CS221: Computational Complexity Prof. Salil Vadhan

Lecture 31: Multiprover Interactive Proofs and Probabilistically Checkable Proofs

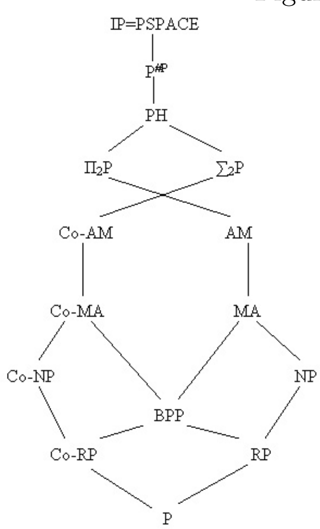
12/09 *Scribe: Qian Zhang*

Contents

1 Recap

Note on hardness of approximate counting: Even though we used allowed randomized algorithms in our definition of α-approximation algorithm, the reductions we give to show hardness of approximate counting are *not* randomized. For example, if there is deterministic polynomial time algorithm to approximate the number of cycles in a graph, we can conclude $\mathbf{P} = \mathbf{NP}$.

Figure 1: Complexity classes involving interactive proofs



AM and**MA** seem much closer to **NP** than **IP** does. There even exists some evidence that $\mathbf{AM} = \mathbf{MA} = \mathbf{NP}$. (Shown in CS225)

2 Multiprover IP

We want to generalize IP to have several cooperating provers. The provers are cooperating since they all want to convince the verifier of the same thing, however, the provers are not allowed to communicate with each other. This is similar to the interrogation of criminals where each suspect is interrogated separately and without allowing them to hear what the others have told the police.

Definition 1 *A k-prover* proof system *for a language L is a protocol between k* provers $(P_1, \dots, P_k)$ *and* verifer V *such that*

1. *(Efficiency) V runs in probabilistic polynomial time, and the number and length of all messages exchanged is at most polynomial in the common input x. (As with* **IP***, the provers $P_1, \dots, P_k$ are computationally unbounded)*

2. *(Noncommunicating provers) $\forall j \neq i$, P_i doesn't see the messages between P_j and V.*

3. *(Completeness) $x \in L \implies \Pr[V$ accepts in $((P_1, P_2, \dots, P_k), V)\,(x)] \geq 2/3$.*

4. *(Soundness) $x \notin L \implies \forall P_1^*, P_2^*, \dots, P_k^*,\ \Pr[V$ accepts in $((P_1^*, P_2^*, \dots, P_k^*), V)\,(x)] \leq 1/3$.*

k-MIP *is the class of languages with k-prover proof systems, and* **MIP** $= \bigcup_k$ **k-MIP**.

More generally, one may also allow the number of provers to grow (polynomially) with the input, but we stick to a constant number of provers for simplicity.

The original motivation for studying **MIP** was cryptography (specifically, one can construct zero-knowledge **MIP**'s for **NP** without any intractability assumptions such as the existence of one-way functions). But our interest in them is complexity-theoretic: Does **MIP** contain more languages than **IP**? Is it true that $(\mathbf{k+1})$-**MIP** $\supsetneq$ **k-MIP**? We answer that question below.

3 Probabilistically Checkable Proofs

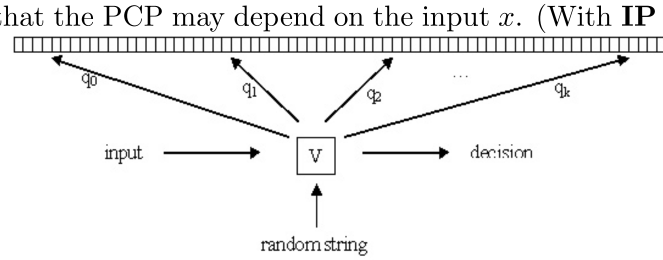
Instead of a prover, we have a proof "oracle". The proof oracle is like a memory-less prover. Thus, while normally provers can remember the history of its interaction with the verifier to avoid contradicting itself, the proof oracle cannot.

Definition 2 *A language L has a* probabilistically checkable proof (PCP) *if there exists a probabilistic polynomial-time* verifier V *such that*

1. *(Completeness) $x \in L \implies \exists \pi : \{0,1\}^* \to \{0,1\}\ :\ \Pr[V^{\pi}(x) = accept] \geq 2/3$.*
2. *(Soundness) $x \notin L \implies \forall \pi^*, \Pr\left[V^{\pi^*}(x) = accept\right] \leq 1/3$.*

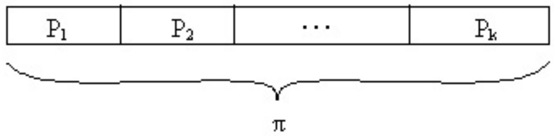
PCP *is the class of languages possessing probabilistically checkable proofs.*

We will sometimes write π_x to emphasize that the PCP may depend on the input x. (With **IP** and **MIP** we gave the input x to the provers.)



