Electronic Colloquium on Computational Complexity, Report No. 60 (2004)

Simple PCPs with Poly-log Rate and Query Complexity

Eli Ben-Sasson * Madhu Sudan †

July 22, 2004

Abstract

We give constructions of PCPs of length $n \cdot \text{poly}(\log n)$ (with respect to circuits of size n) that can be verified by making $\text{poly}(\log n)$ queries to bits of the proof. These PCPs are not only shorter than previous ones, but also simpler. Our (only) building blocks are Reed-Solomon codes and the bivariate low degree test of Polischuk and Spielman [27].

First, we present a new reduction from the verification of SAT to the following verification problem. Given oracle access to a function on a domain of size $n' = n \cdot \text{poly}(\log n)$, verify whether it is close to being an evaluation of a univariate polynomial of degree $n'/10$. While such reductions, from SAT-verification to verification of algebraic properties, have been extensively used in previous PCP constructions, our new reduction favors over them in its simplicity.

The reduction however does not seem to make the task of SAT-verification any easier. The degree of the polynomial to be tested is larger than the size of the original SAT problem. Thus, testing low degree of this string seems to cost more queries than required for reading the original satisfying assignment in its entirety! To overcome this, we present a short "PCP of Proximity" for certain Reed-Solomon codes. Specifically, we design a verifier that makes oracle access to the function (on the domain of size n') and an auxiliary "proof oracle" and accepts polynomials of degree $n'/10$ (when they are accompanied with the right auxiliary information) and rejects functions that are far from any polynomial of degree $n'/10$. The verifier makes only $\text{poly}(\log n')$ queries into two oracles (the function and the auxiliary proof).

Using these results over appropriately chosen fields translates our results into PCP verifiers whose query complexity is a poly-logarithmic number of bits. Our PCPs of proximity for Reed-Solomon codes also yields a new class of locally testable codes with poly-logarithmic rate and query complexity.

1 Introduction

Probabilistically Checkable Proofs [16, 3, 2] (a.k.a. Holographic Proofs [6]) are NP witnesses that allow efficient probabilistic verification based on probing few bits of the NP witness. The celebrated PCP Theorem [3, 2] asserts that probing a constant number of bits suffices, and it turned out that three bits suffice for rejecting false assertions with probability almost 1/2 (cf. [22, 20]). Although most famous for their applications to non-approximability results (see for example [10, 9, 22, 20, 30]),

*Radcliffe Institute for Advanced Study, Cambridge, MA 02139. Email: eli@eecs.harvard.edu.

†Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139. Email: madhu@mit.edu. Supported in part by NSF Award CCR-0312575. This work was done while at the Radcliffe Institute for Advanced Study.

ISSN 1433-8092

PCPs have several other uses in cryptography (e.g., CS-proofs [23, 26] and their applications [7, 13]) and coding theory [19, 12, 11].

Optimizing the length of the new NP witness was the focus of [6, 27, 21, 19, 12, 11], and in this work we continue this research direction. Our main result is a PCP construction that blows up the NP-witness length by a poly-logarithmic factor and can be verified by making a poly-logarithmic number of bit-size queries into the proof. An additional benefit of our constructions is their simplicity. Simplifying PCP constructions ought to be regarded an important problem in its own right, and we hope this paper makes a step in this direction (a combinatorial approach to simplify PCP constructions is given by [15]). The simplicity of our PCPs also means that the "hidden constants" in the construction seem to be relatively small and the few building blocks we use can be implemented with relative ease.

At the core of our new PCPs is a short "proof of proximity" (as defined by [11]) for the Reed-Solomon code (denoted the RS-code). This proof (of nearly linear size) allows one to test efficiently (with poly-logarithmic query complexity) whether a function is close to a low-degree (univariate) polynomial. Our new PCPPs are self-contained and rely only on the bivariate axis-parallel low degree test of [27].

Using the PCPPs for the RS-code our PCP construction becomes much simpler. We show a straightforward reduction of SAT to an algebraic constraint satisfaction problem, such that NP-witnesses for this problem correspond to a pair of Reed-Solomon codewords. Moreover, testing that such a pair corresponds to a good witness (implying satisfiability of the original SAT instance) becomes almost trivial as long as proximity to the RS-code can be guaranteed.

Our PCPPs can be extended to the multivariate case (Reed-Muller Codes) given the extensive literature on testing multivariate polynomials using axis parallel lines [5, 6, 16, 3, 27, 17]. In particular, we show an even simpler PCP construction for graph-3-colorability (albeit of quadratic length), based only on bivariate and uni-variate polynomials.

Related Results Optimizing the query complexity of PCPs has attracted a lot of attention, motivated in part by the significance of query complexity for non-approximability results (see, for example, [10, 9, 22, 20, 30]). However, these works only guarantee that the new NP witness (i.e., the PCP) is of length that is upper-bounded by a polynomial in the length of the original NP witness.[1] In some of these constructions the exponent of the resulting polynomial is too large to allow encoding of any circuit of size larger than one (!). (E.g. the exponent of [22] is $\approx 1,000,000$).

As mentioned above, the question of proof length was the focus of [6, 27, 21, 19, 12, 11] and a comparison of our result to that of the most recent such research [11] is illuminating. [11] gives a spectrum of PCP constructions. At one end the proof length incurs a quasi-poly-logarithmic blow-up and has query complexity $O(\log \log n)$ and at the other end the query complexity is constant, and the proof size is slightly larger (blow-up factor of $2^{\log^{\epsilon} n}$). Our proof length is shorter than both of these constructions, but its query complexity is larger. We stress that our constructions and analysis seem simpler than that of [11] (and our "hidden constants" are smaller).

[1] We stress that in all the above works as well as in the current work, the new NP witness can be computed in polynomial-time from the original NP witness.

2 Definitions and Main Results

Proofs of Proximity We measure distance between $x, y \in \Sigma^n$ using the normalized Hamming distance, i.e. $\Delta(x,y) \triangleq \Pr_{i\in[n]}[x_i \neq y_i]$. Let C be a subset of Σ^n. (Often, but not always, in this paper Σ will be a field and C a linear error correcting code.) The distance of x from C (denoted $\Delta(x,C)$) is the minimal distance between x and a member of C. We say that x is δ-far from C if $\Delta(x,C) > \delta$, and otherwise x is δ-close to C. Our next notion considers the task of proving efficiently verifiable proofs of the statement "$x \in C$", while making few queries into x and the proof. Such a verification task is necessarily probabilistic, and can only guarantee (upon acceptance) that x is δ-close to C. This notion was introduced in [11, Definition 2.3] as "Probabilistically Checkable Proofs of Proximity" (PCPP). Our definition below is a slight variant.[2]

Definition 1 (PCPP) *A set $C \subseteq \Sigma^n$ is said to have a* Probabilistically Checkable Proof of Proximity *(PCPP) over alphabet Σ of* length $\ell(n)$, *with* query complexity $q(n)$, randomness $r(n)$, *(perfect completeness) and* soundness $s(\cdot, n)$, *if there exists a polynomial time randomized* verifier *with oracle access to a pair $(x,\pi) \in \Sigma^{n+\ell(n)}$ such that,*

Operation *Verifier tosses $r(n)$ coins, makes $q(n)$ queries into (α,π) and outputs* accept *or* reject.

Perfect Completeness *If $x \in C$ then $\exists \pi \in \Sigma^{\ell(n)}$ such that verifier accepts (x,π) with probability 1.*

Soundness *If $\Delta(x,C) \geq \delta$ then for any $\pi \in \Sigma^{\ell(n)}$, the verifier rejects (x,π) with probability at least $s(\delta,n)$.*

PCPPs for the RS-Code Our first main result is that certain Reed-Solomon codes have proofs of proximity with $\ell(n) = \tilde{O}(n)$ and $q(n) = \text{poly}(\log n)$. For $P(z)$ a polynomial over a field $\mathbb{F}$ and $S \subseteq \mathbb{F}$ define its *evaluation table* over S to be $\langle P(z)\rangle_{z\leftarrow S} \triangleq \langle P(s) : s \in S\rangle$. The Reed-Solomon code of degree d over $\mathbb{F}$, evaluated at S is defined as

$$\text{RS}(\mathbb{F}, S, d) = \{\langle P(z)\rangle_{z\leftarrow S} : P(z) = \sum_{i=0}^{d-1} a_i z^i, a_i \in \mathbb{F}\}$$

The *fractional degree* of such a code is $d/|S|$. Let $\mathbb{F}^\star$ denote the cyclic multiplicative group of $\mathbb{F}$. Let the order of an element $\omega \in \mathbb{F}^\star$ be the smallest positive integer n such that $\omega^n = 1$. We refer to an integer n as a *power of two* if $n = 2^k$ for integer k. The multiplicative group generated by ω is $\langle\omega\rangle \triangleq \left\{\omega^0, \omega^1, \ldots, \omega^{n-1}\right\}$.

Theorem 1 (RS PCP of Proximity) *There exist universal constant $c \geq 1$ such that the following holds. Let $\omega \in \mathbb{F}^\star$ be an element of order n in the field $\mathbb{F}$, where n is a power of two and let $d < n$ be an integer. The Reed-Solomon code* $\text{RS}(\mathbb{F}, \langle\omega\rangle, d)$ *has a PCPP over alphabet $\mathbb{F}$ with the following parameters,*

Proof length $\ell(n) \leq n \log^c n$.

[2] Our definition is slightly stronger than that of [11]. We require the soundness be a function of the proximity, whereas [11] only needed the soundness to be large whenever the distance is large. Inspection reveals that the results of [11] also hold for the stronger definition of PCPPs.

Randomness $r(n) \leq \log n + c \log\log n$.

Query complexity $q(n) = O(1)$.

Soundness $s(\delta, n) \geq \delta / \log^c n$.

Examination of the proof of Theorem 1 shows it can be derived for any $\langle\omega\rangle$ of size n that is $\text{poly}(\log n)$-smooth, i.e. all prime factors of n are at most $\text{poly}(\log n)$. However, we don't need the stronger statement for our PCPs so opt for the simpler analysis of a 2-smooth n.

The soundness stated above is quite low (at best, it is inverse poly-log). However, randomness efficient repetition can boost it to a constant by mildly increasing the query complexity. In particular, we can test proximity to the RS-code of length n (with essentially the same proof length and randomness as stated above), rejecting with constant probability words that are $1/\text{poly}(\log n)$-far from the code and making $\text{poly}(\log n)$ queries into the codeword and its proof (see Section 3.7 for details).

Going from PCPs of proximity to locally testable codes is straightforward, using the techniques of [11, Section 4.1]. Thus we obtain *locally testable codes* (LTCs) with poly-logarithmic rate and query complexity (see Section 3.7 for definition of LTCs and more details).

Algebraic Constraint Satisfaction Problems In order to obtain length-efficient PCPs we reduce 3-SAT to an NP-complete *algebraic* constraint satisfaction problem. Whereas a witness for the satisfiability of a 3-CNF is a string of bits, an allowable witness for the algebraic problem is a low-degree (univariate) polynomial A, called the *witness polynomial*. A 3-CNF ϕ with n variables and m clauses can be viewed as a mapping $\phi : \{0,1\}^n \to \{0,1\}^m$, sending an assignment to the characteristic vector of the set of clauses satisfied by it. In our algebraic problem, an input A is mapped to a univariate *output* polynomial P of slightly larger degree. The mapping ϕ has the property that any output bit (which is an evaluation of a clause on the assignment) can be computed by making three queries into the input. Similarly, in our algebraic setting, the evaluation of the output polynomial P at x_0 can be computed by examining the witness polynomial A at the set of points $(\alpha_1 x_0, \ldots, \alpha_t x_0)$ where α_i is a constant and $t = O(\log n)$. Thus, to evaluate a P_i at a random x_0, requires making $O(\log n)$ random queries to A. Moreover, the computation of $P(x_0)$ is given by a polynomial of low degree in its inputs. Finally, a witness is *good* for a 3-CNF (i.e. testifies to its satisfiability) iff it is mapped to 1^m. In our algebraic problem, a witness polynomial A is *good* iff it is mapped to a polynomial P that evaluates to zero on a predefined set $H \subset \mathbb{F}$ that is independent of A. In the following definition and Theorem, we assume an infinite sequence of pairs $\{(\mathbb{F}_n, \omega_n) : \omega_n \in \mathbb{F}_n, |\langle\omega_n\rangle| > n\}_{n\in\mathbb{N}}$ is polynomial time computable (i.e., given n we can compute a description of $\mathbb{F}_n$ and ω_n).

Definition 2 (Univariate Algebraic CSP) *The language L_{UACSP} has as its space of instances, tuples of the form $(n, \alpha_1, \ldots, \alpha_t, C)$, where $\alpha_1, \ldots, \alpha_t \in \langle\omega_n\rangle$, $C : \mathbb{F}_n^{t+1} \to \mathbb{F}_n$ is a polynomial of degree at most n in its first variable and degree at most three in the remaining variables. An instance $(n, \alpha_1, \ldots, \alpha_t, C) \in L_{\text{UACSP}}$ if there exists a polynomial $A : \mathbb{F}_n \to \mathbb{F}_n$ of degree at most n such that for every $x \in \{\omega^0, \omega^1, \ldots, \omega^{n-1}\}$, $C(x, A(\alpha_1 x), \ldots, A(\alpha_t x)) = 0$.*

Theorem 2 *For every polynomial time computable sequence $\{(\mathbb{F}_n, \omega_n) : \omega_n \in \mathbb{F}_n, |\langle\omega_n\rangle| > n\}_{n\in\mathbb{N}}$, there exists a polynomial time reduction from 3-SAT to L_{UACSP} reducing 3-CNF formulas of length n to instances of the form $(n', \alpha_1, \ldots, \alpha_t, C)$, where $n' = O(n \log n)$ and $t = O(\log n)$.*

Notice the previous Theorem only requires a field with a large enough multiplicative group. But in order to obtain efficient PCPs we will also need to use proximity testers for RS-codes, thus putting further restrictions on the field structure.

Very similar algebraic reductions are prevalent in many previous PCPs [6, 3, 2, 12, 11], starting with [6]. All previous reductions used multivariate polynomials in order to perform degree reduction. Namely, a message (or assignment) of length n is encoded by an m-variate polynomial of degree $\approx m \cdot n^{1/m}$ (allowing proximity testing with $n^{1/m}$ queries). Our reduction does not reduce the degree at all, in fact it slightly increases it. The PCPPs for the RS code allow us to tolerate this and verify proximity to high-degree polynomials with very small query complexity (logarithmic in the degree).

Verifying Zeros of a Polynomial To complete our PCP constructions we will need to verify that a univariate function $f : S \to \mathbb{F}$ is close to a low degree polynomial that evaluates to zero on $H \subset \mathbb{F}$. This motivates the definition of the code $\mathrm{RS}_H(\mathbb{F}, S, d) \subseteq \mathrm{RS}(\mathbb{F}, S, d)$ which is the set of all RS-codewords that evaluate to zero on H. Formally,

$$\mathrm{RS}_H(\mathbb{F}, S, d) = \{\langle P(z)\rangle_{z \leftarrow S} : \deg(P) \leq d, \forall h \in H,\ P(h) = 0\}$$

Notice we do not require H to be a subset of S. It turns out that if $\mathrm{RS}(\mathbb{F}, S, d)$ has efficient PCPPs, so does $\mathrm{RS}_H(\mathbb{F}, S, d)$.

Lemma 3 *Suppose* $\mathrm{RS}(\mathbb{F}, S, d)$ *has a PCP of proximity with length* ℓ*, query complexity* q*, randomness* r *and soundness* $s(\delta)$*. Then for any* $H \subset \mathbb{F}$*,* $\mathrm{RS}_H(\mathbb{F}, S, d)$ *has a PCP of proximity with length* $|S| + 2\ell$*, query complexity* $2(q+1)$*, soundness* $\geq \min\{s(\delta), 1 - (2\delta + d/|S|)\}$ *and randomness* $\max\{r, \log|S|\}$.

Nearly-Linear PCPs Theorems 1, 2 and Lemma 3 give nearly-linear length PCPs. Indeed, given a 3-CNF formula of size n, our verifier reduces it to an instance of L_{UACSP} via Theorem 2. The field $\mathbb{F}_n$ used in the reduction will have a root of unity ω of order $O(n \log n)$ that is a power of two.[3] As a proof, the verifier expects the evaluation of the polynomials $A(x)$ and $P(x) \triangleq C(x, A(\alpha_1 x), \ldots, A(\alpha_t x))$ over $\langle\omega\rangle$. The first evaluation table is accompanied by a PCP of proximity to the Reed-Solomon code of degree n', and the second evaluation is accompanied by a proof of proximity to RS_H, where $H = \{\omega^0, \ldots, \omega^{n'}\}$. The verifier tests proximity of each polynomial to the proper code, using $\mathrm{poly}(\log n)$ queries to be able to reject with probability $1/2$ any string that is $1/O(\log n)$-far from the proper code. If both proximity tests accept, verifier checks consistency by picking a random $\beta \in \langle\omega\rangle$ and accepting iff $P(\beta) = C(\beta, A(\alpha_1\beta), \ldots, A(\alpha_t\beta))$. This construction gives our main Theorem (the formal proof appears in Section 5).

Theorem 4 (Efficient PCPs) *SAT is in* $PCP_{1,\frac{1}{2}}[\log_2(n\,\mathrm{poly}\log n), \mathrm{poly}\log n]$*. In other words, SAT has a PCP verifier that on inputs of length* n *tosses* $\log_2(n\,\mathrm{poly}\log n)$ *coins, makes* $\mathrm{poly}(\log n)$ *queries to a proof oracle of length* $n\,\mathrm{poly}(\log n)$ *and has perfect completeness and soundness at most* $\frac{1}{2}$.

This ends the survey of our main new results. However, our techniques can also be used to simplify the sum-check in previous PCP constructions, as discussed below.

[3] Such fields can be found in deterministic polynomial time, by Linnik's Theorem 18. See Section 5 for more details.

Multivariate Zero Testing and PCPs We point out a generalization of Lemma 3 to multivariate polynomials that can replace the sum-check based protocols in previous PCP constructions [6, 3, 2, 27, 21, 19, 12, 11]. In the multivariate problem we are given sets $S, H \subset \mathbb{F}$ and oracle access to a multivariate function $f : S^m \to \mathbb{F}$. We are asked to verify f is close to a polynomial of degree $\leq d$ in each variable that evaluates to zero on H^m (once again, we do not need to assume $H \subset S$). We denote by $\mathrm{RM}(\mathbb{F}, S, d, m)$ the m-variate Reed-Muller code of individual degree d, evaluated over S^m and by $\mathrm{RM}_H(\mathbb{F}, S, d, m)$ its sub-code consisting of all (evaluations of) polynomials that vanish on H^m.

