input: Graphs $G_1 = (V_1, E_1),\ G_2 = (V_2, E_2)$
- Output: Decide whether there exists *NO* $\pi : V_1 \mapsto V_2$ such that $(u, v) \in E_1$ iff $(\pi(u), \pi(v)) \in E_2$.

This problem is not known to be in **NP**. The difficulty lies in constructing a succint proof that no permutation of the vertices is good, given that there are exponentially many permutations, let us see how this is in **AM** (2).

Suppose that the skeptic Arthur (the verifier) has the two graphs G_1, G_2 and wishes to know whether or not all the permutations are bad. His only resources are randomness and his wizard Merlin who is all knowing (the prover).

He does the following:

1. Picks a random $b \in \{1, 2\}$.
2. Creates G to be a random permutation of G_b.
3. Sends G to Merlin and demands to know of Merlin which graph he sent him a permutation of.
4. If Merlin was correct Arthur believes that the graphs are non-isomorphic (accepts), otherwise Merlin was incorrect, Arthur believes the graphs are isomorphic (rejects).

Notice that if the graphs are really non-isomorphic, then Merlin will always be able to tell which graph was sent to him as only one of the graphs will be isomorphic to the random permutation that Merlin is given, and with his infinite resources Merlin will figure out which one and thus will reply correctly and so Arthur will accept with probability 1.

If the graphs are isomorphic then Merlin cannot tell as the permutation is a permutation of both $G_1,\ G_2$. Merlin's output depends only on G_1, G_2, G, say it is i. Now $i = b$ with probability $\frac{1}{2}$ as b is picked independantly at the start. Thus in the case the graphs are isomorphic, Arthur accepts only with probability $\frac{1}{2}$. By repeating this k times we can shrink this to 2^{-k}.

Thus we have a protocol, a prover and a verifier for GRAPH NON-ISOMORPHISM and so it is in **AM** (2).

Note: The verifier of the above has learnt nothing at all about the Graphs that he did not know before. He does not have a proof or certificate with which he may convince a third party that the Graphs are non-isomorphic. The entire power was in the interaction between it and the prover, and the one bit that the prover gave it gives it significant confidence but no *knowledge*. This was therefore a *Zero Knowledge* Proof.

REMARK 2 GRAPH NON-ISOMORPHISM $\in$ **IP**$(2) =$ **AM**(2). Thus, GRAPH ISOMORPHISM is not **NP**-complete unless the polynomial heirarchy collapses. Thus is because, if GRAPH ISOMORPHISM is **NP**-complete then GRAPH NON-ISOMORPHISM is **coNP**-complete, and from the above result, **coNP** $\subseteq$ **AM**(2) and from the theorem 6, the polynomial heirarchy would collapse.

In the next couple of lectures we will see how interactive proofs can be given for problems in **coNP**.

THEOREM 7
coNP $\subseteq$ **IP**$(poly(n))$

We shall see that the problem of checking if a 3-CNF formula is not satisfiable, which is a **coNP**-complete problem, is in **IP** and thus conclude that **coNP** is contained in **IP**. This proof may be extended to a proof that the problem of deciding the language of QUANTIFIED BOOLEAN FORMULAE, which is **PSPACE**-complete is in **IP**, and thus shows that **PSPACE** = **IP**.