SMALL PCPS WITH LOW QUERY COMPLEXITY

Prahladh Harsha and Madhu Sudan

Abstract. Most known constructions of probabilistically checkable proofs (PCPs) either blow up the proof size by a large polynomial, or have a high (though constant) query complexity. In this paper we give a transformation with slightly-super-cubic blowup in proof size, with a low query complexity. Specifically, the verifier probes the proof in 16 bits and rejects every proof of a false assertion with probability arbitrarily close to $\frac{1}{2}$, while accepting corrects proofs of theorems with probability one. The proof is obtained by revisiting known constructions and improving numerous components therein. In the process we abstract a number of new modules that may be of use in other PCP constructions.

Keywords. NP completeness, probabilistic proof systems, holographic proofs.
Subject classification. 68Q15

1. Introduction

Probabilistically checkable proofs (PCP) have played a major role in proving the hardness of approximation of various combinatorial optimization problems. Constructions of PCPs have been the subject of active research in the last ten years. In the last decade, there have been several "efficient" construction of PCPs which in turn have resulted in tighter inapproximability results. Arora *et al.* (1998) showed that it is possible to transform any proof into a probabilistically checkable one of polynomial size, such that it is verifiable with a constant number of queries. Valid proofs are accepted with probability one (this parameter is termed the completeness of the proof), while any purported proof of an invalid assertion is rejected with probability 1/2 (this parameter is the soundness of the proof). Neither the proof size, nor the query complexity is explicitly described there; however the latter is estimated to be around 10^6.

[§]A preliminary version of this appeared in *STACS 2001, 18th Annual Symposium on Theoretical Aspects of Computer Science*, Afonso Ferreira & Horst Reichel, editors, volume 2010 of *Lecture Notes in Computer Science*, 327–338. Springer-Verlag, Dresden, Germany.

Subsequently much success has been achieved in improving the parameters of PCPs, constructing highly efficient proof systems either in terms of their size or their query complexity. The best result in terms of the former is a result of Polishchuk & Spielman (1994). They show how any proof can be transformed into a probabilistically checkable proof with only a mild blowup in the proof size, of $n^{1+\epsilon}$ for arbitrarily small $\epsilon > 0$ and that is checkable with only a constant number of queries. This number of queries however is of the order of $O(1/\epsilon^2)$, with the constant hidden by the big-Oh being some multiple of the query complexity of Arora *et al.* (1998). On the other hand, Håstad (1997a) has constructed PCPs for arbitrary NP statements where the query complexity is a mere three bits (for completeness almost 1 and soundness 1/2). However the blowup in the proof size of Håstad's PCPs has an exponent proportional to the query complexity of the PCP of Arora *et al.* (1998). Thus neither of these "nearly-optimal" results provides simultaneous optimality of the two parameters. It is reasonable to wonder if this inefficiency in the combination of the two parameters is inherent; and our paper is motivated by this question.

We examine the size and query complexity of PCPs jointly and obtain a construction with reasonable performance in both parameters. The only previous work that mentions the joint size vs. query complexity of PCPs is a work of Friedl & Sudan (1995), who indicate that NP has PCPs with nearly quadratic size complexity and in which the verifier queries the proof for 165 bits. The main technical ingredient in their proof was an improved analysis of the "low-degree test". Subsequent to this work, the analysis of low-degree tests has been substantially improved. Raz & Safra (1997) and Arora & Sudan (1997a) have given highly efficient analysis of different low-degree tests. Furthermore, techniques available for "proof composition" have improved, as also have the construction for terminal "inner verifiers". In particular, the work of Håstad (1997b), has significantly strengthened the ability to analyze inner verifiers used at the final composition step of PCP constructions.

In view of these improvements, it is natural to expect the performance of PCP constructions to improve. Our work confirms this expectation. However, our work exposes an enormous number of complications in the natural path of improvement. We resolve most of these, with little loss in performance and thereby obtain the following result: Satisfiability has a PCP verifier that makes at most 16 oracle queries to a proof of size at most $n^{3+o(1)}$, where n is the size of the instance of satisfiability. Satisfiable instances have proofs that are accepted with probability one, while unsatisfiable instances are accepted with probability arbitrarily close to 1/2. (See Theorem 2.2.)

We also raise several technical questions whose positive resolution may lead to a PCP of nearly quadratic size and query complexity of 6. Surprisingly, no non-trivial limitations are known on the joint size + query complexity of PCPs. In particular, it is open as to whether nearly linear sized PCPs with query complexity of 3 exist for NP statements.

2. Overview

We first recall the standard definition of the class $\mathrm{PCP}_{c,s}[r, q]$.

