Composition of low-error 2-query PCPs using decodable PCPs[*]

Irit Dinur[†] Prahladh Harsha[‡]

Abstract

The main result of this paper is a generic composition theorem for low error two-query probabilistically checkable proofs (PCPs). Prior to this work, composition of PCPs was well-understood only in the constant error regime. Existing composition methods in the low error regime were non-modular (i.e., very much tailored to the specific PCPs that were being composed), resulting in complicated constructions of PCPs. Furthermore, until recently, composition in the low error regime suffered from incurring an extra 'consistency' query, resulting in PCPs that are not 'two-query' and hence, much less useful for hardness-of-approximation reductions.

In a recent breakthrough, Moshkovitz and Raz [In *Proc. 49th IEEE Symp. on Foundations of Comp. Science (FOCS)*, 2008] constructed almost linear-sized low-error 2-query PCPs for every language in NP. Indeed, the main technical component of their construction is a novel composition of certain specific PCPs. We give a modular and simpler proof of their result by repeatedly applying the new composition theorem to known PCP components.

To facilitate the new modular composition, we introduce a new variant of PCP, which we call a *decodable PCP (dPCP)*. A dPCP is an *encoding* of an NP witness that is both locally checkable and locally decodable. The dPCP verifier in addition to verifying the validity of the given proof like a standard PCP verifier, also locally decodes the original NP witness. Our composition is generic in the sense that it works regardless of the way the component PCPs are constructed.

1 Probabistically Checkable Proofs – Introduction

Probabilistically checkable proofs (PCPs) provide a proof format that enables verification with only a constant number of queries into the proof. This is formally captured by the (by now standard) notion of a probabilistic verifier.

Definition 1.1 (PCP Verifier)**.** *A PCP verifier V for a language L is a polynomial time probabilistic algorithm that behaves as follows: On input x, and oracle access to (proof) string π (over an alphabet Σ), the verifier reads the input x, tosses some random coins r, and based on x and r computes a window $I = (i_1, \ldots, i_q)$ of indices to read from π, and a predicate $f : \Sigma^q \to \{0, 1\}$. The verifier then accepts iff $f(\pi_I) = 1$.*

[*] A full version of this paper appears in the Electronic Colloquium on Computational Complexity [DH09]. The current extended abstract is a modification of the introduction of the full version for the purposes of the ICS mini-workshop on propert testing.

[†] Weizmann Institute of Science, ISRAEL. email: `irit.dinur@weizmann.ac.il`. Research supported in part by the Israel Science Foundation and by the Binational Science Foundation.

[‡] Tata Institute of Fundamental Research, Mumbai, INDIA. email: `prahladh@tifr.res.in`. Research done while the author was at Technion, Israel Institute of Technology and was supported in part by a fellowship from the Aly Kaufman Foundation

- *The verifier is* complete *if for every $x \in L$ there is a proof π accepted with probability* 1*. I.e.,* $\exists \pi,\ \Pr_{I,f}[f(\pi_I) = 1] = 1$.
- *The verifier is* sound *with* soundness error $\delta < 1$ *if for any $x \notin L$, every proof π is accepted with probability at most δ. I.e.,* $\forall \pi,\ \Pr_{I,f}[f(\pi_I) = 1] \leq \delta$.

The celebrated PCP Theorem [AS98, ALM+98] states that every language in NP has a verifier that is complete and sound with a constant $\delta < 1$ soundness error while using only a logarithmic number of random coins, and reading only $q = O(1)$ proof bits. Naturally, (and motivated by the fruitful connection to inapproximability due to [FGL+96]), much attention has been given to obtaining PCPs with "good" parameters, such as $q = 2$, smallest possible soundness error δ, and smallest possible alphabet size $|\Sigma|$. These are the parameters of focus in this paper.

How does one construct PCPs with such remarkable proof checking properties? In general, it is easier to construct such PCPs if we relax the alphabet size $|\Sigma|$ to be large (typically super-constant, but sub-exponential). This issue is similar to a well-known issue that arises in coding theory; wherein it is relatively easy to construct codes with good error-correcting properties over a large, super constant sized, alphabet (e.g., Reed-Solomon codes). Codes over a constant-sized alphabet (e.g., GF(2)) are then obtained from these codes by (repeatedly) applying the "code-concatenation" technique of Forney [For66]. The equivalent notion in the context of PCP constructions is the paradigm of "proof composition", introduced by Arora and Safra [AS98]. Informally speaking, proof composition is a recursive procedure applied to PCP constructions to reduce the alphabet size. Proof composition is applied (possibly several times over) to PCPs over the large alphabet to obtain PCPs over a small (even binary) alphabet.