Lemma 5 *Suppose* $\mathrm{RM}(\mathbb{F}, S, d, m)$ *has a PCPP with length* ℓ*, query complexity* q*, randomness* r *and soundness* $s(\delta)$*. Then for any* $H \subset \mathbb{F}$*,* $\mathrm{RM}_H(\mathbb{F}, S, d, m)$ *has a PCPP with length* $m \cdot |S|^m + (m+1)\ell$*, query complexity* $(m+1)(q+1)$*, soundness* $\geq \min\{s, 1 - ((m+1)\delta + \left(\frac{d}{|S|}\right)^m)\}$ *and randomness* $\max\{r, m \log |S|\}$.

Notice that the query complexity of previous solutions to this problem depended also on the size of H. Our simpler solution has query complexity that depends only on m and is based on a straightforward characterization of RM_H (similar to Alon's Combinatorial Nullstellensatz [1]).

To illustrate the power of Lemma 5, we apply it to a bivariate algebraic constraint satisfaction problem arising from an algebraic version of the graph-3-colorability problem. This leads to PCPs with $\mathrm{poly}(\log n)$ queries and proof length $n^2 \cdot \mathrm{poly}(\log n)$ (Theorem 24). Although this PCP does not obtain the nearly linear length of Theorem 4, its simplicity can de-mystify the magic of PCP Theorems.

Paper Organization Section 3 presents PCPs of Proximity for the Reed-Solomon code, proving Theorem 1. Section 4 gives the reduction of 3-SAT to Algebraic Constraint Satisfaction problem, leading to Theorem 2. The proof of the efficient PCP Theorem 4 appears in Section 5, followed by the simpler (and longer) bivariate based PCP (Section 6). We conclude with a brief discussion of implementation issues (Section 7).

3 Short PCPs of Proximity for Reed-Solomon Codes

In this section we give a PCP of Proximity for Reed-Solomon codes. The PCPP is essentially built from first principles: At a high level, we attempt an elementary reduction from the task of testing a univariate polynomial to the task of testing (a few) bivariate polynomials of significantly smaller degree. We then invoke an analysis of a "bivariate low-degree test" by Polishchuk and Spielman [27], which reduces the task of testing bivariate polynomials back to the task of testing univariate polynomials, of much smaller degree than the original. Recursing on this idea leads to the full test. The main catch in this outline is that the reduction reduces the original testing task to a stronger task of testing a few polynomials and verifying some *consistency* between them. To capture this consistency testing, we define a stronger problem and show that our reduction reduces this problem to itself with smaller parameters. (A related self-reduction in the context of multivariate low-degree testing appears in [14].)

In Section 3.1 we elaborate on our approach and introduce the stronger testing problem informally. In Section 3.2 we formally introduce the Shifted Reed-Solomon (SRS) code and describe our testing

result for this code. In Section 3.3 we describe the tester for the SRS code. In Section 3.4 we analyze the simple properties of this tester (query complexity, randomness, completeness etc.). In Section 3.5 we analyze the soundness of this tester.

3.1 Introducing the Shifted Reed-Solomon Code

We start by considering a polynomial $P(z) = \sum_{i=0}^{n/2-1} a_i z^i$, evaluated at some set $W \subseteq \mathbb{F}$ of cardinality n, and consider the task of "testing" it. Our main idea is that we can define the bivariate polynomial $Q(x,y) = \sum_{j=0}^{\sqrt{n}/2-1} \sum_{k=0}^{\sqrt{n}-1} a_{j\cdot\sqrt{n}+k} y^j x^k$, and this polynomial has degree only $\sqrt{n}$, while "capturing" all the information of P. (Specifically, we can reconstruct P from Q using the identity $P(z) = Q(z, z^{\sqrt{n}})$.) Furthermore, testing of bivariate polynomials reduces to testing of univariate polynomials of roughly the same degree using well-known "low-degree tests" and their analysis (cf. [27]). This leads us to the hope that the polynomial Q might provide, or at least lead to, a good "proof" that P is of low-degree. Specifically, to prove that a table of evaluations of P corresponds to the evaluations of a polynomial of low-degree, the prover can provide a table of evaluations of a bivariate polynomial Q, prove that Q has degree $\sqrt{n}$ in each variable, and then prove that Q is consistent with the table of evaluations of P.

To completely describe the above approach, all we need to do is describe which set of points we will specify Q on, so as to achieve both tasks: (1) verifying that Q has low-degree, and (2) that it is consistent with P. However this part leads to conflicting goals. In order to test that Q has low-degree, using a bivariate tester, we need to know its values on some subset $X \times Y$ where $X, Y \subseteq \mathbb{F}$. To make this efficient, we need to make $|X|, |Y| \approx \sqrt{n}$. On the other hand to test its consistency with P, the natural approach is to ask for its values on the set $Z = \{(z, z^{\sqrt{n}}) | z \in W\}$. Unfortunately the set Z is far from being of the form $X \times Y$. (For starters, the projection of Z onto its first coordinate has cardinality n while we would like this projection to be of cardinality $O(\sqrt{n})$.)

This discrepancy (between Z and cross-product sets) seems to kill this approach entirely, however it turns out it can be salvaged. To do so, we choose W to be special, and let $W = \langle\omega\rangle$ for some element ω of order n in $\mathbb{F}$. We also assume n is a square, so that $\sqrt{n}$ is an integer. Figure 1 plots Z on $\mathbb{F}^* \times \mathbb{F}^*$, where the elements of $\mathbb{F}^*$ are enumerated as powers of a generator of the multiplicative group. While Z does not form a product set, it does have some nice features. The elements of Z lie on a "lattice" in this representation and it turns out this can be exploited to our advantage.[4]

The first observation we make about the set Z in Figure 1 is that its points lie on $\sqrt{n}$ horizontal lines, with y-coordinates given by the set $Y = \{\omega^{j\sqrt{n}} | j \in \{0, \ldots, \sqrt{n}-1\} = \langle\omega^{\sqrt{n}}\rangle$. This motivates us to consider (as supporting a proof that P is a low-degree polynomial) the restriction of Q onto $Y \times Y$ for $Y = \langle\omega^{\sqrt{n}}\rangle$.

A minor hitch with this suggestion is that $Q(x, y_0)$ for $y_0 \in Y$ is a degree $\sqrt{n}$ polynomial in x and its value is only given at $\sqrt{n}$ places on this horizontal line. To get around this degree problem, we reduce the x-degree of Q by expressing it as a sum of two polynomials. Specifically, let $Q^{(0)}(x,y) = \sum_{j=0}^{\sqrt{n}/2-1} \sum_{k=0}^{\sqrt{n}/2-1} a_{j\cdot\sqrt{n}+k} y^j x^k$, and let $Q^{(1)}(x,y) = \sum_{j=0}^{\sqrt{n}/2-1} \sum_{k=0}^{\sqrt{n}/2-1} a_{j\cdot\sqrt{n}+k+\sqrt{n}/2} y^j x^k$. We get $Q^{(0)}, Q^{(1)}$ have degree less than $\sqrt{n}/2$ in each variable and $Q(x,y) = Q^{(0)}(x,y) + x^{\sqrt{n}/2} Q^{(1)}(x,y)$. We now amend our proposal and suggest that the restriction of $Q^{(0)}$ and $Q^{(1)}$ to $X \times Y$, for

[4] We stress that Q restricted to a (non-axis-parallel) line in this layout is not a polynomial of degree $O(\sqrt{n})$, since this grid is an "exponential" one.

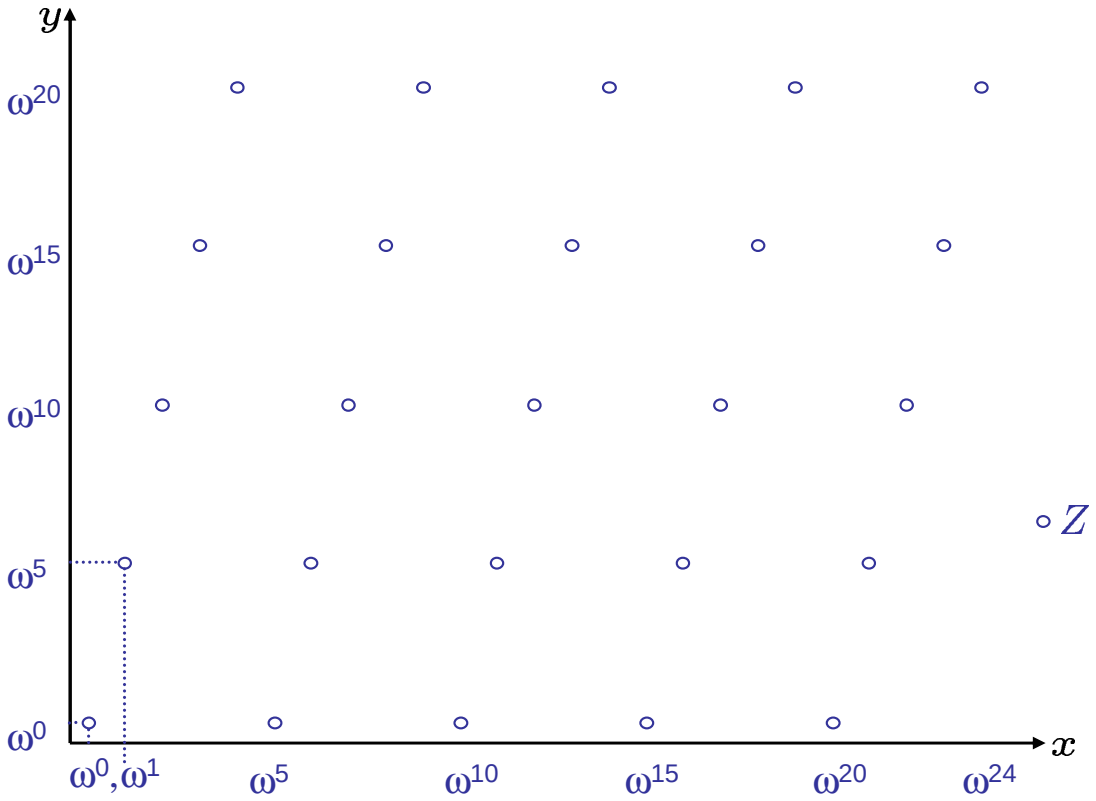

Figure 1: In this case, $\mathbb{F} = \mathbb{Z}_{101}$. Let σ generate $\mathbb{F}^\star$ and $\omega = \sigma^4$ be an element of order $n = 25$. The elements on each axis are ordered by increasing powers of σ and the figure shows the set of points $Z = \{(\omega^i, \omega^{i\cdot\sqrt{n}})|i\} \subset \mathbb{F}^\star \times \mathbb{F}^\star$.

$X = Y = \langle \omega^{\sqrt{n}} \rangle$, as well as their values on Z, can be used to prove that P is a low-degree polynomial.

To verify such a proof, a verifier picks a random choice of $\alpha \in X$ and verifies that $Q^{(0)}$, as specified by its values on $X \times Y$, restricted to the line parallel to the second axis (with the first argument set to α) is a polynomial of degree at most $\sqrt{n}/2 - 1$ in the second variable. Similarly it tests $Q^{(0)}$ for a random setting of its second argument, and does similar tests for $Q^{(1)}$, restricted to $X \times Y$. Finally it tests, on a random point of $(x, y) \in Z$, that $P(x) = Q^{(0)}(x, y) + x^{\sqrt{n}/2} Q^{(1)}(x, y)$. The key ingredient missing in all the above is the consistency of $Q^{(0)}$ (and $Q^{(1)}$) on $X \times Y$ with its evaluations on Z. Restricted to a random line parallel to the first axis with second argument set to β, this reduces to the task of checking the consistency of $Q^{(0)}(\cdot, \beta)$ on the points in $S_1 = \{\omega^{j\sqrt{n}}\}_j$ with the points of the form $S_2 = \{x \in W | x^{\sqrt{n}} = \beta\}$. Thus, our consistency problem can be abstracted as the task of testing if two tables, giving the value of two functions at S_1 and S_2, correspond to the evaluations of the *same* univariate polynomial at sets S_1 and S_2. We don't know how to solve this problem for general S_1 and S_2, but in our case the sets S_1 and S_2 have a special form (see Figure 2). The set S_1 is of the form $\langle \omega' \rangle$ where $\omega' = \omega^{\sqrt{n}}$. And the lattice picture clarifies that the set S_2 is also well-behaved: It is of the form $\{\kappa \cdot \omega^{j\sqrt{n}}\}_j = \kappa \cdot S_1$, where $\kappa \in W$ is such that $\kappa^{\sqrt{n}} = \beta$.

This motivates us to define the Shifted Reed Solomon (SRS) code, whose codewords are the evaluations of a polynomial P at a cyclic subgroup of a field, and its coset (shifted by κ). We investigate

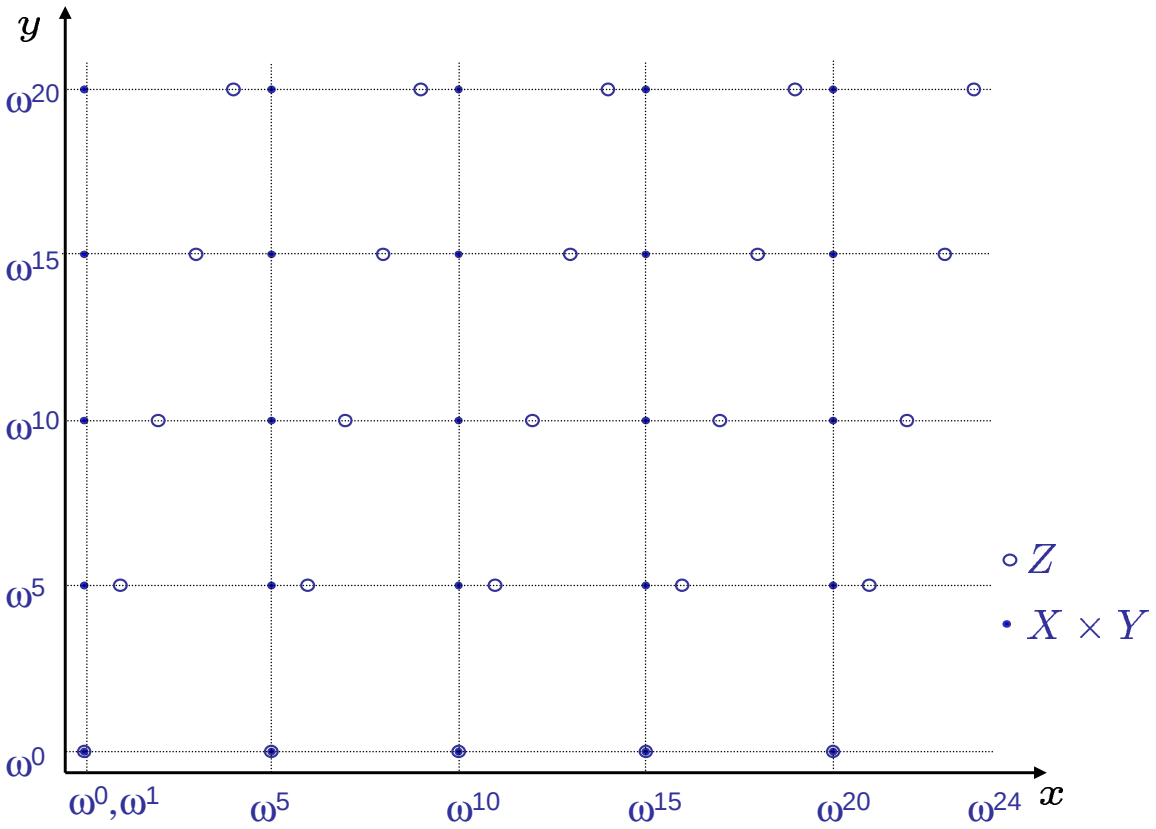

Figure 2: Z and $X \times Y$ plotted on the same grid (as in Figure 1. Note the points lie on five ($\sqrt{n}$) rows. On each row the points correspond to the evaluations of a degree $\sqrt{n}/2$ polynomial evaluated on the fifth roots of unity and their multiplicative shifts, where the shift is fixed for a given row.

the testing of this code, along the same ideas as we proposed for testing the Reed-Solomon code above, and fortunately for us, things fall into place. We can define bivariate polynomials Q, $Q^{(0)}$, $Q^{(1)}$ as above. The evaluations of P on the sets $W = \langle\omega\rangle$ and $W_\kappa = \kappa \cdot \langle\omega\rangle$, translate to the evaluations of Q on Z and $Z_\kappa = \{(\kappa\omega^i, \kappa^{\sqrt{n}}\omega^{i\sqrt{n}}) | i \in \{0, \ldots, n-1\}\}$. Figure 3 shows Z and Z_κ pictorially. To ease the testing we add the values of $Q^{(i)}$'s on $X \times Y$ and $X \times (Y_\kappa)$, where $Y_\kappa = \{\kappa \cdot y | y \in Y\}$. (See Figure 4.) The testing of these tables and their consistency, reduces to the testing of consistency of $Q^{(i)}$'s on three sets of lines: (1) on vertical lines with first coordinate being a point of X (for $X \times Y$ vs. $X \times Y_\kappa$), (2) on horizontal lines through points of Y (for $X \times Y$ vs. Z) and (3) on horizontal lines through points of Y_κ (for $X \times Y_\kappa$ vs. Z_κ). Additionally, we need to test that the $Q^{(i)}$'s, on their values at Z and Z_κ are consistent with the tables of values of P on W and its coset given by $\kappa \cdot W$. The last tests can be carried out with constant query complexity, while the tests (1)-(3) above can be solved recursively. This leads to the verifier described in the next section.

