

Advanced Proof-Systems - Problem Set 3

Yosef Goren

December 31, 2022

1 *MIP* vs *PCP*

1.1

Denote the concatenation of m_1 and m_2 with $m_1 || m_2$.

Let $L \subseteq \{0, 1\}^*$ be a language with a k – prover, 2 – message *MIP*.

Since this protocol is a 2 – message *MIP*, we know the structure of communications: in the first round the verifier sends a message to all provers, and in the second round each prover sends a message to the verifier.

Let $x \in L$, in the run of the protocol on x , Denote the message sent by the V to P_i with $m_{q,i}$, denote the message sent by P_i to V with $m_{r,i}$.

The set of possible values of $m_{q,i}$ is bounded with it's maximal size. Each such value will yield an appropriate $m_{r,i}$ response, which is independent of any other $m_{q,j}$ 'queries' sent by the verifier.

Thus for each $i \in [n]$ we can define a function from query values to response values; for any query value we can define the response value to be 0. Thus we are left with a function $Resp_i : \{0, 1\}^{l_v} \rightarrow \{0, 1\}^{l_p}$, where $\forall m_{q,i} \forall, Resp_i(m_{q,i}) = m_{r,i}$.

For the purposes of accessing the correct response later we can also pad the response values with 0's so that all of them are exactly l_P in length.

Define the *PCP* proof of x as:

$$\begin{aligned} PCP_x &:= Resp_0(0) || Resp_0(1) || \dots || Resp_0(2^{l_v} - 1) \\ &\quad || Resp_1(0) || Resp_1(1) || \dots || Resp_1(2^{l_v} - 1) \\ &\quad \dots \\ &\quad || Resp_{k-1}(0) || Resp_{k-1}(1) || \dots || Resp_{k-1}(2^{l_v} - 1) \end{aligned}$$

Note $|Resp_i(j)| = l_P$, thus each line in the definition of PCP_x is equal to $l_P \cdot 2^{l_v}$. And the whole size is equal to $k \cdot l_P \cdot 2^{l_v}$.

The *PCP* verifier V_p will be based on the *MIP* verifier V_m .

On the run of $V_p(x, PCP_x)$, V_p will first use V_m to ask it what queries to make. For each query $m_{q,i}$ to P_i made by V_m , V_p will look at:

$$PCP_x[l_P \cdot (2^{l_v} \cdot i + m_{q,i}) : l_P \cdot (2^{l_v} \cdot i + m_{q,i}) + l_P]$$

Or in other words, the bits corresponding to the response for $m_{q,i}$ in P_i 's section of PCP_x .

After the PCP verifier gets all these bits, it gives them back to V_m as the responses to the queries - and accepts iff V_m accepts.

- Completeness:

If $x \in L$ - our PCP_x is well defined. And for whatever query V_m makes - it receives the exact response it should get from P_i - thus since it will accept on P_i 's responses - it will accept on the messages sent by V_p , meaning V_p will accept on x .

- Soundness:

Let there be a set of PCP_x values (defined also on $x \notin L$). WLOG $\forall x, |PCP_x| = l_P$ - since it is easy for V to check if that is the case. Thus we can easily use this set of PCP_x 's to construct a set of 'malicious' provers P_0, \dots, P_{k-1} : Each P_i on query $m_{q,i}$ will response with:

$$PCP_x[l_P \cdot (2^{l_V} \cdot j + m_{q,i}) : l_P \cdot (2^{l_V} \cdot j + m_{q,i}) + l_P]$$

Now for each instance where $V_p(x, PCP_x) = 1$, we have $(V_m, P_0, \dots, P_{k-1})(x) = 1$ since the only way for V_p to accept is if V_m does - and it runs on the same inputs (including randomizations) in both cases.

Thus:

$$\Pr[V_p(x, PCP_x) = 1] \leq \Pr[(V_m, P_0, \dots, P_{k-1})(x) = 1] \leq \frac{1}{2}$$

1.2

Let $L \subseteq \{0, 1\}^*$ be a language with a PCP verifier V_p set of proofs bounded by in length m .

Define the following q - prover MIP for it:

The protocol on input x :

- Verifier V_m :

1. sample $B \leftarrow^{\$} \{0, 1\}$.
2. if $B = 0$ (verify PCP):
 - (a) Get the set of queries $Q = \{Q_i \mid i \in [q]\}$ from V_p on input x .
 - (b) For all $i \in [q]$, send Q_i to P_i .
 - (c) Denote the bit returned by P_i with b_i .
 - (d) Verify V_p 's acceptance on query results $\{b_i \mid i \in [q]\}$.
3. if $B = 1$ (Verify consistency):
 - (a) Sample $r \leftarrow^{\$} [m]$.
 - (b) For all $i \in [q]$, send r to P_i .