Proof composition is an essential ingredient of all known constructions of PCPs. Composition of PCPs with high soundness error (greater than 1/2) is by now well understood using the notion of *PCPs of proximity* [BGH+06] (called *assignment testers* in [DR06]) (see also [Sze99]). These allow for modular composition, in the high soundness error regime which in turn led to alternate proofs of the PCP Theorem and constructions of shorter PCPs [BGH+06, Din08, BS08]. However, these composition theorems are inapplicable when constructing PCPs with low-soundness error (arbitrarily small soundness error or even any constant less than 1/2). (See survey on constructing low error PCPs by Dinur [Din08] for a detailed explanation of this limitation).

Our first contribution is a definition of an object which we call a *decodable PCP*, which allows for clean and modular composition in the low error regime.

2 Decodable PCPs (dPCPs)

Consider a probabilistically checkable proof for the language CircuitSat (the language of all satisfiable circuits). The natural NP proof for CircuitSat is simply a satisfying assignment. An intuitive way to construct a PCP for CircuitSat is to *encode* the assignment in a way that enables probabilistic checking. This intuition guides all known constructions, although it is not stipulated in the definition.

In this work, we make the intuitive notion of proof encoding explicit by introducing the notion of a *decodable PCP (dPCP)*. A dPCP for CircuitSat is an encoding of the satisfying assignment that can be both verified and decoded locally in a probabilistic manner. In this setting, the verifier is supposed to both verify that the dPCP is encoding a *satisfying* assignment, as well as to decode a symbol in that assignment. More precisely, we define a *PCP decoder* for CircuitSat to be (along the lines of Definition 1.1) a probabilistic algorithm that is given an input circuit C, oracle access to a dPCP π, and, in addition, an index i. Based on C, i and the randomness r it computes a window I and a *function* f (rather than a predicate). This function is supposed to evaluate to the i-th symbol of a satisfying assignment for C; or to reject.

- The PCP decoder is *complete* if for every y such that $C(y) = 1$ there is a dPCP π such that $\Pr_{i,I,f}[f(\pi_I) = y_i] = 1$.
- The PCP decoder has *soundness error* δ and list size L if for any (purported) dPCP π there is a list of $\leq \mathsf{L}$ valid proofs such that the probability (over the index i and (I, f)) that $f(\pi_I)$ is inconsistent with the list but not reject is at most δ.

The list of valid proofs can be viewed as a "list decoding" of the dPCP π. Since we are interested in the low soundness error regime, list-decoding is unavoidable. Of course, we can define dPCPs for any NP language and not just CircuitSat, but we focus on CircuitSat since it suffices for the purpose of composition.

The notion of dPCPs allows for modular composition in the case of low soundness error (described next) in analogy to the way PCPPs and assignment testers [BGH+06, DR06] allow for modular composition in the case of high soundness error. Moreover, using dPCPs we show a two query composition that yields a completely modular proof of the recent result of Moshkovitz and Raz [MR08b].

Finally, we note that decodable PCPs are not hard to come by. Decodable PCPs or variants of them are implicit in many PCP constructions [AS03, RS97, DFK+99, BGH+06, DR06, MR07, MR08b] and existing PCP constructions can often be adapted to yield decodable PCPs.

3 Composition with dPCPs

There is a natural and modular way to compose a PCP verifier V with a PCP decoder $\mathcal{D}$. The composed PCP verifier V' begins by simulating V on a probabilistically checkable proof Π. It determines a set of queries into Π (a local window I), and a local predicate f. Instead of directly querying Π and testing if $f(\Pi_I) = 1$, V' relies on the inner PCP decoder $\mathcal{D}$ to perform this action. For this task, the inner PCP decoder $\mathcal{D}$ is supplied with a dedicated proof that is supposedly an encoding of the relevant local view Π_I. The main issue is consistency: the composed verifier V' must ensure that the dedicated proofs supposedly encoding the various local views are consistent with the same Π (i.e. they should be encodings of local views coming from a single valid PCP for V). This is achieved easily with PCP decoders: the composed verifier V' asks $\mathcal{D}$ to decode a random value from the encoded local view, and compares it to the appropriate symbol in Π.

