Foundations of Probabilistic Proofs (Fall 2024)
Note 12: Limitations of IPs, PCPs and IOPs
Date: 2024.12.03

---

1 On laconic provers with perfect completeness

During lecture, we have seen various limitations on which languages can have IPs. We have shown a number of results IPs with bounded total communication between prover and verifier. The case in which only prover to verifier communication is bounded is harder. We refer to this set as that of "laconic provers". In this recitation, we show how to prove one of the results mentioned at the end of the lecture, namely that

$$\mathsf{IP}[\epsilon_{\mathsf{c}} = 0, \mathsf{pc}] \subseteq \mathsf{coNTIME}(2^{\mathsf{pc}} \cdot \mathsf{poly}(n)) \ .$$

To prove the statement, we will make use of Zermelo's theorem, which (informally) states that "*in every finite-stage two-player game with complete information and alternating moves one of the two players can force a win*".

1.1 Strategy

Let $(\mathcal{P}, \mathcal{V})$ be the honest prover and verifier of the IP we are considering. Our strategy will be the following:

- For any instance $\mathbb{x}$, we define a two player game $(\mathcal{P}_{\mathbb{x}}, \mathcal{V}_{\mathbb{x}})$. $\mathcal{P}_{\mathbb{x}}, \mathcal{V}_{\mathbb{x}}$ will alternate, and at each turn the player playing will output a message. The transcript of the game (which we denote as tr) will be the concatenation of all such messages. $\mathcal{V}_{\mathbb{x}}$ wins if it is able to produce a randomness ρ such that $\mathcal{V}(\mathbb{x}, \mathsf{tr}; \rho) = 0$ and tr is consistent with the choice of ρ. $\mathcal{P}_{\mathbb{x}}$ wins otherwise.
- We analyze the game behaviour depending on whether $\mathbb{x} \in \mathcal{L}$ and $\mathbb{x} \notin \mathcal{L}$ (the latter case will invoke Zermelo's theorem).
- Finally, we use the winning strategy of the game to construct a witness for the non-membership of $\mathbb{x} \notin \mathcal{L}$.

2 Limitations of IOPs

In the lecture, we consider limitations of soundness of PCPs, in both the setting in which the PCP has perfect completeness and when it does not. We revisit these ideas in the IOP settings, especially focusing on the "algorithms for IOPs" result sketched in the lecture.

2.1 Strategy

First, we show how to convert an IOP into a laconic IP (by possibly increasing the completeness error).

- The IP prover initiates by guessing the choice of queries that the IOP verifier will perform during the interaction, and sends the guessed choice.
- IP prover and verifier interact as in the IOP interaction, but instead of sending the proof oracle, the IP prover instead just sends the proof string elements corresponding to the guessed choice of queries.
- The IP verifier emulates the IOP verifier decision phase, using the sent messages instead of querying the proof oracles. If at any point the IOP verifier queries a location not specified in the initial IP prover message, it rejects.

The analysis of this protocol is similar to those in transforming a PCP into a laconic MA protocol.

Since the previous transformation transforms a public-coin IOP into a public-coin laconic IP, the result that $\mathsf{AM}[c,k] \subseteq \mathsf{BPTIME}[2^{O(c(n)+k(n)\cdot\log k(n))} \cdot \mathsf{poly}(n)]$ suffices to yield the limitation.