From Intuition to Proof Our rigorous analysis will follow the intuition laid so far, with two technical differences. So far we assumed $\sqrt{n}$ to be an integer. Applying the same assumption recursively implies $n = 2^{2^k}$ (for some integer k), and the set of such integers is very sparse. But for our purposes, we only need to factor n into n_0, n_1 such that each is $\approx \sqrt{n}$. We will use $n = 2^k, n_0 = 2^{\lceil k/2 \rceil}, n_1 = 2^{\lfloor k/2 \rfloor}$ and notice the construction still follows. In fact, for obtaining

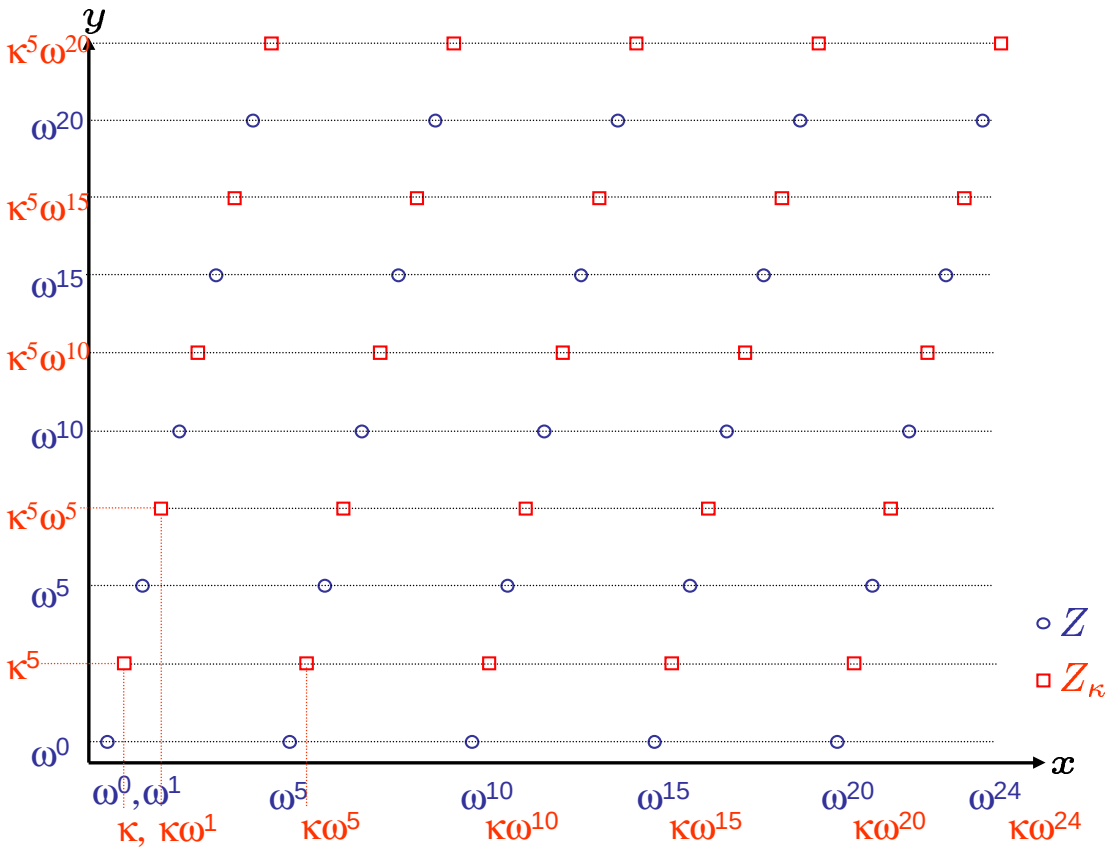

Figure 3: In this example $\kappa = \sigma^2$ (recall σ generates $\mathbb{F}^\star$ and $\omega = \sigma^4$).

short PCPs a polylog(n)-*smooth*[5] n would suffice. Fortunately, Linnik's Theorem (see Theorem 18 and the references therein) assures us of the abundance of small prime fields with a multiplicative sub-group of size 2^k for any k, so we only deal with these codes.

The second technical difference is that in what follows we will consider polynomials of degree at most n/c (evaluated over a domain of size n), for some large constant c as opposed to 2. This is because the bivariate low-degree testing theorem of [27] requires the ratio of the degree to the domain to be this small. (See statement of Theorem 25 for the exact constraints.) This fractional degree forces us to break our polynomial $P(z)$ into *eight*[6] bivariate polynomials $\{Q^{(\ell)}\}_{\ell \in \{0,\ldots,7\}}$ (rather than two as discussed above) of degree $\approx \sqrt{n}/8$ in each variable.

Finally, a few words explaining the notation: In what follows we will describe an SRS verifier that will access several oracles which are supposed to correspond to some of the functions described above (though it will be the verifier's task to verify this correspondence). In particular, we use oracles p, p_κ to describe the two parts of an oracle describing a supposed SRS codeword. We use oracles $f^{(\ell)}$ and $g^{(\ell)}$ to describe the restriction of the function $Q^{(\ell)}$ to Z, and $X \times Y$. Elements of X and Y are described in the natural way, i.e., $X = \{\alpha^{n_1} | \alpha \in \langle\omega\rangle\}$ and $Y = \{\beta^{n_0} | \beta \in \langle\omega\rangle\}$. We use a slightly unnatural (but eventually convenient) way to describe elements of Z: we describe the set by pairs $\{(\tilde{\alpha}, \beta) | \tilde{\alpha} \in X, \beta \in \langle\omega\rangle_{n_1}\}$, where $\langle\omega\rangle_{n_1} = \{1, \omega, \omega^2, \ldots, \omega^{n_1 - 1}\}$. Notice that the elements

[5] n is said to be k-smooth if all its prime factor are $\leq k$.

[6] Inspection of [27] shows we can use fractional degree that is larger than 1/8. However, for the sake of simplicity of exposition we do not optimize our construction thus.

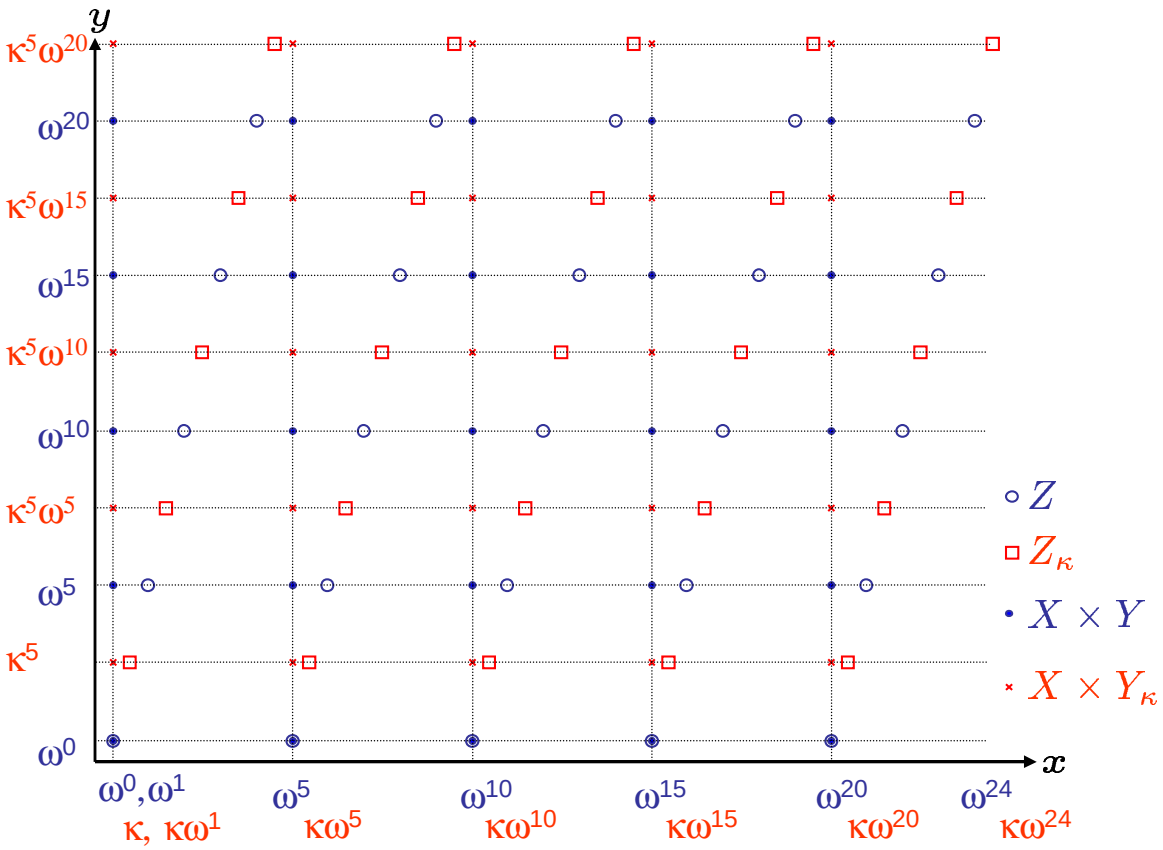

Figure 4: Z and $X \times Y$ share the same rows. So do Z_κ and $X \times Y_\kappa$. Finally, $X \times Y$ and $X \times Y_\kappa$ share columns. Thus, SRS-proximity of size n is reduced to SRS-proximity of size $\sqrt{n}$.

$\tilde{\alpha} \cdot \beta$ are in one to one correspondence with the first coordinates of Z. We use oracles $f_\kappa^{(\ell)}$ and $g_\kappa^{(\ell)}$ to describe the restriction of $Q^{(\ell)}$ to Z_κ and $X \times Y_\kappa$, however we shift their inputs appropriately (by some function of κ) so that they are described also as functions on Z (represented as above) and $X \times Y$. The tests then correspond to verifying consistency of the $f^{(\ell)}$'s and the $f_\kappa^{(\ell)}$'s with p and p_κ, as well as recursive tests to check SRS consistency of $f^{(\ell)}$ vs. $g^{(\ell)}$ on horizontal lines, of $g^{(\ell)}$ vs. $g_\kappa^{(\ell)}$ on vertical lines, and of $g_\kappa^{(\ell)}$ vs. $f_\kappa^{(\ell)}$ on horizontal lines.

3.2 Definitions and Main result

We now formally define the Shifted Reed Solomon (SRS) code, which was introduced informally in the last section.

Definition 3 (Shifted Reed Solomon Code) *For ω of order n in $\mathbb{F}$, integer $d < n$ and $\kappa \in \mathbb{F}^\star$, the degree d* shifted Reed Solomon code (SRS code) *over $\langle\omega\rangle$ with* shift *κ, denoted* $\mathrm{SRS}(\mathbb{F}, d, \omega, \kappa)$, *is the map $\mathcal{C} : \mathbb{F}^d \to \mathbb{F}^{2n}$ given by*

$$\mathcal{C}(a_0, \ldots, a_{d-1}) = \langle P(z)\rangle_{z \leftarrow \langle\omega\rangle}, \langle P(\kappa z)\rangle_{z \leftarrow \langle\omega\rangle}, \qquad \text{where } P(z) \triangleq \sum_{i=0}^{d-1} a_i z^i$$

If $\kappa \in \langle\omega\rangle$ then $\langle P(\kappa z)\rangle_{z\leftarrow\langle\omega\rangle}$ is merely a permutation of $\langle P(z)\rangle_{z\leftarrow\langle\omega\rangle}$. But when $\kappa \notin \langle\omega\rangle$, the more interesting case, $\langle P(\kappa z)\rangle_{z\leftarrow\langle\omega\rangle}$ is an evaluation of $P(z)$ on a coset of $\langle\omega\rangle$ within $\mathbb{F}^\star$.

The shifted Reed Solomon code (SRS-code) is a generalization of the standard RS-code. In particular, taking $\kappa = 1$ gives the RS-code (repeated twice), so the distance and rate of the SRS-code match that of the RS-code up to a multiplicative constant of two. Thus, from here on we will focus on presenting a PCP of proximity for the SRS code, noticing Theorem 1 is a special case of the following theorem, the proof of which occupies the rest of the section.

Theorem 6 (SRS PCP of Proximity) *There exists a universal constant $c \geq 1$ such that the following holds. Let ω be an element of order n in the field $\mathbb{F}$, where n is a power of two, let $\kappa \in \mathbb{F}^\star$ and $d < n$. Then the corresponding SRS-code has a PCPP over alphabet $\mathbb{F}$ with the following parameters,*

Proof length $\ell(n) \leq n \log^c n$,

Randomness $r(n) \leq \log n + c \log\log n$,

Query complexity $q(n) = O(1)$, *and*

Soundness $s(\delta, n) \geq \delta / \log^c n$,

where the proof length and query complexity specify the number of elements of $\mathbb{F}$ written (or read).

Our proof will focus on the special case of degree $d = n/8$. Generalizing to arbitrary degree is deferred to Section 3.7.

3.3 The SRS Proximity Tester

In this section we describe the SRS Proximity tester. We start by recalling some notation. The order of an element $\alpha \in \mathbb{F}$, denoted $\mathrm{ord}(\alpha)$, is the smallest positive integer i such that $\alpha^i = 1$. Recall that we have an element ω of order n, where $n = 2^k$ for some integer k. Let $n_0 = 2^{\lceil k/2 \rceil}$ and $n_1 = 2^{\lfloor k/2 \rfloor}$. Note that $n = n_0 \cdot n_1$ and $\sqrt{n/2} \leq n_1 \leq n_0 \leq \sqrt{2n}$. Throughout $\langle\alpha\rangle$ denotes the set $\{\alpha^0, \alpha^1, \ldots, \alpha^{\mathrm{ord}(\alpha)-1}$. For $\ell \leq \mathrm{ord}(\alpha)$, we use $\langle\alpha\rangle_\ell$ to denote the set $\{\alpha^0, \alpha^1, \ldots, \alpha^{\ell-1}\}$.

We are now ready to describe the inputs to the SRS Proximity tester. Its explicit inputs are the field $\mathbb{F}$, the degree parameter d (set to $n/8$ in our case), the shift κ and the element ω of order n. The more important inputs are the implicit ones, or the oracles accessed by the SRS Proximity Tester. As expected it accesses two "input" oracles denoted (p, p_κ), where $p, p_\kappa : \langle\omega\rangle \to \mathbb{F}$. Additionally it accesses a proof oracle π. If $n \leq 16$, the proof oracle is empty, else it is of the form $\pi = (\{f^{(\ell)}, f_\kappa^{(\ell)}, g^{(\ell)}, g_\kappa^{(\ell)}, \{\pi^{(1,\beta,\ell)}, \pi^{(2,\beta,\ell)}\}_{\beta\in\langle\omega\rangle_{n_1}}, \{\pi^{(3,\tilde{\alpha},\ell)}\}_{\tilde{\alpha}\in\langle\omega^{n_1}\rangle}\}_{\ell\in\{0,\ldots,7\}})$, where $f^{(\ell)}, f_\kappa^{(\ell)} : \langle\omega^{n_1}\rangle \times \langle\omega\rangle_{n_1} \to \mathbb{F}$, $g^{(\ell)}, g_\kappa^{(\ell)} : \langle\omega^{n_1}\rangle \times \langle\omega^{n_0}\rangle \to \mathbb{F}$, and the $\pi^{(\cdot,\cdot,\ell)}$'s are proof oracles as needed by recursive calls to SRS Proximity testers. (Informally, the four functions $f^{(\ell)}, f_\kappa^{(\ell)}, g^{(\ell)}, g_\kappa^{(\ell)}$ correspond in Figure 4 to the evaluation of the bivariate polynomial $Q^{(\ell)}(x, y)$ on the four sets of points: $Z, Z_\kappa, X \times Y, X \times Y_\kappa$, respectively.)

Finally, before describing the Proximity tester we need some new notation. For general sets A, B, C, a "bivariate" function $h : A \times B \to C$, and elements $\alpha \in A$ and $\beta \in B$ we denote the β-row of h

by $h|_\beta^{\leftrightarrow} : A \to C$ and define it to be the function $h|_\beta^{\leftrightarrow}(\alpha') = h(\alpha', \beta)$. Similarly, the α-column is denoted $h|_\alpha^{\updownarrow} : B \to C$ and defined as $h|_\alpha^{\updownarrow}(\beta') = h(\alpha, \beta')$. We are now ready to describe the proximity tester.

Definition 4 (Verifier for SRS-Code) *The verifier for proximity to* $\mathrm{SRS}(\mathbb{F}, d \stackrel{\text{def}}{=} n/8, \omega, \kappa)$ *receives as input the parameters* $\mathbb{F}, \omega, \kappa$ *as defined in the statement of Theorem 6. It has oracle access to a purported codeword* (p, p_κ) *and its purported proof*

$$\pi = \left\{ f^{(\ell)}, f_\kappa^{(\ell)}, g^{(\ell)}, g_\kappa^{(\ell)}, \left\{ \pi^{(1,\beta,\ell)}, \pi^{(2,\beta,\ell)} \right\}_{\beta \in \langle\omega\rangle_{n_1}}, \left\{ \pi^{(3,\tilde{\alpha},\ell)} \right\}_{\tilde{\alpha} \in \langle\omega^{n_1}\rangle} \right\}_{\ell \in \{0,\ldots,7\}},$$

and is denoted $\mathrm{V}_{\mathrm{SRS}}^{\langle (p,p_\kappa),\pi\rangle}(\mathbb{F}, n, \omega, \kappa)$. *If* $n \leq 16$ *(in which case* $\pi = \emptyset$*), the verifier reads* p *and* p_κ *in entirety and accepts iff* $(p, p_\kappa) \in \mathrm{SRS}(\mathbb{F}, 2, \omega, \kappa)$. *Otherwise, it computes* $n_0 = 2^{\lceil k/2 \rceil}, n_1 = 2^{\lfloor k/2 \rfloor}$ *(recall* $n = 2^k$*) and performs one of the following four tests with probability* $1/4$ *each.*

Outer: *Pick* $\tilde{\alpha} \in \langle\omega^{n_1}\rangle, \beta \in \langle\omega\rangle_{n_1}$ *uniformly at random; Query* $p(\tilde{\alpha} \cdot \beta)$, $p_\kappa(\tilde{\alpha} \cdot \beta)$ *and* $f^{(\ell)}(\tilde{\alpha}, \beta)$, $f_\kappa^{(\ell)}(\tilde{\alpha}, \beta)$ *for every* $\ell \in \{0, \ldots, 7\}$*; Accept iff* $p(\tilde{\alpha} \cdot \beta) = \sum_{\ell=0}^{7} (\tilde{\alpha} \cdot \beta)^{\ell n_0/8} \cdot f^{(\ell)}(\tilde{\alpha}, \beta)$ *and* $p_\kappa(\kappa\tilde{\alpha} \cdot \beta) = \sum_{\ell=0}^{7} (\kappa\tilde{\alpha} \cdot \beta)^{\ell n_0/8} \cdot f_\kappa^{(\ell)}(\tilde{\alpha}, \beta)$.

Inner: *Pick* $\ell \in \{0, \ldots, 7\}, \beta \in \langle\omega\rangle_{n_1}$ *at random and run* $\mathrm{V}_{\mathrm{SRS}}^{\langle g^{(\ell)}|_{\beta^{n_0}}^{\leftrightarrow}, f^{(\ell)}|_\beta^{\leftrightarrow}, \pi^{(1,\beta,\ell)}\rangle}(\mathbb{F}, n_0, \omega^{n_1}, \beta)$.