The above description of composition already appears to lead to a modular presentation of the composition performed in earlier low-error PCP constructions [AS03, RS97, DFK+99, MR07]. But at the same time, like these compositions, it incurs an additional query per composition, namely the "consistency" query to the outer PCP Π. (The queries made by V' are the queries of $\mathcal{D}$ plus the one additional consistency query to Π).

Nevertheless, inspired by [MR08b] and equipped with a better understanding of composition in the low soundness error case, we are, now, in a position to remove this extra consistency query.

4 Composition with only two queries

Our main contribution is a composition theorem that does not incur an extra query. The extra query above comes from the need to check that all the inner PCP decoders decode to the same symbol. This check was performed by comparing the decoded symbol to the symbol in the outer PCP Π. Instead, we verify consistency by invoking *all* the inner PCP decoders that involve this symbol *in parallel*, and then checking that they all decode to the same symbol. This avoids the necessity to query the outer PCP Π for this symbol and saves us the extra query.

We describe our new composed verifier V' more formally below. As before, let V be a PCP verifier, and $\mathcal{D}$ a PCP decoder.

1. The composed PCP verifier simulates V on a hypothetical PCP Π; it chooses a random index i in Π, and then determines *all* the possible random strings $R_1, \ldots, R_D$ that cause V to query this index.

2. For each random string R_j ($j = 1 \ldots D$), V' needs to check that the corresponding local view of Π would have lead V to accept. This is done by running $\mathcal{D}$, for each $j = 1 \ldots D$, on a dedicated proof $\pi(R_j)$ that is supposedly the encoding of the j-th local view (i.e., the one generated by V on random string R_j) into Π. Furthermore, V' expects $\mathcal{D}$ to decode the symbol Π_i.

3. Finally V' accepts if and only if *all* the D parallel runs of $\mathcal{D}$ accept and output the same symbol.

Observe that the composed verifier V' does not access the PCP for V (i.e., Π) at all, rather only the dedicated proofs for the inner PCP decoders. The outer PCP Π is only "mentally" present in order to compute $R_1, \ldots, R_D$. A few important points are in order.

- **Two Queries and Robust Soundness** As described, V' makes many queries rather than just two. This is fixed by the following easy transformation: the first query will supposedly be answered by the complete local view V' expects to read, and the second query will consist of one random symbol in the local view of V'. The soundness of the resulting two-query PCP is equal to the *robust soundness* of V': an upper bound on the average agreement between a local view read by V' and an accepting local view.

  Thus, drawing on the above correspondence, the fact that V' has low robust soundness implies the required two-query composition. Of course, the composition could have been described entirely in the 2-query PCP language.

- **Size of alphabet or window size** The purpose of composition is to reduce the alphabet size, or, in the language of robust PCPs, to reduce the window size, that is, the number of queries made by V'. Recall that V' runs $\mathcal{D}$ in parallel on all D local views corresponding to $R_1, \ldots, R_D$. Thus, the window size equals the query complexity of $\mathcal{D}$ multiplied by the number D of local views (which we refer to as the *proof degree* of V). Hence composition is meaningful only if the proof degree is small to begin with (otherwise, the local window of V' is not smaller than that of V and we haven't gained anything from composition). In general PCPs, the proof degree is very high. In fact, this has been one of the obstacles to achieving this result prior to [MR08b]. However, a key observation of [MR08b] is that it is easy to reduce the proof degree using standard tools from derandomization (i.e., expander replacement).

Viewed alternatively, one can handle V of arbitrarily high proof degree by making the following change to V'. Instead of running $\mathcal{D}$ to verify the local tests corresponding to *all* of $R_1, \ldots, R_D$, V' can *pseudo-randomly* sample a small number of these and run $\mathcal{D}$ only on the selected ones.

The fact that the query complexity is at least D is an inherent bottleneck in our composition method. Combined with the bound of $D \geq 1/\delta$, this poses a limitation of this technique towards achieving exponential dependence of the error probability on alphabet size, a point discussed later in this introduction.

The new composition is generic in the sense that it works regardless of how the original components V and $\mathcal{D}$ are constructed.

5 Background and Motivation

Let us step back to give some motivation for obtaining PCPs with small soundness and two queries (for a more comprehensive treatment, see [MR08b]). Two is the absolute minimal number of queries possible for a non-trivial PCP. Thus, it is interesting to find what are the strongest 2-query PCPs that still capture NP. However, the main motivation for two query PCPs is for proving hardness of approximation results.