Theorem 3 **PCP** = **MIP** = **2-MIP**

Proof that MIP $\subseteq$ PCP: We can view the proof oracle as concatenation of P_i's



where if $m_1, m_2, \ldots, m_i$ are previous messages exchanged with the MIP verifier, then $\pi_x\left(i, (m_1, m_2, \ldots, m_i)\right) = P_i\left(x, (m_1, m_2, \ldots, m_i)\right)$. Thus, given any MIP verifier, we can construct PCP verifier which simulates interactions with $P_1, P_2, \ldots, P_k$ by accessing the proof oracle. ■

Proof that PCP $\subseteq$ 2-MIP:
Suppose we have a PCP verifier V_{PCP}. By amplification, we may assume that the error probabilities are at most 2^{-n}. We can now define a 2-MIP verifier V_{MIP} as thus:

- Simulate $V_{PCP}(x)$ to obtain queries $q_1, q_2, \ldots, q_m$
- Send $(q_1, q_2, \ldots, q_m)$ to P_1 and get the answers $(a_1, a_2, \ldots, a_m)$
- Choose $i \in \{1, 2, \ldots, m\}$ at random, send q_i to P_2, and receive answer b
- Accept if $(a_i = b) \wedge [V_{PCP}$ accepts $(a_1, a_2, \ldots, a_m)]$

Completeness: is trivial since if a PCP π_x exists, we can define $P_1(x, (q_1, \ldots, q_m)) \stackrel{\text{def}}{=} (\pi_x(q_1), \ldots, \pi_x(q_m))$ and $P_2(x, q) = \pi_x(q)$. Thus the acceptance probability of $1 - 2^{-n}$ is maintained.
Soundness: P_2 defines an oracle $\pi_x : \{0,1\}^* \to \{0,1\} \ : \ \pi(q) \triangleq P_2(q)$
By PCP soundness and the amplification, $\Pr\left[V^{\pi}_{PCP}(x) \text{ accepts }\right] \leq 2{-}n$
Equivalently, we can say: $\Pr\left[V_{PCP}(x) \text{ accepts } \left(P_2(x, q_1), P_2(x, q_2), \ldots, P_2(x, q_m)\right)\right] \leq 2^{-n} \leq 1/2$. Then:

$$
\begin{aligned}
& Pr\left[V_{MIP} \text{ rejects in } \left((P_1, P_2), V\right)(x)\right] && (1)\\
& \quad \geq \ \Pr\left(V_{PCP}(x) \text{ rejects } \left(P_2(x, q_1), P_2(x, q_2), \ldots, P_2(x, q_m)\right)\right] && (2)\\
& \qquad \cdot \Pr\left[V_{MIP} \text{ rejects} | V_{PCP} \text{ rejects } (P_2(x, q_1), \ldots, P_2(x, q_m))\right] \\
& \quad \geq \ \frac{1}{2} \cdot \Pr\left[V_{MIP} \text{ rejects} | V_{PCP} \text{ rejects } (P_2(x, q_1), \ldots, P_2(x, q_m))\right] && (3)
\end{aligned}
$$

To bound the last probability, we consider two cases, depending on what P_1 sends.
Case 1: $P_1(x, (q_1, q_2, \ldots, q_m)) = (P_2(x, q_1), P_2(x, q_2), \ldots, P_2(x, q_m))$
In this case, V_{MIP} rejects with probability 1.
Case 2: $P_1(x, (q_1, q_2, \ldots, q_m)) \neq (P_2(x, q_1), P_2(x, q_2), \ldots, P_2(x, q_m))$
In this case, V rejects with probability $\geq 1/m$.
So, expression 1 $\geq$ expression 2 $\geq$ expression 3 $\geq (1/2) \cdot (1/m)$, and we conclude $\Pr[V_{MIP}$ accepts in $\left((P_1, P_2), V\right)(x)] \leq 1 - \frac{1}{2m}$.

Note the gap between the completeness probability $(1 - 2^{-n})$ and the soundness probability $(1 - 1/2m)$. Repeating the PCP verifier $\Theta(m)$ times, we can amplify the gap to any desired constant.

■

While we have this nice equivalence between **MIP** and **PCP**, we still have not answered the question of whether **MIP** is more powerful than **IP** = **1-MIP**. Soon after **IP** = **PSPACE**, the power of **MIP** was completely characterized:

Theorem 4 (Babai–Fortnow–Lund) **MIP** = **PCP** = **NEXP**

We won't prove this theorem, but rather look at a remarkable "scaling down" of it that characterizes **NP**.

Theorem 5 (PCP Theorem) **NP** *equals* **PCP** *where the verifier tosses* $O(\log n)$ *coins and reads only a* constant *number of bits from the proof.*