- (c) Denote the bit returned by P_i with b_i .
- (d) Verify $b_i = b_j, \forall i, j \in [q]$.

- Prover P_i :

1. Recive an index i from the verifier.
2. Return $PCP_x[i]$.

Correctness:

- Complexity:

The integer representation of each query is of size $\log(m)$, thus it is the length of the messages sent by V_m .

- Completness:

Let $x \in L$. Denote with $PCP_x[Q]$ the set of bits corresponding to the queries Q in PCP_x .

In the standard usage of the PCP verifier V_p - it will recive $PCP_x[Q]$ as the responses to the queries Q - and since it has perfect completeness (WLOG as seen previously in the course), it will accept.

When V_m invokes V_p - it sends it the same $PCP_x[Q]$, thus V_p accepts here too - and so does V_m .

- Soundness:

Let $x \notin L$.

Let $\{P_i^* \mid i \in [q]\}$ be a set of (possibly mallicious) provers.

Now we use these provers to construct PCP_x :

PCP_x has a section corresponding to each prover P_i^* , and in each section - the j 'th bit corresponds to the response of P_i^* on query j which is the most probable. If there is more than one most probable response select the minimal one.

As a shorthand denote P_1^*, \dots, P_q^* as PS^* .

Denote with $C_{i,j}$ the event that P_i returns the most probable response on query j and $c_{i,j} = \Pr[C_{i,j}]$. Since there are only two possible responses (1 and 0) - $c_{i,j} \geq \frac{1}{2}$.

In the event that $B = 0$:

Let $C := \bigwedge_{i \in [q]} C_{i,Q_i}$ - meaning the event that all provers were consistent with PCP_x . Let $c = \Pr[C]$.

Since we know the provers cannot communicate - the probabilities for their selections are independent. Hence $c = \mathbb{E}_Q[\prod_{i \in [q]} c_{i,Q_i}]$.

$$\Pr[(V_m, PS^*)(x) = 1 \mid B = 0]$$

$$\begin{aligned}
&= \Pr[(V_m, PS^*)(x) = 1 \mid B = 0 \wedge C] \Pr[C] + \Pr[(V_m, PS^*)(x) = 1 \mid B = 0 \wedge \neg C] \Pr[\neg C] \\
&= c \Pr[(V_m, PS^*)(x) = 1 \mid B = 0 \wedge C] + (1-c) \Pr[(V_m, PS^*)(x) = 1 \mid B = 0 \wedge \neg C] \\
&= c \Pr[V_p(x, PCP_x) = 1] + (1-c) \Pr[(V_m, PS^*)(x) = 1 \mid B = 0 \wedge \neg C] \\
&\leq c \Pr[V_p(x, PCP_x) = 1] + (1-c) \cdot 1 = 1 + c(\Pr[V_p(x, PCP_x) = 1] - 1)
\end{aligned}$$

In the event that $B = 1$:

Let $C := \bigwedge_{i \in [q]} C_{i,r}$ and $c' := \Pr[C]$.

Thus $c = \frac{1}{q} \sum_{r \in [q]} \prod_{i \in [q]} c_{i,r}$.

$$\begin{aligned}
&\Pr[(V_m, PS^*)(x) = 1 \mid B = 1] \\
&\Pr[(V_m, PS^*)(x) = 1 \mid B = 1 \wedge C] \Pr[C] + \Pr[(V_m, PS^*)(x) = 1 \mid B = 1 \wedge \neg C] \Pr[\neg C] \\
&= c \Pr[(V_m, PS^*)(x) = 1 \mid B = 1 \wedge C] + (1-c) \Pr[(V_m, PS^*)(x) = 1 \mid B = 1 \wedge \neg C] \\
&= c \cdot 1 + (1-c) \cdot 0 = c
\end{aligned}$$

Thus:

$$\begin{aligned}
\Pr[(V_m, PS^*)(x) = 1] &= \frac{1}{2} \Pr[(V_m, PS^*)(x) = 1 \mid B = 0] + \frac{1}{2} \Pr[(V_m, PS^*)(x) = 1 \mid B = 1] \\
&\leq \frac{1 + c(\Pr[V_p(x, PCP_x) = 1] - 1)}{2} + \frac{c}{2} = \frac{1 + \Pr[V_p(x, PCP_x) = 1]}{2}
\end{aligned}$$

Hence since V_p has constant soundness - V_m has constant soundness too.