Two query PCPs with soundness error δ are (more or less) equivalent to Label-Cover$_\delta$, which is a promise problem defined as follows: The input is a bipartite graph and an alphabet Σ, and for each edge e there is a function $f_e : \Sigma \to \Sigma$, which we think of as a *constraint* on the labels of the vertices. The constraint is satisfied by values a and b iff $f_e(a) = b$. The problem is to distinguish between two cases: (1) there exists a labeling of the vertices satisfying all constraints, or (2) every labeling satisfies at most δ fraction of the constraints.

Label-Cover$_\delta$ is probably the most popular starting point for hardness of approximation reductions. In particular, even though there are 3-query PCPs with much smaller soundness error, they currently have far fewer applications to inapproximability.

The fact that Label-Cover$_\alpha$ is NP-hard for some constant $\alpha < 1$ (and constant alphabet size) is nothing but a reformulation of the PCP Theorem [AS98, ALM$^+$98]. Strong inapproximability results, however, require[1] NP-hardness of Label-Cover$_\delta$ for arbitrarily small, sometimes even sub-constant soundness error δ. There are two known routes to obtaining hardness results for Label-Cover$_\delta$ with small soundness δ. The first, is via an application of the parallel repetition theorem of Raz [Raz98] to the Label-Cover$_\alpha$ instance produced by the PCP Theorem. However, this application of the repetition theorem blows up the size of the problem instance from n to $n^{O(\log(1/\delta))}$ and thus remains polynomial only for constant, though arbitrarily small, δ. One might try to get a polynomial sized construction by carefully choosing a subset of the entire parallel repetition construction. This is known as the problem of "derandomizing the parallel repetition theorem". Feige and Kilian [FK95] showed that such derandomization is impossible under certain (rather general) conditions. Nevertheless, in a recent paper, Impagliazzo et. al. [IKW09] obtained

[1] In some cases the hardness gap is inversely proportional to δ, and in others, it is the sum of two terms: a problem-dependent term (e.g. 7/8 in Håstad's hardness result [Hås01] for 3-SAT), and a "low order" term that is polynomial in δ.

a related derandomization. While their derandomization result applies only to direct products and not to the construction of PCPs, this direction seems promising. Another potential direction is to use the gap-amplification technique of Dinur [Din07], however as shown by Bogdanov [Bog05] gap-amplification fails below a soundness error of 1/2.

The second route to sub-constant δ goes through the classical (algebraic) construction of PCPs. Indeed, hardness for label cover with sub-constant error can be obtained from the low soundness error PCPs of [RS97, AS03, MR08a], more or less by omitting the composition steps, and carefully combining queries. The following "manifold vs. point" PCP construction has been folklore since [RS97, AS03], and formally described in [MR08b].

Theorem 5.1 (Manifold vs. Point PCP)**.** *There exists a constant $c > 1$ such that the following holds: For every $\frac{1}{n} \leq \delta \leq \frac{1}{(\log n)^c}$, there exists an alphabet Σ of size at most $\exp(\mathrm{poly}(1/\delta))$ such that* Label-Cover$_\delta$ *over Σ is NP-hard.*

The above result is unsatisfactory as the size of the alphabet $|\Sigma|$ is super-polynomial. Combined with the fact that hardness-of-approximation reductions are usually exponential in $|\Sigma|$ (and always at least polynomial in $|\Sigma|$) the super polynomial size of Σ renders the above theorem useless. The situation can be redeemed if the theorem could be extended to the entire range of smaller $|\Sigma|$ (with a corresponding increase in δ).

A natural way to perform this extension would be to apply the composition paradigm to the PCPs constructed in Theorem 5.1 and reduce the alphabet size. Indeed, this is how one constructs PCPs with sub-constant error and *a constant* number of queries for the entire range of $\Omega(1) \leq |\Sigma| \leq \exp((\log n)^{1-\varepsilon})$ [RS97, AS03, DFK$^+$99]. However, the composition a la [RS97, AS03, DFK$^+$99] incurs at least one additional query, which means that the final PCP is no longer "two-query", so it does not lead to a hardness result for label cover. Alternatively, the composition technique of [BGH$^+$06, DR06] using PCPs of proximity or assignment testers is inapplicable in this context as it fails to work for soundness error less than 1/2. Thus, all earlier composition techniques are either inapplicable in the low error regime or if applicable, incur an extra query and thus, are no longer in the framework of the Label-Cover problem.

