Precis: Local Proofs Approaching Witness Length

Prashanth Ramakrishna

This precis gives a high-level overview of the paper *Local Proofs Approaching the Witness Length* by Noga Ron-Zewi and Ron Rothblum. We focus on the main results of the paper, ignoring subtleties of complexity and parameter sets. Further, we assume knowledge of the paper's preliminaries, using freely the definitions and notation contained therein. When necessary, intuitive rather than technical explanations will be provided. Note that we take much instruction from the extended preprint of the paper, which provides more explicit constructions. However, this means that some language may be inconsistent with the final publication.

Motivation. The primary goal of this work is to construct an IOP for NP relations that satisfies (1) constant query complexity, (2) constant round complexity, and (3) proof length $(1+\gamma)\cdot n$ where n is the length of the NP witness and any constant $\gamma > 0$.

Prior to this work, optimally short proofs for general NP relations, typically constructed as PCPs, were obstructed by a communication dependence on verification complexity. This dependence was an artifact of having to encode the witness along with the entire computation using an error correcting code (usually with undesirable encoding overhead). The novelty of this paper's results emerges primarily as a consequence of leveraging interaction (IOP rather than PCP constructions) to make communication complexity dependent *only* on NP witness length and, further, leveraging a new high-rate, constant distance tensor code to drastically reduce encoding overhead. These two developments result in transcript size that *approaches witness length* for an IOP that supports polynomial-space bounded NP relations.

High-level construction. The clearest starting point for the construction is the observation that given an $\mathcal{S}$-linear IOP reduction for an NP relation $\mathcal{R}$ and $\mathcal{S}$-sumcheckable ($\mathcal{S}$-SC) code, one can readily obtain an IOPP for $\mathcal{R}$ via a technique called *code switching*, which uses a low-rate multiplication code to check the correctness of the computation but a high-rate tensor product code to encode the witness. The resulting IOPP begins with $\mathcal{P}$ encoding the witness and sending $\mathcal{V}$ the non-systematic portion of the resulting code word. The protocol then proceeds straightforwardly. First, the protocol runs the $\mathcal{S}$-linear IOP reduction on the statement-witness pair to obtain a linear claim on the witness that can be checked with oracle access to its encoding. Then, the protocol runs the IOPP associated with the $\mathcal{S}$-SC code to check the linear claim. Verification only succeeds if verification succeeds for the $\mathcal{S}$-SC code IOPP and no rejections take place at any point prior. Complexity metrics are roughly additive, since the two constituent protocols are run sequentially.

The question immediately becomes how to construct a particular $\mathcal{S}$-linear IOP reduction and $\mathcal{S}$-SC code that are not only composable via code-switching but also exhibit the desired complexity. To this end a particular subset ensemble $\mathcal{S}_{\otimes t}$ is introduced consisting of all vectors (over a particular finite field ensemble) that can be interpreted as rank-1 tensors. To obtain a $\mathcal{S}_{\otimes t}$-linear IOP reduction, the authors begin with the RRR protocol for languages computable in polynomial time and bounded space. However, the protocol's verifier requires oracle access to the low degree extension of the witness and may potentially make many queries to the extension. Because in this setting such oracle access is not provided to the verifier, instead it is necessary to *emulate* queries to the low degree extension of the witness. This is

made possible by the fact that the low degree extension is itself a tensor code and therefore the particular linear claims generated by the RRR verifier feature coefficients with rank-1 tensor structure. With this in hand, transforming the RRR protocol into the desired IOP reduction follows from application of *query reduction* so that only a single linear claim is output by the verifier and *robustification* so that for any view close to that of a verifier interacting with a cheating prover will lead either to rejection or a unique linear claim that is false with high probability. Query reduction leverages the sumcheck protocol on a particular product of low-degree extensions, while robustification trivially amounts to encoding prover messages using an error correcting code with sufficient functionality. We note that the restriction of the paper's results to languages that are computable in polynomially bounded space is inherited from the RRR protocol.

The $\mathcal{S}_{\otimes t}$-SC code employed by the authors is the t-dimensional tensor product code of a high-rate binary base code. The authors construct this code explicitly as a variant of the high-rate Justesen code. An IOPP for such a code is then constructed using a variety of nice features for tensor product codes. In particular, tensor product codes come equipped with a sumcheck protocol, which can be augmented by a randomized local testing procedure, a randomized local correcting procedure, and local decomposability, to obtain the desired IOPP.

To arrive at the desired properties described at the outset, the authors deploy one final trick known as *proof composition*. Briefly, proof composition is a well known technique from the world of PCPs that can be extended to the world of IOPs. In the world of PCPs, proof composition enables an "outer" PCP to be combined with an "inner" PCP in order to achieve a desirable combination of their properties. Importantly, however, the outer PCP must be *robust* and the inner PCP must be a proximity proof in order to satisfy soundness in the composition. The composed PCP has approximately the same prover complexity of the outer PCP, the verifier complexity of the inner PCP, and consists of the outer PCP string along with an inner PCP string for each choice of randomness. Extending to the world of IOPs, composition can be performed when the outer IOP is robust and the inner IOP is an IOPP. Here, interaction brings efficiency gains for free since the inner IOPP need only be invoked once after running the outer IOP. In addition, we note that the first message of the composed prover is identical to the first message of the outer IOP prover (this is essential to our setting, as the first prover message dominates communication).

In the context of this work, the authors wish to compose a short, robust outer IOP with large query complexity and a long inner IOPP with small query complexity in order to obtain a final IOP that has both small communication and query complexity. This does not come directly, however. Rather, two IOPPs are composed, then the composition is subjected to transformation into an IOP for non-deterministic languages. The robust outer IOPP is constructed via code-switching from the $\mathcal{S}_{\otimes t}$-linear IOP reduction and $\mathcal{S}_{\otimes t}$-SC code, while the inner IOPP is chosen off the shelf in a manner that enables reducing query complexity to a constant in the composition. Lastly, the transformation to a constant round IOP for non-deterministic (but still bounded space) languages is carefully performed to avoid even a constant blowup in communication. The resulting constant query IOP is ultimately dominated by a deterministic, one-time first prover message of $(1 + \frac{\gamma}{2}) \cdot n$ bits, achieving the desired result.