Instructor: Andrej Bogdanov **Notes by:** Lin Yang

In this lecture we discuss *interactive proofs* (IP). We introduce interaction into the class NP, and find it of great power if combined with randomness, although adding either one does not make much difference. To see how interaction can be added, let's first recall the definition of NP.

A language L is said to be in NP if, there exists NTM V s.t.

$$x \in L \to \exists y, |y| \leq p(|x|), \text{s.t. } V(x, y) = 1$$
$$x \notin L \to \forall y, |y| \leq p(|x|), \text{s.t. } V(x, y) = 0$$

Usually we call y to be a witness or a certificate, as it *proofs* x's identity to V. We can think of a *prover* who provides y for given x to convince V the identity of x. In terms of the new roll, a language L is in NP if, there exists a polynomial time verifier V and a prover P which is computationally unbounded, so that

$$x \in L \to V(x, P(x)) = 1$$
$$x \notin L \to \forall P^*, V(x, P^*(x)) = 0$$

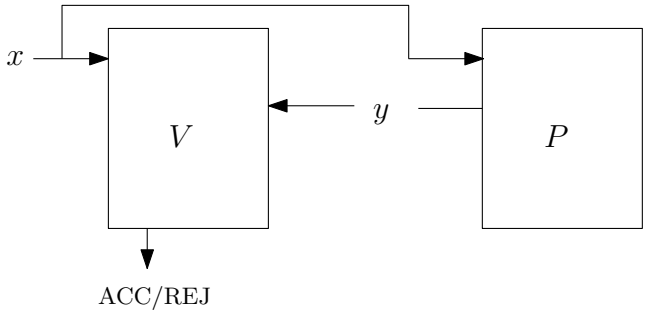

Figure 1: NP with prover

Now we introduce *interaction*. The verifier can ask the prover whatever questions, and the prover gives answers accordingly. Then, the certificate y is split into a series of questions and answers, as illustrated in figure 2.

1 Interaction with Deterministic Verifier

So does interaction help given that $V \in$ DTIME? Sadly the answer is no.

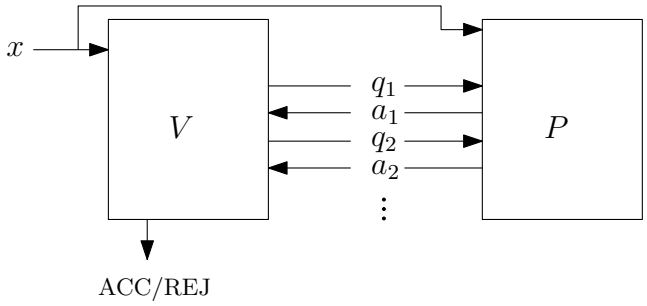

Figure 2: NP with interaction

Claim 1. *Even if we allow interaction, this model is equivalent with* NP*.*

Proof. If $V \in$ DTIME, as P is computationally unbounded, P can just simulate V, predicting every question V would ask, and give all the answers in a batch back to V, denoted by $\tau = (a_1, a_2, \ldots, a_w)$. See figure 3 for illustration. □

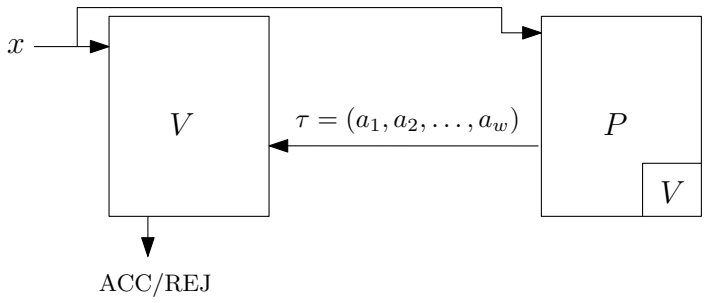

Figure 3: Interaction with DTIME verifier

2 Interaction Combined with Randomness

Now let's suppose V is not determined but randomized. Then the previous argument fails, as the prover can no longer simulate the verifier exactly. To our surprise, the model can do more things now, that is, there exists V and P such that

- P can convince V if V has access to randomness
- We don't know whether V can be derandomized

For example, consider the Graph Non-Isomorphism problem:

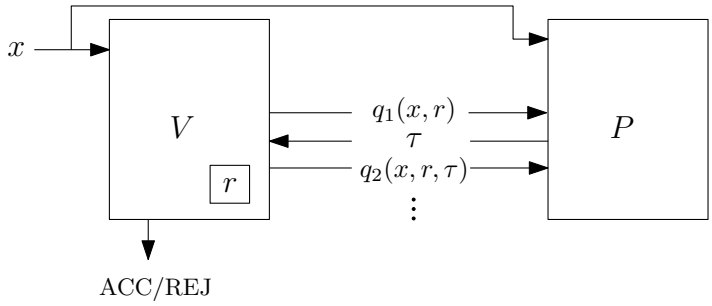

Figure 4: Interaction combined with randomness

GRAPH NON-ISOMORPHISM
Instance Graph G_1 and G_2
Question Are G_1 and G_2 not isomorphic?

Graph G_1 and G_2 are *isomorphic* iff there exists some permutation π of $V(G_1)$ such that after renaming the vertices according to the permutation to G_1, the graph is the same with G_2.

This problem is not known to be in NP. However, it can be proved by using interaction and randomness. The intuition is, V shuffles G_1 and G_2, take one of them and show it to P. As P has unbounded computational resource, it can tell G_1 from G_2 if they are indeed different, and if not, P's guess won't be better than $\frac{1}{2}$. Thus P answers if the given graph is G_1 or G_2. They repeat this procedure for several times. If P got it right in all of the rounds, then G_1 and G_2 are not isomorphic. A protocol is given as follows.

Protocol for GNI

On graph G_1 and G_2:

V: Randomly choose a permutation of G_i, send it to P.

P: Answer whether it is G_1 or G_2.

V: Check P's answer. If not correct, reject; otherwise, repeat from the beginning, until V is fully convinced.

So how many rounds are needed? We first define r round IP, and then prove that GNI is in 2 round IP.

Definition 2. *A language L has r round* IP *iff there exists prover P, and randomized polytime verifier V such that*

$$x \in L \to \Pr\left[(P,V)(x) = 1\right] \geq \frac{2}{3}$$
$$x \notin L \to \Pr\left[(P,V)(x) = 1\right] \leq \frac{1}{3}$$

Let $\mathrm{IP}(r)$ *be the class of languages that has r round* IP.

Claim 3. GNI $\in$ IP(2)

Proof. If G_1 and G_2 are not isomorphic, then $\Pr[V \text{ accepts}] = 1$.

If G_1 and G_2 are isomorphic, let b be the correct answer known only to V and b' be the answer given by P. Then

$$\Pr[V \text{ accepts}] = \Pr\left[b' = b\right] = \frac{1}{2}$$

So two rounds suffices. □

3 Questions

We raise the following questions to introduce more profound properties of IP.

1. Do more rounds of interaction allow us to do more things? That is, is IP(n) the proper subset of IP($n+1$)?
2. Is there a "normal form" for interaction? For example, asking random questions?
3. Is IP really more powerful than NP?
   → GNI: "Yes!"
   So what's the relationship between IP and NP?

First, for the second question, we try to use some "normal form" to regulate the behavior of the verifier. In the model we have so far, the randomness is private for the verifier, and he can do whatever transformation to the random string before he sends it to the prover. He may even send the same thing to the prover on different random strings. So by "normal form" we mean the protocol where the randomness is *public*, i.e., the prover knows what random string verifier gets in each round. Call the IP with this assumption to be IP *with public-coins*. Now we ask: does IP with public-coins work as good as the original one? The answer is "yes".

Theorem 4. IP(r) *can be simulated by a public-coin protocol with* $r+2$ *rounds.* □

Second, for the first question, the answer is "no". In fact, in normal forms, $r+2$ rounds of interaction can be simulated by r rounds of interaction for any $r \geq 2$. Let AM(r) be the class of languages that can be decided by a r round public-coin interactive proof. Then

Theorem 5. AM(r) $\subseteq$ AM($r-2$). □

As IP(r) $\subseteq$ AM($r+2$), and all AM(r) is equivalent to AM(2), that means 2 rounds of interaction are sufficient for IP(r) as long as r is a constant. We abbreviate AM(2) as AM, then we have

Corollary 6. AM = IP(r) *for any constant* r.