6 The Two-Query PCP of Moshkovitz and Raz [MR08b]

In a recent breakthrough, [MR08b] show that the above theorem can in fact, be extended to the entire range of δ and $|\Sigma|$ (and maintaining $|\Sigma| \approx \exp(\mathrm{poly}(1/\delta))$). This is done by composing certain specific 2-query PCPs with low soundness error without incurring an additional query per composition.

Theorem 6.1 ([MR08b])**.** *For every $\delta \in (1/\mathrm{polylog} n, 1)$, there exists an alphabet Σ of size at most $\exp(\mathrm{poly}(1/\delta))$ such that* Label-Cover$_\delta$ *over Σ is NP-hard (in fact, even under nearly length preserving reductions).*

The main technical component of their construction is a novel composition of certain specific PCPs. However, the construction is so organically tied to the specific algebraic components that are being composed, as to make it extremely difficult to differentiate between the details of the PCP, and what it is that makes the composition go through.

We give a modular and simpler proof of this theorem using our composition theorem. Our proof relies on a PCP system based on the manifold vs. point construction. The parameters we need are rather weak: it is enough that on input size n the PCP decoder / verifier makes n^{α} queries and has soundness error $\delta = 1/n^{\beta}$, for small constants α, β. After one composition step the number of queries goes (roughly) from n^{α} to n^{α^2}, and so on. After each composition step we add a combinatorial step, consisting of degree and alphabet reduction, that prepares the verifier for the next round of composition. After i rounds the number of queries is about n^{α^i}, and the soundness error is about $\delta = 1/n^{O(\alpha^i)}$. Choosing $1 \leq i \leq \log\log n$ appropriately gives us the result.

The modular composition theorem allows us to easily keep track of a super-constant number of steps, thus avoiding the need for another tailor-made Hadamard-based PCP which was required in the proof of [MR08b]. (The later approach could also be implemented in our setting).

Randomness and the length of the PCP: The above discussion completely ignores the randomness complexity of the underlying PCPs. However, it is easy to verify that the composition described above is, in fact, randomness efficient; this is because the same inner randomness can be used for all the D parallel runs of the inner PCP decoder. Thus, if we start from a version of the Theorem 5.1 (the manifold vs. point PCP) based on an almost linear-size low-degree test (c.f., [MR08a]), we obtain a nearly length preserving version of Theorem 6.1 (i.e., a reduction taking instances of size n to instances of size almost linear in n). Furthermore, the fact that we account for the input index i separately from the inner randomness r of the PCP decoder leads to an even more randomness-efficient composition, however, we do not exploit this fact in the proof of Theorem 6.1.

Polynomial dependence of soundness error on alphabet size: Theorem 6.1 suffers from the following bottleneck: the error probability δ is inverse logarithmic (and not inverse-polynomial) with respect to the size of the alphabet Σ. This limitation is inherent in our composition method as discussed above. Thus, the "sliding-scale conjecture" of Bellare et al. [BGLR93] that for every $|\Sigma| \in (1, n)$, Label-Cover$_{\delta}$ over Σ is NP-hard for $\delta = \mathrm{poly}(1/|\Sigma|)$ remains open.

References

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. *Proof verification and the hardness of approximation problems.* J. ACM, 45(3):501–555, May 1998. (Preliminary Version in *33rd FOCS*, 1992). eccc:TR98-008, doi:10.1145/278298.278306.

[AS98] Sanjeev Arora and Shmuel Safra. *Probabilistic checking of proofs: A new characterization of NP.* J. ACM, 45(1):70–122, January 1998. (Preliminary Version in *33rd FOCS*, 1992). doi:10.1145/273865.273901.

[AS03] Sanjeev Arora and Madhu Sudan. *Improved low-degree testing and its applications.* Combinatorica, 23(3):365–426, 2003. (Preliminary Version in *29th STOC*, 1997). eccc:TR97-003, doi:10.1007/s00493-003-0025-0.

[BGH+06] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil Vadhan. *Robust PCPs of proximity, shorter PCPs and applications to coding.* SIAM J. Computing, 36(4):889–974, 2006. (Preliminary Version in *36th STOC*, 2004). eccc:TR04-021, doi:10.1137/S0097539705446810.

[BGLR93] Mihir Bellare, Shafi Goldwasser, Carsten Lund, and Alexander Russell. *Efficient probabilistically checkable proofs and applications to approximation.* In *Proc. 25th ACM Symp. on Theory of Computing (STOC)*, pages 294–304. ACM, 1993. doi:10.1145/167088.167174.

