Probabilistic Proof Systems	May 28, 2021
Some Basic Facts about PCPs	
Instructor: Alessandro Chiesa	*Scribe: Shuangjun Zhang*

1 Some Special Case

We wish to understand $\mathcal{PCP}[\epsilon_c, \epsilon_s, \Sigma, l, q, r, ...]$ in different regimes. Let's start with some special cases to warm up.

Suppose there is no proof ($q = 0$):

- $\mathcal{PCP}[q = 0, r = 0] = \mathcal{P}$.
- $\mathcal{PCP}[q = 0, r = \mathcal{O}(\log n)] = \mathcal{P}$.
- $\mathcal{PCP}[q = 0, r = \text{poly}(n)] = \mathcal{BPP}$.

Suppose there is no randomness ($r = 0$):

- $\mathcal{PCP}[q = \text{poly}(n), r = 0] = \mathcal{NP}$.

We denote by $\mathcal{PCP}$ the complexity class with no restrictions beyond "V is PPT". This means that $q = \text{poly}(n), r = \text{poly}(n)$ and allow for $l = \exp(n), |\Sigma| = \exp(n)$.

2 Upper Bound and Lower Bound on PCPs

Theorem 1 (Upper Bound) $\mathcal{PCP} \subseteq \mathcal{NEXP}$.

Lemma 2 *The proof length $l \leq 2^r q$ for non-adaptive verifiers, and $l \leq 2^r |\Sigma|^q q$ for adaptive verifiers. (in constructions l is usually smaller than these upper bounds)*

Proof: For non-adaptive verifier, there are at most 2^r different query sets, and for adaptive one each answer from the proof can lead to a different next query. □

Lemma 3 $\mathcal{PCP}[l, r] \subseteq \mathcal{NTIME}((2^r + l) \cdot \mathit{poly}(n))$.

Proof: Suppose (P, V) is a $\mathcal{PCP}$ system for L where the PCP verifier users r random bits to query a proof of length l. Consider the decider:

- $D(x, \pi) :=$ For every $\rho \in \{0,1\}^r$ compute $b_\rho := V^\pi(x; \rho)$ and output 1 if and only if $\Sigma_\rho b_\rho / 2^r \geq 1 - \epsilon_c$

If $x \in L$, then $\exists \pi$ s.t. $D(x, \pi) = 1$. If $x \notin L$ then $\forall \pi$, $D(x, \pi) = 0$. $\square$

The upper bound theorem follows from this two lemma.

Theorem 4 (Lower Bound) $\mathcal{PSPACE} \subseteq \mathcal{PCP}$

Proof: We prove $\mathcal{IP} \subseteq \mathcal{PCP}$.

Suppose that (P, V) is a public-coin IP for L. Consider proofs in this format: $\pi = \{a_{r_1}\}_{r_1} \cup \{a_{r_1,r_2}\}_{r_1,r_2} \cup \{a_{r_1,...,r_k}\}_{r_1,...,r_k}$ The PCP verifier samples $r_1, ..., r_k$ and accepts if the IP verifier accepts:

$$V(x, a_{r_1}, a_{r_1,r_2}, ..., a_{r_1,...,r_k}; r_1, ..., r_k) \stackrel{?}{=} 1.$$

- Completeness: consider the honest proof

$$\pi = \{P(x, r_1)\}_{r_1} \cup \{P(x, r_1, r_2)\}_{r_1,r_2} \cup \{P(x, r_1, ..., r_k)\}_{r_1,...,r_k}.$$

- Soundness: any proof in the above format corresponds to an "unrolled" IP prover.

$\square$

In summarize, $\mathcal{PSPACE} \subseteq \mathcal{PCP} \subseteq \mathcal{NEXP}$. We will see that $\mathcal{PCP} = \mathcal{NEXP}$ by recycling techniques (arithmetization, sumcheck) and using new ones (Low Degree Testing), we will also see how to "scale down" to get PCPs for $\mathcal{NP}$.