Inner$_\kappa$: *Pick* $\ell \in \{0, \ldots, 7\}, \beta \in \langle\omega\rangle_{n_1}$ *at random and run* $\mathrm{V}_{\mathrm{SRS}}^{\langle g_\kappa^{(\ell)}|_{\beta^{n_0}}^{\leftrightarrow}, f_\kappa^{(\ell)}|_\beta^{\leftrightarrow}, \pi^{(2,\beta,\ell)}\rangle}(\mathbb{F}, n_0, \omega^{n_1}, \kappa\beta)$.

Inner$_c$: *Pick* $\ell \in \{0, \ldots, 7\}, \tilde{\alpha} \in \langle\omega^{n_1}\rangle$ *at random and run* $\mathrm{V}_{\mathrm{SRS}}^{\langle g^{(\ell)}|_{\tilde{\alpha}}^{\updownarrow}, g_\kappa^{(\ell)}|_{\tilde{\alpha}}^{\updownarrow}, \pi^{(3,\alpha,\ell)}\rangle}(\mathbb{F}, n_1, \omega^{n_0}, \kappa^{n_0})$.

The remaining subsections analyze the performance of this verifier, thus yielding Theorem 1. Specifically, the next subsection analyzes the simple properties including the query complexity, the randomness/size complexity, and the completeness. The hard part, the soundness analysis is addressed in Section 3.5. Throughout this analysis we assume degree $d = n/8$, and Section 3.6 generalizes this to arbitrary degree, thus proving Theorems 6 and 1. We conclude with some corollaries in Section 3.7.

3.4 Basic properties

Proposition 7 $\mathrm{V}_{\mathrm{SRS}}^{\langle (p,p_\kappa),\pi\rangle}(\mathbb{F}, n, \omega, \kappa)$ *makes at most* 32 *queries into* p, p_κ, π. *It tosses at most* $\log_2 n + O(\log\log n)$ *random coins. The size of the oracle* π *accessed by* $\mathrm{V}_{\mathrm{SRS}}^{\langle (p,p_\kappa),\pi\rangle}(\mathbb{F}, n, \omega, \kappa)$ *is* $O(n \cdot \mathrm{poly}\log n)$.

Proof: The proof is straightforward from the definition. The query complexity is easy to verify. In the base case, the verifier reads 32 field elements. In the inductive case, if the verifier chooses to execute the **Outer** step, then it makes $18 < 32$ queries, else it makes a recursive query to $\mathrm{V}_{\mathrm{SRS}}^{\langle\rangle}$ which makes 32 queries by induction.

The randomness complexity is similar. In the base case the verifier tosses 0 coins. In the inductive case, the verifier tosses $O(1)$ coins to determine which step to perform. If it chooses the outer

test, it picks $\tilde{\alpha}$ and β at random with $\log n + O(1)$ coins. If it chooses one of the inner tests, it tosses $\log\sqrt{n} + O(1)$ coins to determine the inner call, and then $\log\sqrt{n} + O(\log\log\sqrt{n})$ coins in the recursive call. Adding up, we get a total of $\log n + O(\log\log n)$ coins in all.

The size of the oracle can be similarly analyzed (or bounded by $2^{\text{randomness}}$ to get the same bound). ∎

Next we move to the completeness part of the proof. This part is straightforward given the intuition developed in Section 3.1. We first abstract the notion of expressing a univariate polynomial by bivariate polynomials of low degree in the following proposition. We then use this as described in Section 3.1 to describe a proof π that is accepted with probability one by $\mathrm{V}_{\mathrm{SRS}}^{\langle(p,p_\kappa),\pi\rangle}(\mathbb{F}, n, \omega, \kappa, \delta)$ when the input oracles correspond to an SRS codeword.

Proposition 8 *Given positive integers d_1, d_2, L, and d such that $d_1 \cdot d_2 \cdot L \geq d$, the following holds: For every univariate polynomial $P(x)$ of degree less than d there exists a sequence of L bivariate polynomial $Q^{(0)}(y,z), \ldots, Q^{(L-1)}(y,z)$, of degree less than d_1 in y and d_2 in z such that $P(x) = \sum_{\ell=0}^{L-1} x^{\ell \cdot d_1} Q^{(\ell)}(x, x^{L \cdot d_1})$. Furthermore, such a sequence is unique if $d_1 \cdot d_2 \cdot L = d$.*

Proof: Let a_i's be the coefficients of P, i.e., $P(x) = \sum_{i=0}^{d-1} a_i x^i$. Now let

$$Q^{(\ell)}(y,z) = \sum_{i=0}^{d_1-1} \sum_{j=0}^{d_2-1} a_{i+\ell \cdot d_1 + j \cdot d_1 \cdot L} y^i z^j,$$

where a_i is defined to be 0 if $i \geq d$. It can be verified by inspection that we have

$$P(x) = \sum_{\ell=0}^{L-1} x^{\ell \cdot d_1} Q^{(\ell)}(x, x^{L \cdot d_1}).$$

Furthermore, the uniqueness follows from a counting argument: the set of sequences of polynomials $Q^{(0)}, \ldots, Q^{(L-1)}$ form a vector space of dimension $L \cdot d_1 \cdot d_2 = d$, the dimension of the space of polynomials of degree less than d. ∎

Proposition 9 *If (p, p_κ) equal the SRS encoding of some polynomial P of degree less than $n/8$, then there exists a proof oracle that causes the SRS Proximity tester to accept with probability one.*

Proof: We prove the proposition by induction on n.

Let $Q^{(0)}, \ldots, Q^{(7)}$ be the polynomials as given by Proposition 8 applied to P with integers $d_1 = n_0/8$, $d_2 = n_1/8$, $L = 8$ and $d = n/8$. (Note that we have $d_1 \cdot d_2 \cdot L = d$, since $n_0 \cdot n_1 = n$.) For every $\ell \in \{0, \ldots, 7\}$, we let $f^{(\ell)}(\tilde{\alpha}, \beta) = Q^{(\ell)}(\tilde{\alpha}\beta, \beta^{n_0})$, $f_\kappa^{(\ell)}(\tilde{\alpha}, \beta) = Q^{(\ell)}(\kappa\tilde{\alpha}\beta, \kappa^{n_0}\beta^{n_0})$, $g^{(\ell)}(\tilde{\alpha}, \beta^{n_0}) = Q^{(\ell)}(\tilde{\alpha}, \beta^{n_0})$, and $g_\kappa^{(\ell)}(\tilde{\alpha}, \beta^{n_0}) = Q^{(\ell)}(\tilde{\alpha}, \kappa^{n_0}\beta^{n_0})$, for every $\tilde{\alpha} \in \langle\omega^{n_1}\rangle$, $\beta \in \langle\omega\rangle_{n_1}$, and $\beta^{n_0} \in \langle\omega^{n_0}\rangle$.

Note that the above choice of table $f^{(\ell)}$, $f_\kappa^{(\ell)}$, $g^{(\ell)}$, $g_\kappa^{(\ell)}$ are such that the **Outer** test accepts with probability one. Specifically, we have

$$\begin{aligned} p(\tilde{\alpha}\cdot\beta) &= P(\tilde{\alpha}\cdot\beta) \\ &= \sum_{\ell\in\{0,\ldots,7\}} (\tilde{\alpha}\cdot\beta)^{\ell n_0/8} Q^{(\ell)}(\tilde{\alpha}\beta, \tilde{\alpha}^{n_0}\beta^{n_0}) \\ &= \sum_{\ell\in\{0,\ldots,7\}} (\tilde{\alpha}\cdot\beta)^{\ell n_0/8} Q^{(\ell)}(\tilde{\alpha}\beta, \beta^{n_0}) \\ &= \sum_{\ell\in\{0,\ldots,7\}} (\tilde{\alpha}\cdot\beta)^{\ell n_0/8} f^{(\ell)}(\tilde{\alpha},\beta) \end{aligned}$$

Similarly we get $p_\kappa(\kappa\tilde{\alpha}\cdot\beta) = \sum_{\ell=0}^{7}(\kappa\tilde{\alpha}\cdot\beta)^{\ell n_0/8}\cdot f_\kappa^{(\ell)}(\tilde{\alpha},\beta)$.

Now we describe how to set up the rest of the proof oracles $\pi^{(\cdot,\cdot,\cdot)}$ such that the inner tests accept. For this part, note that the recursive calls to the SRS proximity testers are oracles that satisfy the completeness condition on smaller inputs. Consider, for example, the invocation $\mathrm{V}_{\mathrm{SRS}}^{\langle g^{(\ell)}|^{\leftrightarrow}_{\beta^{n_0}}, f^{(\ell)}|^{\leftrightarrow}_{\beta}, \pi^{(1,\beta,\ell)}\rangle}(\mathbb{F}, n_0, \omega^{n_1}, \beta)$, by **Inner** for some $\ell\in\{0,\ldots,7\}$ and $\beta\in\langle\omega\rangle_{n_1}$. We may relate the oracles to the polynomial $Q^{(\ell)}$ as follows: We have $g^{(\ell)}|^{\leftrightarrow}_{\beta^{n_0}}(\tilde{\alpha}) = g^{(\ell)}(\tilde{\alpha},\beta^{n_0}) = Q^{(\ell)}(\tilde{\alpha},\beta^{n_0})$, $f^{(\ell)}|^{\leftrightarrow}_{\beta}(\tilde{\alpha}) = f^{(\ell)}(\tilde{\alpha},\beta) = Q(\tilde{\alpha}\cdot\beta,\beta^{n_0})$. Thus, if we let $P'(\tilde{\alpha}) = Q^{(\ell)}(\tilde{\alpha},\beta^{n_0})$, and $\omega' = \omega^{n_1}$, then the pair $f^{(\ell)}|^{\leftrightarrow}_{\beta}, g^{(\ell)}|^{\leftrightarrow}_{\beta^{n_0}}$ is a codeword of the SRS code $\mathrm{SRS}(\mathbb{F}, n_0/8, \omega', \beta)$ corresponding to the encoding of P', and thus (by induction) there exists a proof $\pi^{(1,\beta,\ell)}$ that causes the recursive verifier to accept with probability one. Similar reasoning shows that the verifier also accepts with probability one when invoking $\mathbf{Inner}_\kappa$ or $\mathbf{Inner}_c$. ∎

3.5 Soundness

Finally we argue the soundness of the SRS tester. The main idea of the analysis is a simple induction. By induction, we argue that for most $\tilde{\alpha}$ and β, the functions $g^{(\ell)}|^{\updownarrow}_{\tilde{\alpha}}$, $g^{(\ell)}|^{\leftrightarrow}_{\beta^{n_0}}$, $g_\kappa^{(\ell)}|^{\updownarrow}_{\tilde{\alpha}}$, and $g_\kappa^{(\ell)}|^{\leftrightarrow}_{\beta^{n_0}}$ are close to polynomials of degree roughly $\sqrt{n}$. We then apply the Polishchuk-Spielman analysis of the "axis-parallel lines bivariate test" to conclude that $g^{(\ell)}$ and $g_\kappa^{(\ell)}$ are very close to some low-degree bivariate polynomials $Q^{(\ell)}$ and $Q_\kappa^{(\ell)}$. Furthermore, we show that these polynomials are appropriate shifts of each other. Next, we claim that the function $f^{(\ell)}(z,z^{n_0})$ is close to the function $Q^{(\ell)}(z,z^{n_0})$ and similarly for $f_\kappa^{(\ell)}$. Finally, we claim that $p(z)$ is close to $\sum_{\ell=0}^{7} z^{\ell n_0/8}\cdot Q^{(\ell)}(z,z^{n_0})$ which is a low-degree polynomial in z. We argue similarly for p_κ and the consistency of the nearest polynomials to p and p_κ follows from the consistency of Q and Q_κ. We elaborate on the details below. But first, we present a version of the Polishchuk-Spielman analysis of the bivariate test, in the form we will need it.

For set $S,T\subseteq\mathbb{F}$, function $f: S\times T\to\mathbb{F}$, and non-negative integers d_1,d_2 define $\delta^{(d_1,d_2)}(f)$ to be the fractional distance of f from a polynomial of degree d_1 in its first variable and d_2 in its second. Formally,

$$\delta^{(d_1,d_2)}(f) \triangleq \min_{\{Q:S\times T\to\mathbb{F}|\ \deg_x(Q)\le d_1, \deg_y(Q)\le d_2\}}\{\delta(f,Q)\}.$$

Let $\delta^{(d,*)}(f) = \delta^{(d,|T|-1)}(f)$ and $\delta^{(*,d)}(f) = \delta^{(|S|-1,d)}(f)$ denote the fractional distances when the degree in one of the variables is unrestricted.

Lemma 10 (Essentially from [27]) *There exists a universal constant c_0 such that the following holds. For every $S, T \subseteq \mathbb{F}$ and integers $d \leq |S|/8, e \leq |T|/8$ and function $f : S \times T \to \mathbb{F}$, it is the case that*

$$\delta^{(d,e)}(f) \leq c_0 \cdot \left(\delta^{(d,*)}(f) + \delta^{(*,e)}(f)\right).$$

A formal proof showing how this lemma follows from [27] is included in Appendix A.

We now analyze the soundness of the SRS tester.

Lemma 11 *There exists a constant c such that for every n and ϵ, if the SRS tester rejects (p, p_κ, π) with probability at most ϵ, then (p, p_κ) is within a distance of $c^{\log\log n} \cdot \epsilon$ from some codewords (P, P_κ) of the SRS code.*

Proof: Let c_0 be as in Lemma 10. Let $c_1 = 128 \cdot c_0$, $c_2 = (320 + 2c_1)$, and $c_3 = 8c_2 + 4$. We prove the lemma for $c = c_3^2$, which is (a large) constant. (Note the conditions imply $c > 1$ and $c > (2 \cdot (256 + 4c_1))^2$ as will be used later.)

We assume the lemma is true by induction for smaller n (and in particular for the recursive calls to the $\mathbf{Inner}_*$ tests), and now prove it for n. Assume $c^{\log\log n} \cdot \epsilon \leq 1$ or else the claim is vacuously true. We use below the fact that $c^{\log\log n_0} \leq c^{\log\log\sqrt{2n}} \leq c^{\log\log n - \frac{1}{2}}$ for every $c \geq 1$ and $n \geq 16$.

Let $\pi = (\{f^{(\ell)}, f_\kappa^{(\ell)}, g^{(\ell)}, g_\kappa^{(\ell)}, \{\pi^{(1,\beta,\ell)}, \pi^{(2,\beta,\ell)}\}_\beta, \{\pi^{(3,\tilde{\alpha},\ell)}\}_{\tilde{\alpha}}\}_\ell)$ be such that (p, p_κ, π) is rejected by the SRS tester with probability at most ϵ. We show below that (p, p_κ) are within distance $c^{\log\log n} \cdot \epsilon$ of some SRS codeword.

Denote by $\epsilon_O(\tilde{\alpha}, \beta)$ the probability that the **Outer** verifier rejects (p, p_κ, π) on random choice $\tilde{\alpha}$ and β. Let ϵ_O denote the expectation of $\epsilon_O(\tilde{\alpha}, \beta)$ over the choice of $\tilde{\alpha}$ and β. Similarly let $\epsilon_I(\ell, \beta)$, $\epsilon_\kappa(\ell, \beta)$, and $\epsilon_c(\ell, \tilde{\alpha})$ denote the probability that **Inner**, $\mathbf{Inner}_\kappa$, and $\mathbf{Inner}_c$, reject on random choice ℓ, β, and $\tilde{\alpha}$. Let $\epsilon_I(\ell)$, $\epsilon_\kappa(\ell)$ and $\epsilon_c(\ell)$ denote the expectations of these quantities over β and $\tilde{\alpha}$, and Let ϵ_I, ϵ_κ and ϵ_c denote the expectations over β, $\tilde{\alpha}$, and ℓ. By definition of the tester, we have $\epsilon = \frac{1}{4} \cdot (\epsilon_O + \epsilon_I + \epsilon_\kappa + \epsilon_c)$. Since these quantities are non-negative, we get $\epsilon_O, \epsilon_I, \epsilon_\kappa, \epsilon_c \leq 4\epsilon$. Similarly, we have $\epsilon_O(\ell), \epsilon_I(\ell), \epsilon_\kappa(\ell), \epsilon_c(\ell) \leq 32\epsilon$, for every $\ell \in \{0, \ldots, 7\}$.

For $\ell \in \{0, \ldots, 7\}$, denote by $Q^{(\ell)}(x, y)$ the polynomial of degree at most $n_0/8$ in x and $n_1/8$ in y that is closest to $g^{(\ell)}$ (on the domain $\langle\omega^{n_1}\rangle \times \langle\omega^{n_0}\rangle$), where ties may be broken arbitrarily. Similarly let $Q_\kappa^{(\ell)}$ be the closest polynomial to $g_\kappa^{(\ell)}$. Let $P(z) = \sum_{\ell=0}^{7} z^{\ell n_0/8} \cdot Q^{(\ell)}(z, z^{n_0})$ and let $P_\kappa(z) = \sum_{\ell=0}^{7} z^{\ell n_0/8} \cdot Q_\kappa^{(\ell)}(\kappa z, z^{n_0})$. We show below that (p, p_κ) is close to the SRS encoding of the polynomial P. (Among other facts, we also show that $P_\kappa(z) = P(\kappa \cdot z)$.)

Step 1: The functions $Q^{(\ell)}$ (and $Q_\kappa^{(\ell)}$) By the inductive hypothesis applied to $\mathbf{Inner}(\ell, \beta)$, we have $(g^{(\ell)}|_{\beta^{n_0}}^{\leftrightarrow}, f^{(\ell)}|_\beta^{\leftrightarrow})$ is $(c^{\log\log n_0} \cdot \epsilon_I(\ell, \beta))$-close to the SRS encoding of some degree $n_0/8$ polynomial. Thus $g^{(\ell)}|_{\beta^{n_0}}^{\leftrightarrow}$ is at most $(2 \cdot c^{\log\log n_0} \cdot \epsilon_I(\ell, \beta))$-close to the RS encoding of some degree $n_0/8$ polynomial. Averaging over β, we get $g^{(\ell)}$ is $(2 \cdot c^{\log\log n_0} \cdot \epsilon_I(\ell))$-close to some bivariate polynomial of degree $n_0/8$ in x (and arbitrary degree in y). A similar argument based on the $\mathbf{Inner}_c$ tests yields that $g^{(\ell)}$ is $(2 \cdot c^{\log\log n_1} \cdot \epsilon_c(\ell))$-close to some bivariate polynomial of degree

$n_1/8$ in y (and arbitrary degree in x). Now applying Lemma 10, we get that $g^{(\ell)}$ is close to some polynomial of degree $n_0/8$ in x and $n_1/8$ in y. More specifically, we have:

$$\begin{aligned}
\delta^{(n_0/8,n_1/8)}(g^{(\ell)}) &\leq c_0 \cdot \left(\delta^{(n_0/8,*)}(g^{(\ell)}) + \delta^{(*,n_1/8)}(g^{(\ell)})\right) \\
&\leq c_0 \cdot \left(2 \cdot c^{\log\log n_0} \cdot \epsilon_I(\ell) + 2 \cdot c^{\log\log n_1} \cdot \epsilon_c(\ell)\right) \\
&\leq 64 \cdot c_0 \left(c^{\log\log n_0} + c^{\log\log n_1}\right) \cdot \epsilon \\
&\leq 128 \cdot c_0 \cdot c^{\log\log n_0} \cdot \epsilon.
\end{aligned}$$

Letting $c_1 \stackrel{\text{def}}{=} 128 \cdot c_0$, we have that $\delta(g^{(\ell)}, Q^{(\ell)}) \leq c_1 \cdot c^{\log\log n_0} \cdot \epsilon$. A similar argument shows that $\delta(g_\kappa^{(\ell)}, Q_\kappa^{(\ell)}) \leq c_1 \cdot c^{\log\log n_0} \cdot \epsilon$.