[Bog05] Andrej Bogdanov. *Gap amplification fails below 1/2*, 2005. (Comment on "Dinur, *The PCP theorem by gap amplification*"). eccc:TR05-046.

[BS08] Eli Ben-Sasson and Madhu Sudan. *Short PCPs with polylog query complexity.* SIAM J. Computing, 38(2):551–607, 2008. (Preliminary Version in *37th STOC*, 2005). eccc:TR04-060, doi:10.1137/050646445.

[DFK+99] Irit Dinur, Eldar Fischer, Guy Kindler, Ran Raz, and Shmuel Safra. *PCP characterizations of NP: Towards a polynomially-small error-probability.* In *Proc.* 31*st ACM Symp. on Theory of Computing (STOC)*, pages 29–40. ACM, 1999. eccc:TR98-066, doi:10.1145/301250.301265.

[DH09] Irit Dinur and Prahladh Harsha. *Composition of low-error 2-query PCPs using decodable PCPs.* Technical Report TR09-042, Electronic Colloquium on Computational Complexity, 2009. eccc:TR09-042.

[Din07] Irit Dinur. *The PCP theorem by gap amplification.* J. ACM, 54(3):12, 2007. (Preliminary Version in *38th STOC*, 2006). eccc:TR05-046, doi:10.1145/1236457.1236459.

[Din08] ———. *PCPs with small soundness error.* SIGACT News, 39(3):41–57, 2008. doi:10.1145/1412700.1412713.

[DR06] Irit Dinur and Omer Reingold. *Assignment testers: Towards a combinatorial proof of the PCP Theorem.* SIAM J. Computing, 36:975–1024, 2006. (Preliminary Version in *45th FOCS*, 2004). doi:10.1137/S0097539705446962.

[FGL+96] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. *Interactive proofs and the hardness of approximating cliques.* J. ACM, 43(2):268–292, March 1996. (Preliminary version in *32nd FOCS*, 1991). doi:10.1145/226643.226652.

[FK95] Uriel Feige and Joe Kilian. *Impossibility results for recycling random bits in two-prover proof systems.* In *Proc.* 27*th ACM Symp. on Theory of Computing (STOC)*, pages 457–468. ACM, 1995. doi:10.1145/225058.225183.

[For66] G. David Forney. *Concatenated Codes.* MIT Press, Cambridge, MA, USA, 1966.

[Hås01] Johan Håstad. *Some optimal inapproximability results.* J. ACM, 48(4):798–859, July 2001. (Preliminary Version in *29th STOC*, 1997). doi:10.1145/502090.502098.

[IKW09] Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. *Direct product testing: Improved and derandomized.* In *Proc.* 41*st ACM Symp. on Theory of Computing (STOC)*, pages 131–140. ACM, 2009. eccc:TR09-090, doi:10.1145/1536414.1536435.

[MR07] Dana Moshkovitz and Ran Raz. *Sub-constant error probabilistically checkable proof of almost linear size*, 2007. eccc:TR07-026.

[MR08a] ———. *Sub-constant error low degree test of almost-linear size.* SIAM J. Computing, 38(1):140–180, 2008. (Preliminary Version in *38th STOC*, 2006). eccc:TR05-086, doi:10.1137/060656838.

[MR08b] ———. *Two query PCP with sub-constant error.* In *Proc.* 49*th IEEE Symp. on Foundations of Comp. Science (FOCS)*, pages 314–323. IEEE, 2008. eccc:TR08-071, doi:10.1109/FOCS.2008.60.

[Raz98] Ran Raz. *A parallel repetition theorem.* SIAM J. Computing, 27(3):763–803, June 1998. (Preliminary Version in *27th STOC*, 1995). doi:10.1137/S0097539795280895.

[RS97] Ran Raz and Shmuel Safra. *A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP.* In *Proc.* 29*th ACM Symp. on Theory of Computing (STOC)*, pages 475–484. ACM, 1997. doi:10.1145/258533.258641.

[Sze99] Mario Szegedy. *Many-valued logics and holographic proofs.* In Jirí Wiedermann, Peter van Emde Boas, and Mogens Nielsen, eds., *Proc.* 26*th International Colloquium of Automata, Languages and Programming (ICALP)*, volume 1644 of *LNCS*, pages 676–686. Springer, 1999. doi:10.1007/3-540-48523-6_64.