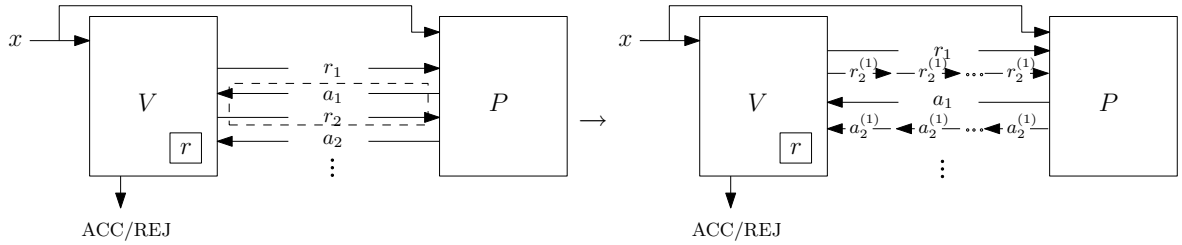

Figure 5: Reversing a pair of question and answer to reduce the number of rounds

Now we sketch a proof for Theorem 5.

To reduce one round in the AM protocol we try to reverse an adjacent pair of question and answer. As questions or answers can be combined, the number of rounds is reduced.

We assume the interaction to be public-coin, where the verifier asks random questions. To switch the order of a_1 and r_2, the verifier tosses the coin for m times and sends the m question generated, namely $r_2^{(1)}, r_2^{(2)}, \ldots, r_2^{(m)}$, to the prover before the prover sends a_1. Then, the prover answers r_1 together with $r_2^{(i)}$ for $i = 1, \ldots, m$. All the interactions that follows would consist of m sub-questions or m sub-answers, and in the end, the verifier looks at all the m copies of interactions and take a census. This protocol would produce the right thing in most of the cases. For the setting of m, we can make it linear in k, where k is the number of bits in each messages in the original protocol.

To see another proof of the round reduction, one can refer to Section F.2.2 in the book by Oded Goldreich, which is linked to on the course homepage.

For the last question, we compare the class AM with NP. AM gives a randomized approximation of NP, just like BPP approximating P. Recall that, under some believable assumption on circuit lower bounds, BPP would be equal to P. So would there also be some believable assumption under which AM would be equal to NP?

Theorem 7. *If there are decision problems decidable in time $2^{O(n)}$ but not decidable by nondeterministic circuits of size $2^{\delta(n)}$ for some $\delta > 0$, then* AM = NP.

A *nondeterministic circuit* is a circuit C such that, on input x, C accepts x iff there exists y s.t. $C(x, y) = 1$. This assumption is the non-uniform generalization of saying that all EXP computations cannot be performed by nondeterministic algorithms running in time $2^{o(n)}$, and is believed to be true.[1]

4 Interactive proofs and the polynomial hierarchy

Sometimes it helps to think of AM as the probabilistic counterpart of NP just like BPP is the probabilistic counterpart of P. In fact, the relations we have seen among P, P/poly, BPP and the polynomial hierarchy translate to the context of interactive proofs:

[1] In fact a somewhat weaker assumption is sufficient.

$\mathrm{BPP} \subseteq \mathrm{P/poly}$	$\mathrm{AM} \subseteq \mathrm{NP/poly}$
$\mathrm{BPP} \subseteq \Sigma_2 \cap \Pi_2$	$\mathrm{AM} \subseteq \Pi_2$
$\mathrm{NP} \subseteq \mathrm{BPP} \;\Rightarrow\; \Sigma_2 = \Pi_2$	$\mathrm{AM} \subseteq \mathrm{coAM} \;\Rightarrow\; \Sigma_2 = \Pi_2$

Here NP/poly is the class of problems decided by nondeterministic families of polynomial-size circuits.

One consequence of the last relation is that graph isomorphism is unlikely to be NP-complete: Since graph isomorphism is in coAM, if graph isomorphism were NP-complete, then we would have $\mathrm{NP} \subseteq \mathrm{coAM}$ and $\Sigma_2 = \Pi_2$.