Step 2: The functions $f^{(\ell)}$ and $f_\kappa^{(\ell)}$ Next we move to the functions $f^{(\ell)}$ (for any $\ell \in \{0,\ldots,7\}$) and show that for most $\tilde{\alpha}, \beta$ $f^{(\ell)}(\tilde{\alpha},\beta) = Q^{(\ell)}(\tilde{\alpha}\cdot\beta, \beta^{n_0})$ (and similarly for most $\tilde{\alpha},\beta$, $f_\kappa^{(\ell)}(\tilde{\alpha},\beta) = Q_\kappa^{(\ell)}(\kappa\cdot\tilde{\alpha}\cdot\beta, \beta^{n_0})$.

We first describe the argument informally. Consider a β such that $g^{(\ell)}|_{\beta^{n_0}}^{\leftrightarrow}$ and $f^{(\ell)}|_{\beta}^{\leftrightarrow}$ pass the **Inner** test with high probability *and* the SRS codeword correspond to the encoding of $Q(\cdot,\beta^{n_0})$. For such β, we have $f^{(\ell)}|_{\beta}^{\leftrightarrow}(\tilde{\alpha},\beta) = Q(\tilde{\alpha}\cdot\beta,\beta^{n_0})$ for most $\tilde{\alpha}$. It remains to make this argument quantitative and we do so below.

Define a β to be *good* if the fractional distance between $(g^{(\ell)}|_{\beta^{n_0}}^{\leftrightarrow}, f^{(\ell)}|_{\beta}^{\leftrightarrow})$ and the $\mathrm{SRS}(\mathbb{F}, n_0/8, \omega^{n_1}, \beta)$ encoding of $Q^{(\ell)}(\cdot,\beta^{n_0})$ is at most $1/8$. Let $\delta(\beta)$ denote the relative distance of $f^{(\ell)}|_{\beta}^{\leftrightarrow}$ to the projection of the SRS codeword nearest to $(g^{(\ell)}|_{\beta^{n_0}}^{\leftrightarrow}, f^{(\ell)}|_{\beta}^{\leftrightarrow})$ (onto the second half of the coordinates). Note that

$$\begin{aligned}
&\Pr_{\tilde{\alpha},\beta}[f^{(\ell)}(\tilde{\alpha},\beta) \neq Q^{(\ell)}(\tilde{\alpha}\cdot\beta,\beta^{n_0})] \\
&\quad\leq \mathbb{E}_\beta[\delta(\beta)|\beta \text{ is good}] \cdot \Pr_\beta[\beta \text{ is good}] + \Pr_\beta[\beta \text{ is not good}] \\
&\quad\leq \mathbb{E}_\beta[\delta(\beta)] + \Pr_\beta[\beta \text{ is not good}]
\end{aligned}$$

Note that the first term above is easily estimated as in Step 1. We get $\mathbb{E}_\beta[\delta(\beta)] \leq (2 \cdot c^{\log\log n_0} \cdot \epsilon_I(\ell)) \leq 64 \cdot c^{\log\log n_0} \cdot \epsilon$.

Next we describe two sets that cover the case where β is not good. Let S_1 be the set of all β such that the distance of $(g^{(\ell)}|_{\beta^{n_0}}^{\leftrightarrow}, f^{(\ell)}|_{\beta}^{\leftrightarrow})$ from every SRS codeword is more than $\frac{1}{8}$. For every $\beta \in S_1$ note that the $\epsilon_I(\ell,\beta) \geq \frac{1}{8c^{\log\log n_0}}$. Thus, the probability that $\beta \in S_1$ is at most $8 \cdot c^{\log\log n_0} \cdot \epsilon_I(\ell) \leq 256 \cdot c^{\log\log n_0} \cdot \epsilon$. Next, let S_2 be the set of β for which $(g^{(\ell)}|_{\beta^{n_0}}^{\leftrightarrow}, f^{(\ell)}|_{\beta}^{\leftrightarrow})$ is $\frac{1}{8}$-close to an SRS codeword, but the SRS codeword is not the encoding of $Q^{(\ell)}(\cdot,\beta^{n_0})$. For every $\beta \in S_2$, we have $Q^{(\ell)}(\tilde{\alpha},\beta^{n_0})$ and $g^{(\ell)}(\tilde{\alpha},\beta^{n_0})$ disagree for at least $\frac{5}{8}$ fraction of the $\tilde{\alpha}$'s (since $Q^{(\ell)}(\cdot,\beta^{n_0})$ and the other SRS codeword can agree on at most $n_0/8$ values of $\tilde{\alpha}$'s). Since the distance between $g^{(\ell)}$ and $Q^{(\ell)}$ is at most $c_1 \cdot c^{\log\log n_0} \cdot \epsilon$, we get that the probability that $\beta \in S_2$ is at most $\frac{8}{5} \cdot c_1 \cdot c^{\log\log n_0} \cdot \epsilon \leq 2c_1 \cdot c^{\log\log n_0} \cdot \epsilon$. Finally, we note that if β is not good, then $\beta \in S_1 \cup S_2$. Thus we get $\Pr_\beta[\beta \text{ is not good}] \leq (256 + 2c_1) \cdot c^{\log\log n_0} \cdot \epsilon$.

Putting the above together, and recalling $c_2 = (320 + 2c_1)$, we get $\Pr_{\tilde{\alpha},\beta}[f^{(\ell)}(\tilde{\alpha},\beta) \neq Q^{(\ell)}(\tilde{\alpha}\cdot\beta,\beta^{n_0})] \leq c_2 \cdot c^{\log\log n_0} \cdot \epsilon$. Similarly we also get $\Pr_{\tilde{\alpha},\beta}[f_\kappa^{(\ell)}(\tilde{\alpha},\beta) \neq Q_\kappa^{(\ell)}(\kappa\cdot\tilde{\alpha}\cdot\beta,\beta^{n_0})] \leq c_2 \cdot c^{\log\log n_0} \cdot \epsilon$.

Step 3: The functions p and p_κ Next we move to the functions p and show that $p(z)$ usually equals $P(z) = \sum_{\ell=0}^{7} z^{\ell \cdot n_0/8} Q^{(\ell)}(z, z^{n_0})$, for $z \in \langle \omega \rangle$. Note that $\langle \omega \rangle$ is in one-to-one correspondence with $\{\tilde{\alpha} \cdot \beta\}$ where $\tilde{\alpha} \in \langle \omega^{n_1} \rangle$ and $\beta \in \langle \omega \rangle_{n_1}$ and so we are interested in estimating the probability that $p(\tilde{\alpha} \cdot \beta) \neq \sum_{\ell=0}^{7} (\tilde{\alpha}\beta)^{\ell \cdot n_0/8} Q^{(\ell)}(\tilde{\alpha}\beta, \beta^{n_0})$. We consider the following events: For $\ell \in \{0, \ldots, 7\}$, let E_ℓ be the event that $f^{(\ell)}(\tilde{\alpha}, \beta) \neq Q^{(\ell)}(\tilde{\alpha}\beta, \beta^{n_0})$. Further, let E' be the event that $p(\tilde{\alpha} \cdot \beta) \neq \sum_{\ell=0}^{7} (\tilde{\alpha}\beta)^{\ell \cdot n_0/8} f^{(\ell)}(\tilde{\alpha}\beta, \beta^{n_0})$. For any ℓ, we have E_ℓ happens with probability at most $c_2 \cdot c^{\log\log n_0} \cdot \epsilon$. Further E' happens with probability at most $\epsilon_O \leq 4\epsilon \leq 4 \cdot c^{\log\log n_0} \cdot \epsilon$ (using $c \geq 1$).. Furthermore, if none of the events E', $\{E_\ell\}_\ell$ occur, then we do have $p(\tilde{\alpha} \cdot \beta) = \sum_{\ell=0}^{7} (\tilde{\alpha}\beta)^{\ell \cdot n_0/8} Q^{(\ell)}(\tilde{\alpha}\beta, \beta^{n_0})$. Thus recalling $c_3 = 8c_2 + 4$, we get that $\delta(p, P) \leq c_3 \cdot c^{\log\log n_0} \cdot \epsilon$. Similarly, we get $\delta(p_\kappa, P_\kappa) \leq c_3 \cdot c^{\log\log n_0} \cdot \epsilon$. Combining, we get that $\delta((p, p_\kappa), (P, P_\kappa)) \leq c_3 \cdot c^{\log\log n_0} \cdot \epsilon$. By the definition of $c = c_3^2$ and the condition $c^{\log\log n_0} \leq c^{\log\log n - \frac{1}{2}}$, we get that the final proximity above is at most $c^{\log\log n} \cdot \epsilon$, as desired.

All that remains to be shown is that P and P_κ are consistent, i.e., that $P_\kappa(z) = P(\kappa \cdot z)$.

Step 4: Consistency of the κ shifts We prove this part by showing that for every ℓ, Q and Q_κ are consistent, i.e., $Q_\kappa^{(\ell)}(x, y) = Q^{(\ell)}(x, \kappa^{n_0} y)$. This suffices, since we will then have $P_\kappa(z) = \sum_\ell z^{\ell n_0/8} Q_\kappa^{(\ell)}(\kappa z, z^{n_0}) = \sum_\ell z^{\ell n_0/8} Q^{(\ell)}(\kappa z, \kappa^{n_0} z^{n_0}) = P(\kappa z)$.

Fix $\ell \in \{0, \ldots, 7\}$. Define $\tilde{\alpha} \in \langle \omega^{n_1} \rangle$ to be *good* if $(g^{(\ell)}|_{\tilde{\alpha}}^{\updownarrow}, g_\kappa^{(\ell)}|_{\tilde{\alpha}}^{\updownarrow})$ is $1/8$ close to some SRS codeword *and* $g^{(\ell)}|_{\tilde{\alpha}}^{\updownarrow}$ is $1/4$ close to the evaluations of $Q^{(\ell)}(\tilde{\alpha}, \cdot)$, *and* $g_\kappa^{(\ell)}|_{\tilde{\alpha}}^{\updownarrow}$ is $1/4$ close to the evaluations of $Q_\kappa^{(\ell)}(\tilde{\alpha}, \cdot)$, It is straightforward to see that if $\tilde{\alpha}$ is good, then $Q_\kappa^{(\ell)}(\tilde{\alpha}, y) = Q^{(\ell)}(\tilde{\alpha}, \kappa^{n_0} y)$. Furthermore, if the fraction of good $\tilde{\alpha}$'s is more than $1/8$, then we will have $Q_\kappa^{(\ell)}(x, y) = Q^{(\ell)}(x, \kappa^{n_0} y)$ as desired. So it suffices to bound the probability of $\tilde{\alpha}$ being not good (to be less than $7/8$).

The three conditions above can be analyzed in a manner similar to the analysis of the probability of β not being *good* in Step 2. Specifically, we have: The probability that $(g^{(\ell)}|_{\tilde{\alpha}}^{\updownarrow}, g_\kappa^{(\ell)}|_{\tilde{\alpha}}^{\updownarrow})$ is not $1/8$ close to some SRS codeword is at most $8 \cdot c^{\log\log n_1} \cdot \epsilon_c(\ell) \leq 256 \cdot c^{\log\log n_0} \cdot \epsilon$. The probability that $g^{(\ell)}|_{\tilde{\alpha}}^{\updownarrow}$ is $1/8$ close to some SRS codeword and not $1/4$ close to the evaluations of $Q^{(\ell)}(\tilde{\alpha}, \cdot)$ is at most $2 \cdot c_1 \cdot c^{\log\log n_0} \cdot \epsilon$. Finally, the probability that $g_\kappa^{(\ell)}|_{\tilde{\alpha}}^{\updownarrow}$ is $1/8$ close to some SRS codeword and not $1/4$ close to the evaluations of $Q_\kappa^{(\ell)}(\tilde{\alpha}, \cdot)$ is at most $2 \cdot c_1 \cdot c^{\log\log n_0} \cdot \epsilon$. Combining the above we get that the probability that $\tilde{\alpha}$ is not good is at most $(256 + 4 \cdot c_1) \cdot c^{\log\log n_0} \cdot \epsilon$. In turn the final quantity is at most $(256 + 4 \cdot c_1) \cdot c^{\log\log n - \frac{1}{2}} \cdot \epsilon \leq \frac{1}{2} c^{\log\log n} \cdot \epsilon \leq \frac{1}{2} < \frac{7}{8}$ as desired. (The first inequality follows from the fact that we have $c > (2 \cdot (256 + 4 \cdot c_1))^2$.) This concludes the proof that Q and Q_κ and hence P and P_κ are consistent. Combined with Step 3, this concludes the soundness analysis. ∎

3.6 Proof of Main Theorems for RS and SRS Code

Proof of Theorem 6 (for special case of $d = n/8$): The verifier is given in Section 3.3. Its query complexity, randomness and proof length are given by Proposition 7. Its completeness is asserted by Proposition 9. Its soundness is analyzed in Lemma 11. ∎

Proof of Theorem 1: Follows from Theorem 6 by setting $\kappa = 1$. ∎

We now generalize to the case of arbitrary degree $d < n$, completing the proof of Theorems 6 and 1.

Proposition 12 *Suppose* $\mathrm{SRS}(\mathbb{F}, d, \omega, \kappa)$ *has a PCPP with length* ℓ*, query complexity* q*, randomness* r *and soundness* $s(\delta)$*. Then,*

1. *For any* $d' < d$ *the corresponding code* $\mathrm{SRS}(\mathbb{F}, d', \omega, \kappa)$ *has a PCPP with the same parameters.*
2. *For any* $d'' \leq b \cdot d$ *the corresponding code* $\mathrm{SRS}(\mathbb{F}, d'', \omega, \kappa)$ *has a PCPP with length* $b \cdot (|S| + \ell)$*, query complexity* $b \cdot q$*, randomness* r *and soundness* $\geq s(\delta/b)$*.*

Proof: For $d' < d$ we fix a polynomial $R(z)$ of degree $d - d'$, and given oracle access to $p, p_\kappa : \langle\omega\rangle \to \mathbb{F}$, we test proximity of $(p'(z), p'_\kappa(z)) = (p(z) \cdot R(z), p_\kappa(z) \cdot R(\kappa z))$ to $\mathrm{SRS}(\mathbb{F}, d, \omega, \kappa)$. Verifier can compute $R(z)$ independently of $p(z)$. The distance of (p', p'_κ) from $\mathrm{SRS}(\mathbb{F}, d, \omega, \kappa)$ equals the distance of (p, p_κ) from $\mathrm{SRS}(\mathbb{F}, d', \omega, \kappa)$. This proves part 1.

For $d'' \leq b \cdot d$, we think of a polynomial $P(z)$ of degree d'' as a sum of at most b polynomials $P^{(i)}(z)$ of degree $\leq d$,

$$P(z) = \sum_{i=0}^{b-1} z^{d \cdot i} \cdot P^{(i)}(z)$$

A proof for the degree d'' code will be composed of b proofs, one for each $P^{(i)}$. The verifier for this code will test each pair $(p^{(i)}, p_\kappa^{(i)})$ individually (using the same random coins) and then test consistency by picking a random $\sigma \in \langle\omega\rangle$ and accepting iff

$$p(\sigma) = \sum_{i=0}^{7} \sigma^{Di} p^{(i)}(\sigma) \quad \text{and} \quad p_\kappa(\sigma) =_{i=0}^{7} \sigma^{Di} p_\kappa^{(i)}(\sigma) \tag{1}$$

Proof length, completeness, randomness and query complexity follow from construction. As to soundness, if (p, p_κ) is δ-far from $\mathrm{SRS}(\mathbb{F}, d'', \omega, \kappa)$ then at least one $p^{(i)}$ must be δ/b-far from $\mathrm{SRS}(\mathbb{F}, d, \omega, \kappa)$. ∎

3.7 Corollaries

3.7.1 Boosting Soundness to half

Theorem 1 gives very weak soundness, that is at best $1/\operatorname{poly}(\log n)$. However, using randomness efficient samplers we can boost the soundness up to any constant, paying only a small price in query complexity and randomness. The following proposition follows from [18, Corollary C.5].

Proposition 13 *Assume* $S \subset \Sigma^n$ *has a PCPP with length* ℓ*, randomness* r*, query complexity* q *and soundness* s *for proximity parameter* δ *(i.e. inputs that are* δ*-far from* S *are rejected with probability at least* s*). Then for any* $s' \in (0, 1)$ *it has a PCPP with length* ℓ*, soundness* s'*, randomness* $r + O(\log \frac{1}{1-s'})$ *and query complexity* $O\left(\frac{\log \frac{1}{1-s'}}{s}\right)$*.*

Proof: We use the definition of a hitter [18, Definition C.1]. Assume $\alpha \in \Sigma^n$ is δ-far from S. For any proof $\pi \in \Sigma^\ell$, let $f = f_{(\alpha,\pi)} : \{0,1\}^r \to \{0,1\}$ be the function that specifies whether the verifier accepts (one) or rejects (zero) the pair (α, π) on random coins R. By assumption $|\{R : f(R) = 0\}| \geq s \cdot 2^r$. Thus, using the notation of [18, Definition C.1] we get $n = r, \epsilon = s, \delta = 1 - s'$. [18, Corollary C.5] gives explicit constructions of such hitters with sample complexity $O\left(\frac{\log \frac{1}{1-s'}}{s}\right)$ and randomness $r + O(\log \frac{1}{1-s'})$, completing our proof. ∎

The previous proposition shows that soundness 1/2 can be achieved for proximity as small as $\delta = 1/\operatorname{poly}(\log n)$ with only $\operatorname{poly}(\log n)$ queries. In following statement, we say $\mathrm{RS}(\mathbb{F}, \langle w \rangle, d)$ has a PCPP with soundness half for proximity parameter δ, if every word that is δ far from the code is rejected with probability at least half (the choice of half is arbitrary and could be replaced by any constant smaller than one).

Corollary 14 *There exists a universal constant $c \geq 1$ such that for any $\delta \in (0,1)$ the following holds. For $\mathbb{F}, \omega, d, n$ be as in the statement of Theorem 1, the Reed-Solomon code $\mathrm{RS}(\mathbb{F}, \langle \omega \rangle, d)$ has a PCPP over alphabet $\mathbb{F}$ with,*

Proof length $\ell(n) \leq n \log^c n$.

Randomness $r(n) \leq \log n + c \log\log n$.

Query complexity $q(n) = \log^c n/\delta$.

Soundness *half for proximity parameter δ.*

3.7.2 Locally Testable Codes

Using [11, Section 4.1] we obtain *locally testable codes* (LTCs) with poly-logarithmic rate and query complexity (over alphabet $\mathbb{F}$). A code is said to be locally testable with query complexity q and proximity parameter δ if it has a tester (randomized oracle access machine) that accepts with probability one every code-word and rejects with probability 1/2 every word that is δ-far from it. We refer the reader to [11, Section 4.1] for a proof of the following Lemma.

Lemma 15 *If $\mathrm{RS}(\mathbb{F}, S, d)$ has a PCPP of length ℓ, query complexity q and soundness $s(\delta)$, then for any $\delta \in (0,1)$, there exists a locally testable code over alphabet $\mathbb{F}$ of length $L = O(\ell/\delta)$, rate d/L, query complexity $O(q/s(\delta))$ and proximity parameter δ.*

Codes with poly-log rate and query complexity are obtained by picking $d = \Omega(|S|/\operatorname{poly}(\log |S|))$ and proximity parameter $\delta = \Omega(1/\operatorname{poly}(\log |S|))$.

Corollary 16 *There exists an explicitly constructible family of codes of arbitrarily large size n that have rate $1/\operatorname{poly}(\log n)$ over an alphabet of size n and are locally testable with $\operatorname{poly}(\log n)$ queries and proximity parameter $1/\operatorname{poly}(\log n)$.*

4 Algebraic NP-Complete Problems

In this section we develop two algebraically posed problems that are NP-complete. We start with a simple problem that searches for a univariate polynomial of low-degree such that a related bivariate polynomial is zero over an appropriate subset of its inputs. The bivariate problems leads to PCPs with quadratic proof length (Theorem 24). Next, we prove Theorem 2 by showing a somewhat more complicated problem to be NP-complete, where both the solution being sought and the constraint to be satisfied are expressed as univariate polynomials. Furthermore, the final reduction yields a problem that is NP-hard under nearly-linear length reductions from SAT, which is crucial in establishing nearly linear length PCPs.

4.1 Warmup - An NP-complete bivariate problem

In what follows we assume $\{\mathbb{F}_n\}_{n\in\mathbb{N}}$ is a infinite sequence of fields, $|\mathbb{F}_n| \geq n$, and $\{V_n \subseteq \mathbb{F}_n\}_{n\in\mathbb{N}}$ is a sequence of subsets, $|V_n| = n$.

Definition 5 (Arithmetized Graph 3-Coloring) *Instances of the language L_{AGC} are pairs of the form $(n, C : \mathbb{F}_n^4 \to \mathbb{F}_n)$, where $C(x, y, v, w)$ is a polynomial of degree at most n in x, y and of degree at most 6 in v, w. An instance $(n, C) \in L_{\mathrm{AGC}}$ if there exists a polynomial $\chi : \mathbb{F}_n \to \mathbb{F}_n$ of degree at most n such that $\forall x, y \in V_n \times V_n$, it is the case that $C(x, y, \chi(x), \chi(y)) = 0$.*

Proposition 17 *The language L_{AGC} is NP-hard. Namely, for any polynomial time computable sequence $\{(\mathbb{F}_n, V_n) : V_n \subset \mathbb{F}_n, |V_n| > n\}_{n\in\mathbb{N}}$, there exists a polynomial time reduction from 3-Col to L_{AGC} reducing graphs with n vertices to instances of the form (n, C).*

Proof: We show a reduction from Graph 3-colorability (3-Col). Given a graph $G = (V, E)$ on n vertices, we will produce an instance (n, C) such that $G \in$ 3-Col $\Leftrightarrow (n, C) \in L_{\mathrm{AGC}}$. To define the polynomial $C(x, y, v, w)$, we first define two intermediate polynomials $P_{\mathrm{Edge}}(x, y)$ and $P_{\mathrm{Eq}}(x, y)$, where P_{Edge} will encode the graph G, while P_{Eq} is fixed given V_n.

We identify the vertex set V with V_n and thus the edges maybe viewed as a function $E' : V_n \times V_n \to \{0, 1\}$ (where $E'(x, y) = 1$ iff $(x, y) \in E$; and $E'(x, x) = 0$). Now we extend[7] $E'(x, y)$ to a bivariate polynomial $P_{\mathrm{Edge}} : \mathbb{F}_n^2 \to \mathbb{F}_n$ of degree at most $n - 1$ in each variable. This gives us the first of the two intermediate polynomials. The second polynomial P_{Eq} is obtained by extending the equality function. Let $I(x, y)$ be the function that is 1 if $x = y \in V_n$ and 0 if $x, y \in V_n$ are distinct. We obtain $P_{\mathrm{Eq}}(x, y)$ by extending I to $\mathbb{F}_n \times \mathbb{F}_n$. Again P_{Eq} is of degree at most $n - 1$ in each variable.

Now, we are ready to define the polynomial C. Fix a canonical set $S \subseteq \mathbb{F}_n$ containing three distinct elements in $\mathbb{F}_n$ (representing the three colors), and let $T = \{\alpha - \beta | \alpha, \beta \in S\}$. We let

$$C(x, y, v, w) = P_{\mathrm{Edge}}(x, y) \prod_{\alpha \in T - \{0\}} (v - w - \alpha) \; + \; P_{\mathrm{Eq}}(x, y) \prod_{\beta \in S} (u - \beta).$$

(Roughly, the first term is zero if the coloring is legal, while the second term is zero if the coloring uses one of the three colors (from S).) Note that $C(x, y, v, w)$ is of degree at most n in x, y and of

[7] For $S, T \subset \mathbb{F}$, the *low degree extension* of a function $f : S \times T \to \mathbb{F}$ is the polynomial $\hat{f} : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$ that agrees with f on $S \times T$. It is easy to verify $\deg_x(\hat{f}) \leq |S| - 1, \deg_y(\hat{f}) \leq |T| - 1$.

degree at most $|T| - 1 \leq 6$ in v, w. (To verify the cardinality of $T - \{0\}$, note that it size is at most $|S| \cdot |S| - 1$.) We claim $G \in$ 3-Col $\Leftrightarrow (n, C) \in L_{\text{AGC}}$.

In the forward direction, assume there exists a 3-coloring of G. W.l.o.g., we may think of the coloring as a function $\chi : V_n \to S$. The fact that the coloring takes on values from S implies that for every $x \in V_n$, $\prod_{\beta \in S}(\chi(x) - \beta) = 0$. Thus $P_{\text{Eq}}(x, y) \cdot \prod_{\beta \in S}(\chi(x) - \beta) = 0$ every pair $x, y \in V_n$. Furthermore, the validity of the coloring implies, for $x \neq y \in V_n$, that if $P_{\text{Edge}}(x, y) = 1$, then $\chi(x) \neq \chi(y)$ and so $\chi(x) - \chi(y) \in T - \{0\}$, and thus we get,

$$\forall x, y \in V_n,\ P_{\text{Edge}}(x, y) \cdot \prod_{\alpha \in T - \{0\}} (\chi(x) - \chi(y) - \alpha) = 0.$$

Extending χ to a polynomial of degree at most n we get a polynomial $\hat{\chi} : \mathbb{F}_n \to \mathbb{F}_n$ such that $C(x, y, \chi(x), \chi(y)) = 0$ for every $x, y \in V_n$.

In the reverse direction, now assume there exists a function $\chi : \mathbb{F}_n \to \mathbb{F}_n$ such that it satisfies $C(x, y, \chi(x), \chi(y)) = 0$ for every $x, y \in V_n^2$. We first claim that for every $x \in V_n$, $\chi(x) \in S$. We get this by looking at $\phi(x, x) = P_{\text{Edge}}(x, x)(\cdots) + P_{\text{Eq}}(x, x) \cdot \prod_{\beta \in S}(\chi(x) - \beta)$. Since $P_{\text{Edge}}(x, x) = 0$ and $P_{\text{Eq}}(x, x) = 1$, we get $\phi(x, x) = \prod_{\beta \in S}(\chi(x) - \beta) = 0$ which implies $\chi(x) \in S$. Thus χ restricted to V_n gives a function $\chi' : V_n \to S$, a 3-coloring of the vertices of G. To verify that this is a legal 3-coloring (leaves no edge monochromatic), we now consider any edge $(x, y) \in E$. By construction, we have $P_{\text{Edge}}(x, y) = 1$ and $P_{\text{Eq}}(x, y) = 0$ and so we have $0 = \phi(x, y) = P_{\text{Edge}}(x, y) \cdot \prod_{\alpha \in T - \{0\}}(\chi(x) - \chi(y) - \alpha)$ implying thus that $\chi(x) \neq \chi(y) \neq 0$. This implies that G is 3-colorable, as desired. ∎

4.2 An NP-Complete univariate problem

While the reduction above is sufficient to get moderately simple PCPs, to get an even simpler PCP verifier, it would be nicer if the final condition were a "univariate" condition. Furthermore, the natural PCP for the problem described requires proofs that are quadratic in length of the original proof (the original proof being a 3-coloring has size linear in the number of vertices, while the new proof would be quadratic.) To get shorter proofs, a univariate problem seems essential and this motivates the next problem. Such a problem is the Univariate Algebraic CSP of Definition 2. Before showing a reduction of 3-SAT to this problem, thus proving Theorem 2, we compare it to the bivariate problem.

In arithmetized graph coloring, each constraint $C(x, y, \chi(x), \chi(y))$ on the witness χ is indexed by an element $(x, y) \in V_n \times V_n$. Furthermore, given the index (x, y), the values of χ needed in order to verify the constraint are given by a linear map applied to the index. Finally, the predicate applied to the inputs (x, y, v, w) is a moderately low-degree polynomial in the constraint index, and a very low (constant) degree polynomial in the remaining variables. The same holds for the L_{UACSP} given in Definition 2. Namely, a constraint on the witness A is indexed by $x \in \{\omega^0, \ldots, \omega^{n-1}\}$. Furthermore, given index x, the values of A needed to verify the constraint are given by a linear map applied to x (as specified by the constants $\alpha_1, \ldots, \alpha_t$). Finally, the predicate applied has degree n in the constraint index, and constant degree in the remaining variables.

Proof (of Theorem 2): We reduce 3-SAT-4 (3-SAT where every variable appears in at most 4 clauses) to L_{UACSP}. We first give an overview of the reduction. Given an instance ϕ of 3-SAT-4 of length n, we "embed" the instance on a hypercube of size $n' = O(n \log n)$. In other words, we associate vertices of the cube with the variables and constraints of ϕ and establish vertex disjoint

paths between vertices corresponding to constraints and the variables appearing in them. This allows us to pose the constraint satisfaction problem as one of finding an assignment to the vertices of the cube that satisfies a collection of "local constraints" (vertex variables should be 0/1, internal path variables should propagate values correctly, while clause vertices should verify that the three paths emanating from them lead to vertices have an assignment that satisfies the constraint). We then arithmetize the hypercube, so that (local) constraint names, and the (local) variables participating in them are related by simple linear relations (as one would expect from neighbors in a hypercube). The catch here is to ensure that even though the hypercube is k-dimensional for some $k \gg 1$, the arithmetization only needs one variable. Here is where we use the fact that our field $\mathbb{F}_{n'}$ contains a large cyclic subgroup $\langle \omega_{n'} \rangle$ in its multiplicative group, and we use the exponent of the generator of this cyclic group ($\omega_{n'}$) to represent the k-bit vectors that label the vertices of the hypercube. Finally, we arithmetize the constraints, in a standard way to get an instance of the univariate algebraic constraint satisfaction problem. Details below.

Assume we are given an instance ϕ of the 3-SAT-4 problem on n variables and $m \leq 4n$ clauses. Let n' be such that $n' \geq 2^k$ and there exists a hypercube with vertex set $\{0,1\}^k$ with disjoint sets $S, T \subseteq \{0,1\}^k$ of size $|S| \geq n$, $|T| \geq 4n$ such that every 4-to-3 relation $R \subseteq S \times T$ can be "routed with vertex disjoint paths" on the hypercube on $\{0,1\}^k$ in polynomial time. (Specifically, given any relation $R \subseteq T \times S$ such that any element of S is related to at most 4 elements of T and any element of T is related to at most 4 elements of S, the algorithm finds vertex disjoint paths between every pair $(i,j) \in R$.) Such a routing can be found in deterministic polynomial time provided $n' = \Omega(n \log n)$ (see [24] for several solutions to this problem).

Embedding on a hypercube: We use the routing algorithm to embed the 3-SAT instance onto a hypercube, where constraints and vertices are associated with vertices of the hypercube and a constraint only applies to the variables in its neighborhood. In particular, let $\text{Var} \subseteq S$ be any subset of size n. Associate Var with the variables of ϕ. Similarly let $\text{Clause} \subseteq T$ be any set of size m. Associate Clause with the set of clauses of ϕ and let R denote the relation that relates $i \in \text{Var}$ to $j \in \text{Clause}$ if the jth clause depends on variable i. Let ρ be a routing of R. We use ρ to define some new subsets of $\{0,1\}^k$ and to refine the set Clause. For $\ell \in [k]$ and $b \in \{0,1\}$, let $E_{\ell,b}$ be the set of vertices $\{i \in \{0,1\}^k - \text{Clause}\}$ such that $i_\ell = b$ and the edge $i + e_\ell \to i$ is used in the routing ρ. Finally, for every tuple $\tau = (\ell_1, \ell_2, \ell_3, b_1, b_2, b_3, c_1, c_2, c_3)$ with $\ell_1, \ell_2, \ell_3 \in [k]$ and $b_1, b_2, b_3, c_1, c_2, c_3 \in \{0,1\}$, we define the set Clause_τ to be the set of all vertices $i \in \text{Clause}$ such that for every $t \in \{1,2,3\}$, it holds that $i_{\ell_t} = b_t$ and $i + e_{\ell_t} \to i \in \rho$ and this edge routes a variable i to clause j where the variable i is negated iff $c_t = 1$. Notice that the formula ϕ (or the routing ρ) is completely specified by the sets Var, $\{E_{\ell,b}\}_{\ell,b}$ and $\{\text{Clause}_\tau\}_\tau$ and these sets are disjoint. Indeed the satisfiability problem can be reformulated as the task of finding an assignment A of the vertices of $\{0,1\}^k$ (to an arbitrary universe that includes 0 and 1) such that the following conditions hold.

- For every vertex $i \in \text{Var}$, $A(i) \in \{0,1\}$.
- For every vertex $i \in E_{\ell,b}$, $A(i + e_{\ell,b}) = A(i)$ (for every ℓ, b).
- For every vertex $i \in \text{Clause}_\tau$ where $\tau = (\ell_1, \ell_2, \ell_3, b_1, b_2, b_3, c_1, c_2, c_3)$, there exists a $t \in \{1,2,3\}$ such that $A(i + e_{\ell_t}) = c_t$.

Arithmetizing the hypercube: Let $\mathbb{F} = \mathbb{F}_{n'}$ and $\omega = \omega_{n'} \in \mathbb{F}$ generate a multiplicative group of size $> n'$ (recall we assume $\mathbb{F}$ and ω can be found in polynomial time). We embed the hypercube

on a Cayley graph G over vertex set $\langle\omega\rangle$. The generators of the (edges of the) graph will be the set of elements $\{\beta_{\ell,b}\}_{\ell\in[k],b\in\{0,1\}}$ where $\beta_{\ell,b} = \omega^{(-1)^b\cdot 2^\ell}$. In other words, vertices x and y are adjacent in G iff there exists ℓ, b such that $x = \beta_{\ell,b}\cdot y$. To see that this graph embeds the hypercube on $\{0,1\}^k$ let $[i]\in\{0,\ldots,2^k-1\}$ denote the integer with binary representation $i\in\{0,1\}^k$. Associate with i the element $\omega^{[i]}\in\langle\omega\rangle$. We claim that the elements associated with i and $i+e_\ell$ are adjacent in G. Let x be the vertex of G associated with i and let $i_\ell = b$. Then $i+e_\ell$ is associated with $\omega^{[i]+(-1)^b 2^\ell} = \beta_{\ell,b}\cdot\omega^{[i]}$, and thus the corresponding vertices are adjacent in G.

Arithmetizing the constraints: Notice that under the association from the previous paragraph, the sets Var, $E_{\ell,b}$, and Clause_τ can be thought of as (disjoint) subsets of the vertices of G. In fact, with some abuse of notation, we'll let Var denote the function from $H = \{\omega^0,\omega^1,\ldots,\omega^{n'-1}\}$ to $\{0,1\}$ serving as the indicator of the set Var (so $\text{Var}(\omega^{[i]}) = 1$ iff $i\in\text{Var}$). Similarly for $E_{\ell,b}$ and Clause_τ. Now we extend the functions Var, $E_{\ell,b}$ and Clause_τ to get polynomials $\widehat{\text{Var}}$, $\widehat{E}_{\ell,b}$ and $\widehat{\text{Clause}}_\tau$ from $\mathbb{F}$ to $\mathbb{F}$ of degree at most $|H|-1 = n'-1$. The satisfiability of ϕ now reduces to the question of the existence of a function $A:\mathbb{F}\to\mathbb{F}$ such that the following conditions hold:

1. For every $x\in H$, $\widehat{\text{Var}}(x)\cdot A(x)\cdot(A(x)-1) = 0$.
2. For every ℓ, b and for every $x\in H$, $\widehat{E}_{\ell,b}(x)\cdot(A(x)-A(\beta_{\ell,b}\cdot x)) = 0$.
3. For every $\tau = (\ell_1,\ell_2,\ell_3,b_1,b_2,b_3,c_1,c_2,c_3)$, and for every $x\in H$, $\widehat{\text{Clause}}_\tau(x)\cdot\prod_{t\in\{1,2,3\}}(A(i+e_{\ell_t})-c_t) = 0$.

The reduced instance: These conditions motivate the following definition of the clause polynomial (which is the sum of the three classes of constraints above):

$$\begin{aligned} C(x,y_0,\{y_{\ell,b}\}_{\ell,b}) = &\ \widehat{\text{Var}}(x)\cdot y_0\cdot(y_0-1) \\ &+ \textstyle\sum_{\ell,b}\widehat{E}_{\ell,b}(x)\cdot(y_0-y_{\ell,b}) \\ &+ \textstyle\sum_{\tau=(\ell_1,\ell_2,\ell_3,b_1,b_2,b_3,c_1,c_2,c_3)}\widehat{\text{Clause}}_\tau(x)\cdot\prod_{t\in\{1,2,3\}}(y_{\ell_t,b_t}-c_t) \end{aligned}$$

Notice that C is indeed a polynomial of degree at most $n'-1$ in the first variable and of total degree at most 3 in the remaining variables. For the function C as above, we claim that the instance $(n',1,\{\beta_{\ell,b}\}_{\ell,b},C)$ is a member of L_{UACSP} iff ϕ is satisfiable. Notice the number of inputs to the constraint polynomial C is $t = 1+|\{\beta_{\ell,b}\}_{\ell,b}| = 1+2k = O(\log n)$, as claimed.

Completeness: In the forward direction, note that if ϕ is satisfiable, then there exists a partial function $A':H\to\mathbb{F}$ that satisfies conditions (1)-(3) enumerated above. (In particular, note that whenever $E_{\ell,b}(x)\neq 0$ for $x\in H$, then $\beta_{\ell,b}\cdot x\in H$, and similarly for the arguments of A' considered in condition (3). So the values of A' on the domain H are all that determine satisfiability.) Taking A to be the low degree extension of A' to the entire domain $\mathbb{F}$ gives us a polynomial of degree $n'-1$ that satisfies conditions (1)-(3), and thus satisfies the condition $C(x,A(x),\{A(\beta_{\ell,b}\cdot x)\}_{\ell,b}) = 0$ for every $x\in H$.

Soundness: In the reverse direction, suppose A is a polynomial satisfying $C(x, A(x), \{A(\beta_{\ell,b} \cdot x)\}_{\ell,b}) = 0$ for every $x \in H$. We claim that A also satisfies conditions (1)-(3). To see this, consider an arbitrary $x \in H$. Recall that at most one of the polynomials $\widehat{\mathrm{Var}}$, $\widehat{E}_{\ell,b}$, $\widehat{\mathrm{Clause}}_\tau$ is non-zero for this choice of x. Assume, for simplicity, that $\widehat{\mathrm{Var}}(x) \neq 0$. Then since $C(x, \cdots) = 0$, we have $\widehat{\mathrm{Var}}(x) \cdot A(x) \cdot (1 - A(x)) = 0$ and thus condition (1) is satisfied for this x. But so are conditions (2) and (3), since $\widehat{E}_{\ell,b}$ and $\widehat{\mathrm{Clause}}_\tau$ are also zero. The other cases are similar. Thus A satisfies conditions (1)-(3) and this is equivalent to finding a satisfying assignment for ϕ (in particular the assignment to elements in Var is Boolean, and forms a satisfying assignment to ϕ). This concludes proof of the soundness of the reduction. ∎

5 PCPs for Univariate Algebraic CSPs

We now provide efficient PCPs for L_{UACSP} and prove Theorem 4. Before we do so we attend to remaining problems. We prove Lemma 3 and discuss the abundance of prime fields with 2-smooth multiplicative sub-groups (necessary for our efficient PCPPs for the RS code).

5.1 Proof of Lemma 3

Recall a polynomial $P(z)$ is zero on H iff the polynomial $g_H(z) \triangleq \prod_{h \in H}(z - h)$ divides it, i.e. $P(z) = g_H(z) \cdot \tilde{P}(z)$ for some polynomial $\tilde{P}, \deg(\tilde{P}) \leq d - |H|$. The verifier for $\mathrm{RS}_H(\mathbb{F}, S, d)$ has oracle access to the purported codeword $p \in \mathbb{F}^S$ and its proof, combined of three parts: *(i)* $\tilde{p} : S \to \mathbb{F}$ a supposed evaluation of $\tilde{P}$ on S; *(ii)* A proof of proximity for p to $\mathrm{RS}(\mathbb{F}, S, d)$ and *(iii)* A proof of proximity for $\tilde{p}$ to $\mathrm{RS}(\mathbb{F}, S, d - |H|)$. Proof length is as claimed. The verifier operates as follows. First, proximity of p (to degree d) and $\tilde{p}$ (to degree $d - |H|$) are tested. Then, a random $\alpha \in S$ is selected and verifier accepts iff $p(\alpha) = g_H(\alpha) \cdot \tilde{p}(\alpha)$. Notice $g_H(\alpha)$ can be computed by the verifier as long as H is known in advance. The query complexity is as claimed. Completeness follows by our previous discussion (taking $\tilde{p}$ to be the evaluation of $\tilde{P}$). As to the soundness and proximity parameter, if p is δ-far from $\mathrm{RS}(\mathbb{F}, S, d)$ or $\tilde{p}$ is δ-far from $\mathrm{RS}(\mathbb{F}, S, d - |H|)$ the verifier rejects with probability $\geq s$. Otherwise, the polynomial $\tilde{P}$ closest to $\tilde{p}$ does not divide P (because p is far from $\mathrm{RS}_H(\mathbb{F}, S, d)$), so $g_H \cdot \tilde{P}$ is a polynomial of degree d that is not equivalent to P. The two polynomials can agree on at most d inputs. Thus, our verifier accepts with probability $\leq 2\delta + d/|S|$. Randomness can be reused across different tests, completing the proof.

5.2 Linnik's Theorem

Recall that our efficient PCPPs are constructed only for Reed-Solomon codes $\mathrm{RS}(\mathbb{F}, \langle\omega\rangle, d)$ where $|\langle w\rangle|$ is a power of two (the reduction from SAT to L_{UACSP} only needs $\langle\omega\rangle$ to be large enough). The following (special case of a) Theorem due to Linnik shows there is a polynomial time computable sequence $\{\mathbb{F}_n\}_{n \in \mathbb{N}}$ such that $n \leq |\mathbb{F}_n| \leq n^{O(1)}$ and $\mathbb{F}_n$ has an element $\omega \in \mathbb{F}_n^\star$ such that $|\langle w\rangle| = O(n)$ is a power of two.[8] Notice that ω can be found in polynomial time once $\mathbb{F}_n$ is known, by exhaustively searching $\mathbb{F}_n^\star$. Additionally, each element of $\mathbb{F}_n$ is represented by $O(\log n)$ bits.

[8] The general statement of Linnik's Theorem says there exists a universal constant L such that for every pair of integers $0 < a < n$, there exists a prime $p < n^L$ such that $n|(p - a)$. Our case is derived by setting $a = 1$. For more details, see `http://mathworld.wolfram.com/LinniksTheorem.html`

Theorem 18 (Linnik's Theorem) *[25] There exists a constant $1 < L < 6$ such that for any sufficiently large d, there exists a prime of size $\leq d^L$ such that $d|(p-1)$*

Let d be a power of two such that $n < d \leq O(n)$. Let $\mathbb{F}_n$ be the prime field $\mathbb{Z}_p$ for p as in Linnik's Theorem. p can be found in polynomial time (and its primality verified) by exhaustive search. $|\mathbb{F}_n^\star| = p - 1 = k \cdot d$. Let σ be a generator of $\mathbb{F}^\star$. It is easy to verify $n < |\langle \sigma^k \rangle| \leq O(n)$ is a power of two.

5.3 Proof of PCP Theorem 4

Let $\{(\mathbb{F}_n, \omega_n) : \omega_n \in \mathbb{F}_n^\star\}_{n \in \mathbb{N}}$ be a polynomial time computable sequence where $16n < |\langle \omega \rangle| \leq 32n$ and $|\langle \omega \rangle|$ is a power of two. Such a sequence is guaranteed by Linnik's Theorem 18. Assume w.l.o.g. we are given a 3-SAT-4 instance ϕ with n variables and $\leq 4n$ clauses (recall there is a nearly linear size reduction from SAT to 3-SAT-4). Our verifier reduces ϕ to an instance $(n', \alpha_1, \ldots, \alpha_t, C)$ of L_{UACSP} using Theorem 2. Recall $n' = O(n \log n), t = O(\log n)$ and $\alpha_1, \ldots, \alpha_t \in \langle \omega_{n'} \rangle$. Additionally, $C : \mathbb{F}_{n'}^{t+1} \to \mathbb{F}_{n'}$ has degree n' in its first variable and degree 3 in the remaining ones. Let $\mathbb{F} = \mathbb{F}_{n'}, \omega = \omega_{n'}$ and $H = \{\omega^0, \ldots, \omega^{n'-1}\}$.

The PCP Construction Verifier has oracle access to two purported Reed-Solomon codewords and their PCPPs. These are the assignment oracle $p_A \in \mathbb{F}^{\langle \omega \rangle}$ and its proof of proximity to $\text{RS}(\mathbb{F}, \langle \omega \rangle, n')$ (denoted π_A) and the constraint oracle $p_B \in \mathbb{F}^{\langle \omega \rangle}$ and its proof of proximity to $\text{RS}_H(\mathbb{F}, \langle \omega \rangle, 4n')$ (denoted π_B). The length of each polynomial is $O(n \log n)$ and that of its PCPP is $n \,\text{poly}(\log n)$, as claimed. The total proof length is $n \cdot \text{poly}(\log n)$ even when measured in bits, because every element of $\mathbb{F}$ can be written using $O(\log n)$ bits.

Verifier's Operation Fix $\delta = 1/4(t+1)$. Verifier tests proximity of each polynomial to its respective code using the proofs of proximity. Each test makes $O(\delta)$ queries, as to reject with probability $1/2$ if the table is δ-far from the respective code (as guaranteed by Corollary 14). If any test rejects, the Verifier rejects. Otherwise, Verifier selects a random $\beta \in \langle \omega \rangle$, and queries p_B at β and p_A at values $(\beta, \alpha_1\beta, \ldots, \alpha_t\beta)$. Verifier accepts iff $p_B(\beta) = C(\beta, \alpha_1\beta, \ldots, \alpha_t\beta)$. Query complexity is as claimed, and so is the randomness that can be reused across different tests.

Completeness By Theorem 2, if ϕ is satisfiable, there exists an assignment polynomial P_A of degree n' such that $P_B(x) \triangleq C(x, P_A(\alpha_1 x), \ldots, P_A(\alpha_t x))$ is zero on every $x \in H$. The proof for satisfiability of ϕ is the evaluation of P_A, P_B on $\langle \omega \rangle$ (with their proofs of proximity). This proof is accepted with probability one, as implied by Theorem 1 and Lemma 3.

Soundness Suppose ϕ is unsatisfiable. There are several cases to consider. If one of p_A, p_B is δ-far from the corresponding RS-code, Corollary 14 implies Verifier rejects with probability $1/2$, and we are done.

Otherwise, let P_A be the unique degree n' polynomial that is δ-close to p_A and let P_B be the unique polynomial of degree $4n'$ that is δ-close to p_B and vanishes on H. Since ϕ is unsatisfiable, Theorem 2 implies $P_B(x) \neq C(x, P_A(\alpha_1 x), \ldots, P_A(\alpha_t x))$, so the two polynomials agree on at most $4n'$ values. Thus, by a union bound, Verifier accepts with probability $\leq (t+1)\delta + \frac{4}{16}) = 1/2$.

Randomness The rejection of any one test makes Verifier reject. Thus, we can reuse random coins across different tests. The randomness of the proximity test is $\log n + O(\log\log n)$ as claimed in Theorem 1, and the randomness required by the other tests is $\log|\langle\omega\rangle| = \log n + O(\log\log n)$. The proof of Theorem 4 is complete.

6 PCPs for Multivariate Algebraic CSPs

6.1 PCPPs for Reed-Muller Codes

It is easy to extend the PCPP for the RS-code into one for the Reed-Muller code (based on multivariate polynomials), given the extensive literature on testing multivariate polynomials using axis parallel lines [5, 6, 16, 3, 27, 17]. Let $\mathrm{RM}(\mathbb{F}, S, d, m)$ be the m-variate Reed-Muller code with degree bound d, evaluated at S^m,

$$\mathrm{RM}(\mathbb{F}, S, d, m) = \left\{\langle Q(x_1, \ldots, x_m)\rangle_{x_1 \leftarrow S, \ldots, x_m \leftarrow S} : \forall i \in [m], \deg_{x_i}(Q) \leq d\right\}$$

For a set $S \subseteq \mathbb{F}$ and m-variate function $f : S^m \to \mathbb{F}$, let $\delta^d_m(f)$ be the fractional distance of f from $\mathrm{RM}(\mathbb{F}, S, d, m)$. Let $\delta^d_{m,i}(f)$ denote the fractional distance of f from a polynomial of degree d in the ith variable, and unbounded degree in all other variables. Finally, let $\mathbb{E}[\delta^d_{m,i}(f)]$ be the expectation of $\delta^d_{m,i}$ over random $i \in [m]$. The following Lemma is a rephrasing of [3, Lemma 5.2.1]. Notice Lemma 10 is a special case of it (with tighter parameters).

Lemma 19 *[3] There exists a universal constant c such that for every $S \subset \mathbb{F}$ such that $|S| \geq \mathrm{poly}(m, d)$,*

$$\delta^d_m(f) \leq c \cdot \mathbb{E}[\delta^d_{m,i}(f)]$$

This Lemma together with Theorem 1 imply efficient PCPPs for Reed-Muller codes.

Lemma 20 (RM PCP of Proximity) *Let $S \subset \mathbb{F}$ and d, m be integers such that $|S| \geq poly(m, d)$ for the polynomial of Lemma 19. If $\mathrm{RS}(\mathbb{F}, S, d)$ has a PCPP with length ℓ, query complexity q, randomness r and soundness $s(\delta)$, then the Reed-Muller Code $\mathrm{RM}(\mathbb{F}, S, d, m)$ has a PCPP with length $\leq m \cdot n^{m-1} \cdot \ell$, query complexity q, randomness $\log(m \cdot n^{m-1}) + r$ and soundness $\geq s(\delta)/m$.*

Proof: The proof for a purported RM-codeword is the collection of proofs of proximity for each axis parallel line (to the RS code). (A line parallel to the ith axis is $\{(b_1, \ldots, b_{i-1}, x_i, b_{i+1}, \ldots, b_m) : x_i \in S\}$ where $b_1, \ldots, b_m \in S$.) The verifier selects a random axis parallel line and invokes the RS-verifier of Definition 4 on the line and its proof. The proof follows from Lemma 19. ∎

Remark 21 *A more query efficient test can be constructed when $\langle\omega\rangle = \mathbb{F}^\star$. Instead of axis parallel lines, we use an ϵ-biased set of directions as in [12]. This results in proofs of similar length and query complexity and slightly larger randomness, but the soundness query complexity is as large as $\Omega(s(\delta))$ (and independent of m).*

6.2 Multivariate Zero Testing and Proof of Lemma 5

The following generalization of the univariate zero testing (Section 5.1) is crucial in many PCP constructions, starting with [6]. We are given $H, S \subseteq \mathbb{F}$ and oracle access to a function $f : S^m \to \mathbb{F}$. Our task is to verify that f is close to a low-degree multivariate polynomial that evaluates to zero on H^m. In other words, we are interested in testing proximity to the code $\mathrm{RM}_H(\mathbb{F}, S, d, m) \subseteq \mathrm{RM}(\mathbb{F}, S, d, m)$ corresponding to low degree polynomials that vanish on H^m (as in the univariate case, we do not require H to be a subset of S). The catch in immediately extending the univariate tester of Lemma 3 to even the bivariate case is that the "factoring" concept does not extend immediately. Specifically, if we are given that a bivariate polynomial $Q(x, y)$ has a zero at (α, β) this does not imply that $Q(x, y)$ has some nice factors. However, one can abstract a nice property about Q from this zero. Specifically, we can say that there exist polynomials $A(x, y), B(x, y)$ (of the right degree) such that $Q(x, y) = A(x, y) \cdot (x - \alpha) + B(x, y) \cdot (x - \beta)$. Thus to prove that $Q(\alpha, \beta) = 0$, we may ask the prover to give oracles for $Q(x, y)$. $A(x, y)$ and $B(x, y)$. We can then test that Q, A and B are of low-degree and that they satisfy the identity above. Extending this idea to m-variate polynomials that are zero on an entire generalized rectangle is straightforward. The technical lemma giving the identity is included below. (The lemma is also a key ingredient in Alon's "Combinatorial Nullstellensatz" [1]. We include a proof for completeness.)

Lemma 22 *Let $Q(x_1, \ldots, x_m)$ be a polynomial over $\mathbb{F}_Q$ of degree d in each of m variables. Let $H \subseteq \mathbb{F}_Q$ and let $g_H(z) \stackrel{\text{def}}{=} \prod_{\beta \in H}(z - \beta)$. Then Q evaluates to zero on H^m iff there exist m-variate polynomials $A_1, \ldots, A_m$ of individual degree at most d such that $Q(\vec{x}) = \sum_{i=1}^{m} A_i(\vec{x}) \cdot g_H(x_i)$.*

Remark 23 *The lemma above is intentionally sloppy with degree bounds. While tighter degree bounds on A_i's can be obtained, this won't be needed for our PCPs.*

Proof: One direction is immediate. If $Q(\vec{x}) = \sum_{i=1}^{m} A_i(\vec{x}) \cdot g_H(x_i)$ then $Q(\vec{\alpha}) = 0$ for every $\vec{\alpha} \in H^m$. The other direction is proved in three steps. First, we show that for any polynomial $P(x_1, \ldots, x_m)$ of degree d_j in x_j, and any $i \in \{1, \ldots, m\}$, there exist polynomials $B(x_1, \ldots, x_m)$ and $C(x_1, \ldots, x_m)$ of degree at most d_j in x_j, with the degree of C in x_i being at most $\min\{d_j, |H| - 1\}$, such that $P(\vec{x}) = B(\vec{x}) \cdot g_H(x_i) + C(\vec{x})$. Second, we show that there exist polynomials $A_1, \ldots, A_m$ and R with the A_i's having degree at most d in each variable and R having degree at most $|H| - 1$ in each variable such that $Q(\vec{x}) = \sum_{i=1}^{m} A_i(\vec{x}) \cdot g_H(x_i) + R(\vec{x})$ (where Q is the polynomial from the lemma statement). In the final step, we show that $R(\vec{x}) = 0$, concluding the proof.

STEP 1: Recall that any polynomial $f(x_i)$ can be written as $q(x_i) \cdot g_H(x_i) + r(x_i)$ where r has degree less than $|H|$. Applying this fact to the monomials x_i^D (for non-negative D) we find that there exist polynomials $q_D(x_i)$ and $r_D(x_i)$, with degree of q_D being at most D and degree of r_D being less than $|H|$, such that $x_i^D = q_D(x_i) \cdot g_H(x_i) + r_D(x_i)$. Now consider any polynomial $P(x_1, \ldots, x_m)$ of degree d_i in x_i. Suppose $P(\vec{x}) = \sum_{D=0}^{d_i} P_i(\vec{x}') \cdot x^D$, where $\vec{x}' = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_m)$. Writing the monomials x_i^D in terms of the q_D's and r_D's, we get: $P(\vec{x}) = \left(\sum_{D=0}^{d_i} P_i(\vec{x}') q_D(x_i)\right) \cdot g_H(x_i) + \left(\sum_{D=0}^{d_i} P_i(\vec{x}') r_D(x_i)\right)$. Letting $B(\vec{x}) = \sum_{D=0}^{d_i} P_i(\vec{x}') q_D(x_i)$ and $C(\vec{x}) = \left(\sum_{D=0}^{d_i} P_i(\vec{x}') r_D(x_i)\right)$, yields the polynomials as claimed. (In particular the degrees of B and C in any variable are no more than of P, and the degree of C in x_i is smaller than $|H|$.)

STEP 2: We now claim that there exist polynomials $A_1, \ldots, A_m$ and $R_0, \ldots, R_m$ such that for every $j \in \{0, \ldots, m\}$, $Q(\vec{x}) = \sum_{i=0}^{j} A_i(\vec{x}) \cdot g_H(x_i) + R_j(\vec{x})$, with A_i's being of degree at most d in each

variable and R_j being of degree less than $|H|$ in $x_1, \ldots, x_j$ and of degree at most d in the remaining variables. The proof is straightforward by induction on j, with the induction step using Step 1 on the polynomial $P() = R_j()$ and the variable x_{j+1}. The final polynomials $A_1, \ldots, A_m$ and $R = R_m$ are the polynomials as required to yield the sub-claim of this step.

STEP 3: Finally we note that for every $\vec{\alpha} \in H^m$, we have $R(\vec{\alpha}) = Q(\vec{\alpha}) - \sum_{i=1}^{m} A_i(\vec{\alpha}) \cdot g_H(\alpha_i) = 0 - \sum_{i=1}^{m} 0 = 0$. But R is a polynomial of degree less than $|H|$ in each variable and is zero on the entire box H^m. This can only happen if $R \equiv 0$. Thus we get that $Q(\vec{x}) = \sum_{i=1}^{m} A_i(\vec{x}) \cdot g_H(x_i)$, with A_i's being of degree at most d in each variable, as required in the completeness condition. ∎

The lemma above, combined with the multivariate polynomial tester from Lemma 20 prove Lemma 5.

Proof (of Lemma 5): As a proof of the proximity of $q \in \mathbb{F}^{S^m}$ to $\mathrm{RM}_H(\mathbb{F}, S, d, m)$ our verifier expects *(i)* the evaluations of $A_1, \ldots, A_m$ from Lemma 22 on S^m (denoted $a_1, \ldots, a_m$) and *(ii)* for each of $q, a_1, \ldots, a_m$, a proof of proximity of A_i to $\mathrm{RM}(\mathbb{F}, S, d, m)$. Proof length is as claimed. The verifier operates as follows. First, it tests proximity of each of $q, a_1, \ldots, a_m$ to $(\mathbb{F}, \mathrm{S}, \mathrm{d}, \mathrm{m})$. Then, a random $\langle \alpha_1, \ldots, \alpha_m \rangle \in S^m$ is selected and verifier accepts iff $q(\vec{\alpha}) = \sum_{i=1}^{m} g_H(\alpha_i) \cdot a_i(\vec{\alpha})$. The query complexity is as claimed. Completeness follows from Lemma 22. As to the soundness, if any of $q, a_1, \ldots, a_m$ is δ-far from $(\mathbb{F}, \mathrm{S}, \mathrm{d}, \mathrm{m})$ Verifier rejects with probability $s(\delta)$. Otherwise, q is δ close to a polynomial Q that doesn't vanish on H^m. If $A_1, \ldots, A_m$ are the polynomials closest to $a_1, \ldots, a_m$ respectively, then by Lemma 22 we get $Q(\vec{x}) \neq \sum_i A_i(\vec{x}) \cdot g_H(x_i)$ and Q has degree at most $2d$ in each variable. Thus, the two polynomials agree on $\leq (2d)^m$ points so the acceptance probability of Verifier is $\leq (m+1)\delta + \left(\frac{d}{|S|}\right)^m$ as claimed. ∎

6.3 PCPs for L_{AGC}

For completeness, and to illustrate the use of the multivariate polynomial zero tester, we now describe a PCP for the algebraic graph coloring problem. While the parameters of the following theorem are strictly weaker than those of Theorem 4 (the randomness is twice as large and the proof length is quadratically bigger), its proof is simpler.

Theorem 24 $NP \subseteq PCP_{1,\frac{1}{2}}[2\log n + O(\log\log n), \mathrm{poly}\log n]$.

Proof: First we reduce 3-COL to L_{AGC} as in Proposition 17. Given graph G on n vertices we pick our field $\mathbb{F} = \mathbb{F}_n$ to have an element $\omega \in \mathbb{F}^\star$ such that $|\langle\omega\rangle|$ is a power of two and $16n < |\langle\omega\rangle| \leq 32n$. Let (n, C) be the resulting Algebraic CSP.

Proof Oracle Our verifier expects oracle access to: *(i)* An evaluation of a degree n polynomial $p \in \mathbb{F}^{\langle\omega\rangle}$ (supposedly encoding the three coloring), together with its proof of proximity to $\mathrm{RS}(\mathbb{F}, \langle\omega\rangle, n)$, denoted π_p; And *(ii)* An evaluation of a bivariate polynomial $q \in \mathbb{F}^{\langle\omega\rangle \times \langle\omega\rangle}$ together with its proof of proximity to $\mathrm{RM}_{V_n}(\mathbb{F}, \langle\omega\rangle, 4n, 2)$, denoted π_q.

Operation Verifier tests proximity of p to $\mathrm{RS}(\mathbb{F}, \langle\omega\rangle, n)$ (using π_p) and of q to $\mathrm{RM}_{V_n}(\mathbb{F}, \langle\omega\rangle, 4n, 2)$ (using π_q). Both tests make $\mathrm{poly}(\log n)$ queries as to guarantee rejection probability $1/2$ for proximity parameter $\delta = 1/16$. If either test rejects then verifier rejects. Otherwise, verifier chooses random $(x, y) \in \langle\omega\rangle \times \langle\omega\rangle$ and accepts iff $q(x, y) = C(x, y, p(x), p(y))$. Proof length is as claimed.

Completeness If G has a three coloring, the low-degree extension of a non-monochromatic three-coloring χ has the property that $C(x, y, \chi(x), \chi(y)) = 0$ for all $(x, y) \in V_n \times V_n$. Thus, the evaluation of these two polynomials (with their proofs of proximity) pass the verifier's test with probability one.

Soundness Suppose G is not three-colorable. If either of p, q is not δ-close to the desired code, verifier rejects with probability 1/2. Otherwise, by the soundness of Proposition 17, if P, Q are the polynomials closest to p, q respectively, then $Q(x, y) \neq C(x, y, P(x), P(y))$, because Q evaluates to zero on $V_n \times V_n$ and G is not three colorable. Thus, the two polynomials can agree on $\leq (4n)^2$ entries. In this case, the acceptance probability is at most $2\delta + (\frac{4n}{|\langle\omega\rangle|})^2 < 1/2$. ∎

7 Implementation Considerations

We briefly discuss the issues arising when implementing our PCP constructions. We start by noting that most constants discussed above can be further optimized and we leave this issue for future research.

Fields Our PCPs of proximity for the RS code require fields with multiplicative groups of size that is a power of two. While such a field can be found in polynomial time by brute force given a circuit of size n (as guaranteed by Linnik's Theorem 18), we can also envision a short list that would good enough to deal with all "small" circuits (say, smaller than the number of atoms in the universe). Such a list would have less than a thousand entries (using conservative estimates of the size of the universe). An alternative solution would be to find explicit fields with multiplicative sub-groups that are poly$(\log n)$-smooth (noticing Theorem 1 can be extended to such groups).

Routing The most time-consuming part of our computations (both for proof construction and for the Verifiers operation) is computing a good embedding of the SAT instance into a hypercube. Efficient (nearly-linear running time) algorithms for this problem exist [24]. However, to obtain a poly-logarithmic running time Verifier for uniform computations as in [6], we need to compute an edge of the embedding in poly-logarithmic time. We leave this problem for future investigation. Notice the efficient routing problem does not arise in the longer (quadratic length) bivariate PCPs of Section 6.2.

Consistency Tests Most of the tests the verifier makes (assuming the routing is known) require poly-logarithmic running time. The only exception is the computation of the degree $n' = O(n \cdot \text{poly}(\log n))$ polynomial $g_H(x) = \prod_{h \in H}(x - h)$. There is a simple fix to this problem, suggested by Salil Vadhan [Personal Communication]. Namely, if we take $H = \langle\omega'\rangle$ for some $\omega' \in \mathbb{F}^\star$ then $g_H(x) \equiv x^{|H|} - 1$ (both sides of the equation are of degree $|H|$ and vanish on H). This polynomial can be computed in poly-logarithmic time. It can be readily verified that we can use such an H for our PCP constructions (e.g., if ω generates the multiplicative group needed for the RS-testing to work, and $|\langle\omega\rangle|$ is a power of two, we may take $\omega' = \omega^{2k}$ for some small constant k, such that the size of $\langle\omega'\rangle$ is still larger than n').

8 Acknowledgements

We thank Oded Goldreich, Prahladh Harsha, Salil Vadhan and Chris Umans for helpful discussions. We thank Don Coppersmith for pointing us to Linnik's Theorem.

References

[1] Alon, N. Combinatorial Nullstellensatz. *Combinatorics, Probability and Computing*, 8 (1999), 7-29.

[2] Arora, S., Lund, C., Motwani, R., Sudan, M., and Szegedy, M. Proof verification and the hardness of approximation problems. *Journal of the ACM 45*, 3 (May 1998), 501–555. (Preliminary Version in *33rd FOCS*, 1992).

[3] Arora, S., and Safra, S. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM 45*, 1 (Jan. 1998), 70–122. (Preliminary Version in *33rd FOCS*, 1992).

[4] Arora, S., and Sudan, M. Improved low degree testing and its applications. *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 485-495, El Paso, Texas, 4-6 May 1997.

[5] Babai, L., Fortnow, L. and Lund, C. Nondeterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1(1):3-40, 1991.

[6] Babai, L., Fortnow, L., Levin, L. A., and Szegedy, M. Checking computations in polylogarithmic time. In *Proc. 23rd ACM Symp. on Theory of Computing* (New Orleans, Louisiana, 6–8 May 1991), pp. 21–31.

[7] Barak, B. How to go beyond the black-box simulation barrier. In *Proc. 42nd IEEE Symp. on Foundations of Comp. Science* (Las Vegas, Nevada, 14–17 Oct. 2001), pp. 106–115.

[8] Bellare, M., Coppersmith, D., Hastad, J., Kiwi, M. and Sudan, M. Linearity testing over characteristic two. *IEEE Transactions on Information Theory*, 42(6): 1781–1795, November 1996. Preliminary version in *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pages 432-441, Milwaukee, Wisconsin, 23-25 October 1995.

[9] Bellare, M., Goldreich, O., and Sudan, M. Free bits, PCPs, and nonapproximability—towards tight results. *SIAM Journal of Computing 27*, 3 (June 1998), 804–915. (Preliminary Version in *36th FOCS*, 1995).

[10] Bellare, M., Goldwasser, S., Lund, C., and Russell, A. Efficient probabilistically checkable proofs and applications to approximation. In *Proc. 25th ACM Symp. on Theory of Computing* (San Diego, California, 16–18 May 1993), pp. 294–304.

[11] Ben-Sasson, E., Goldreich, O., Harsha, P., Sudan, M. and Vadhan, S. Robust PCPs of Proximity, Shorter PCPs and Applications to Coding. In *STOC 2004*, Chicago.

[12] Ben-Sasson, E., Sudan, M., Vadhan, S., and Wigderson, A. Randomness-efficient low degree tests and short PCPs via epsilon-biased sets. In *Proc. 35th ACM Symp. on Theory of Computing* (San Diego, California, 9–11 June 2003), pp. 612–621.

[13] Canetti, R., Goldreich, O., and Halevi, S. The random oracle methodology, revisited. In *Proc. 30th ACM Symp. on Theory of Computing* (Dallas, Texas, 23–26 May 1998), pp. 209–218.

[14] Dinur, I., Fischer, E., Kindler, G., Raz, R., and Safra, S. PCP Characterizations of NP: Towards a Polynomially-Small Error-Probability In *Proc. 31st ACM Symp. on Theory of Computing*, 1999, pp. 29-40

[15] Dinur, I., and Reingold, O. PCP testers: Towards a more combinatorial proof of PCP theorem. FOCS 2004.

[16] Feige, U., Goldwasser, S., Lovász, L., Safra, S., and Szegedy, M. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM 43*, 2 (Mar. 1996), 268–292. (Preliminary version in *32nd FOCS*, 1991).

[17] K. Friedl, Z. Hatsagi, and A. Shen. Low-degree tests. *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 57–64, 1994.

[18] Goldreich, O. A sample of samplers – a computational perspective on sampling. Tech. Rep. TR97-020, Electronic Colloquium on Computational Complexity, 1997.O. Goldreich

[19] Goldreich, O., and Sudan, M. Locally testable codes and PCPs of almost linear length. In *Proc. 43rd IEEE Symp. on Foundations of Comp. Science* (Vancouver, Canada, 16–19 Nov. 2002), pp. 13–22. (See ECC Report TR02-050, 2002).

[20] Guruswami, V., Lewin, D., Sudan, M., and Trevisan, L. A tight characterization of NP with 3-query PCPs. In *Proc. 39th IEEE Symp. on Foundations of Comp. Science* (Palo Alto, California, 8–11 Nov. 1998), pp. 18–27.

[21] Harsha, P., and Sudan, M. Small PCPs with low query complexity. *Computational Complexity 9*, 3–4 (Dec. 2000), 157–201. (Preliminary Version in *18th STACS*, 2001).

[22] Håstad, J. Some optimal inapproximability results. *Journal of the ACM 48*, 4 (July 2001), 798–859. (Preliminary Version in *29th STOC*, 1997).

[23] Kilian, J. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *Proc. 24th ACM Symp. on Theory of Computing* (Victoria, British Columbia, Canada, 4–6 May 1992), pp. 723–732.

[24] Leighton, F. T. *Introduction to Parallel Algorithms and Architectures.* Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1992.

[25] Linnik, U. V. On the Least Prime in an Arithmetic Progression. I. The Basic Theorem. In *Mat. Sbornik N. S.* 15 (57), 139-178, 1944.

[26] Micali, S. Computationally sound proofs. *SIAM Journal of Computing 30*, 4 (2000), 1253–1298. (Preliminary Version in *35th FOCS*, 1994).

[27] Polishchuk, A., and Spielman, D. A. Nearly-linear size holographic proofs. In *Proc. 26th ACM Symp. on Theory of Computing* (Montréal, Québec, Canada, 23–25 May 1994), pp. 194–203.

[28] Raz, R. A parallel repetition theorem. *SIAM Journal of Computing 27*, 3 (June 1998), 763–803. (Preliminary Version in *27th STOC*, 1995).

[29] Rubinfeld, R., and Sudan, M. Robust characterizations of polynomials with applications to program testing. *SIAM Journal of Computing 25*, 2 (Apr. 1996), 252–271. (Preliminary Version in *23rd STOC*, 1991 and *3rd SODA*, 1992).

[30] Samorodnitsky, A., and Trevisan, L. A PCP characterization of NP with optimal amortized query complexity. In *Proc. 32nd ACM Symp. on Theory of Computing* (Portland, Oregon, 21–23 May 2000), pp. 191–199.

[31] Spielman, D. A. Computationally Efficient Error-Correcting Codes and Holographic Proofs Ph. D. Thesis, MIT, Cambridge, MA, 1995.

A Proof of Lemma 10

The lemma is an immediate corollary of the Bivariate Testing Theorem of Polischuk and Spielman [27, Theorem 9]. We use here the general version of it appearing in Spielman's Thesis [31, Theorem 4.2.19].

Theorem 25 *[31, Bivariate Testing] Let $\mathbb{F}$ be a field, $S, T \subseteq \mathbb{F}$. Let $R(x,y)$ be a polynomial over $\mathbb{F}$ of degree $(d, |T|-1)$ and let $C(x,y)$ be a polynomial over $\mathbb{F}$ of degree $(|S|-1, e)$. If*

$$\Pr_{(x,y)\in S\times T}[R(x,y) \neq C(x,y)] < \gamma^2, \qquad \text{and} \qquad 2(\frac{d}{|S|} + \frac{e}{|T|} + \gamma) < 1,$$

then there exists a polynomial $Q(x,y)$ of degree (d,e) such that

$$\Pr_{(x,y)\in S\times T}[R(x,y) \neq Q(x,y) \text{ or } C(x,y) \neq Q(x,y)] < 2\gamma^2$$

Proof (of Lemma 10): We prove the contra-positive form for $c_0 = 128$ (we don't try to optimize constants). We may assume w.l.o.g. $\delta^{(d,\star)}, \delta^{(\star,e)} < 1/c_0$, otherwise the claim is trivial. Correct each row of f to its closest RS-codeword (breaking ties arbitrarily), obtaining a bivariate polynomial $R(x,y)$ of degree $(d, |T|-1)$. By definition, $\Delta(R(x,y), f) = \delta^{(d,\star)}(f)$. Similarly, correct the columns of f to obtain the polynomial $C(x,y)$ of degree $(|S|-1, e)$ that is within fractional distance $\delta^{(\star,e)}(f)$ of f. We get

$$\Pr_{(x,y)\in S\times T}[R \neq C] \leq \delta^{(d,\star)}(f) + \delta^{(\star,e)}(f) = \gamma^2 < 1/64$$

Since $\gamma \leq 1/8, d \leq |S|/8, e \leq |T|/8$, both conditions of Theorem 25 hold, allowing us to conclude $R(x,y)$ is $(2\gamma^2)$-close to $\mathrm{RM}(\mathbb{F}, S\times T, (d,e))$. The triangle inequality completes the proof:

$$\delta^{(d,e)}(f) \leq \Delta(f,R) + \Delta(R, \mathrm{RM}(\mathbb{F}, S\times T, (d,e))) \leq 3\delta^{(d,\star)}(f) + 2\delta^{(\star,e)}(f)$$

∎

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'