DEFINITION 2.1. *For functions $r, q : \mathbb{Z}^+ \to \mathbb{Z}^+$, a probabilistic oracle machine (or verifier) V is (r, q)-restricted if on input x of length n, the verifier tosses at most $r(n)$ random coins and queries an oracle π for at most $q(n)$ bits. We use the notation $V^\pi(x; R)$ to denote the outcome of verifier V on input string x, random coins R and with oracle access to π. For $c, s \in [0, 1]$, a language $L \in PCP_{c,s}[r, q]$ if there exists an (r, q)-restricted verifier V that satisfies the following properties on input x.*

> COMPLETENESS: *If $x \in L$ then there exists π such that V on oracle access to π accepts with probability at least c. (i.e., $\exists \pi, \Pr_R[V^\pi(x; R) = \mathsf{accept}] \geq c$.)*
>
> SOUNDNESS: *If $x \notin L$ then for every oracle π, the verifier V accepts with probability strictly less than s. (i.e., $\forall \pi, \Pr_R[V^\pi(x; R) = \mathsf{accept}] < s$.)*

While our principal interest is in the size of a PCP and not in the randomness, it is well-known that the size of a probabilistically checkable proof (or more precisely, the number of distinct queries to the oracle π) is at most $2^{r(n)+q(n)}$. Thus the size is implicitly governed by the randomness and query complexity of a PCP. The main result of this paper is the following.

THEOREM 2.2. *For every $\varepsilon, \mu > 0$,*

$$\mathrm{SAT} \in PCP_{1,\frac{1}{2}+\mu}\left[(3+\varepsilon)\log n, 16\right].$$

REMARK. *Actually the constants ε and μ above can be replaced by some $o(1)$ functions; but we don't derive them explicitly.*

It follows from the parameters that the associated proof is of size at most $O(n^{3+\varepsilon})$.

Cook (1988) showed that any language in $\mathrm{NTIME}(t(n))$ could be reduced to SAT in $O(t(n) \log t(n))$ time such that instances of size n are mapped to

Boolean formulae of size at most $O(t(n) \log t(n))$. Combining this with Theorem 2.2, we have that every language in NP has a PCP with at most a slightly super-cubic blowup in proof size and a query complexity as low as 16 bits.

2.1. MIP and recursive proof composition. As pointed out earlier, the parameters we seek are such that no existing proof system achieves them. Hence we work our way through the PCP construction of Arora *et al.* (1998) and make every step as efficient as possible. The key ingredient in their construction (as well as most subsequent constructions) is the notion of recursive composition of proofs, a paradigm introduced by Arora & Safra (1998). The paradigm of recursive composition is best described in terms of multi-prover interactive proof systems (MIPs).

Definition 2.3. *For integer p, and functions $r, a : \mathbb{Z}^+ \to \mathbb{Z}^+$, an MIP verifier V is (p, r, a)-restricted if it interacts with p mutually-non-interacting provers $\pi_1, \ldots, \pi_p$ in the following restricted manner. On input x of length n, V picks a random $r(n)$-bit string R and generates p queries $q_1, \ldots, q_p$ and a circuit C of size at most $a(n)$. The verifier then issues query q_i to prover π_i. The provers respond with answers $a_1, \ldots, a_p$ each of length at most $a(n)$ and the verifier accepts x iff $C(a_1, \ldots, a_p) =$*true*. We use the notation $V^{\pi_1,\ldots,\pi_p}(x; R)$ to denote the outcome of the MIP verifier V on input string x, random string R and with oracle access to the provers $\pi_1, \ldots, \pi_p$. A language L belongs to $MIP_{c,s}[p, r, a]$ if there exists a (p, r, a)-restricted MIP verifier V such that on input x:*

> Completeness*: If $x \in L$ then there exist $\pi_1, \ldots, \pi_p$ such that V accepts with probability at least c. (i.e., $\exists \pi_1, \ldots, \pi_p, \Pr_R[V^{\pi_1,\ldots,\pi_p}(x; R) =$ accept$] \geq c$.)*
>
> Soundness*: If $x \notin L$ then for every $\pi_1, \ldots, \pi_p$, V accepts with probability less than s. (i.e., $\forall \pi_1, \ldots, \pi_p, \Pr_R[V^{\pi_1,\ldots,\pi_p}(x; R) =$ accept$] < s$.)*

It is easy to see that $\mathrm{MIP}_{c,s}[p, r, a]$ is a subclass of $\mathrm{PCP}_{c,s}[r, pa]$ and thus it is beneficial to show that SAT is contained in MIP with nice parameters. However, much stronger benefits are obtained if the containment has a small number of provers, even if the answer size complexity (a) is not very small. This is because the verifier's actions can usually be simulated by a much more efficient verification procedure, one with much smaller answer size complexity, at the cost of a few more provers. Results of this nature are termed proof composition lemmas; and the efficient simulators of the MIP verification procedure are usually called "inner verification procedures".

The next three lemmas divide the task of proving Theorem 2.2 into smaller subtasks. The first gives a starting MIP for satisfiability, with 3 provers, but poly-logarithmic answer size. We next give the composition lemma that is used in the intermediate stages. The final lemma gives our terminal composition lemma – the one that reduces answer sizes from some slowly growing function to a constant.

LEMMA 2.4. *For every $\varepsilon, \mu > 0$, there exists a polynomial p such that*

$$\mathrm{SAT} \in MIP_{1,\mu}[3, (3+\varepsilon)\log n, p(\log n)]$$

Lemma 2.4 is proven in Section 3. This lemma is critical to bounding the proof size. This lemma follows the proof of a similar one (the "parallelization" step) in Arora *et al.* (1998); however various aspects are improved. We show how to incorporate advances made by Polishchuk & Spielman (1994), and how to take advantage of the low-degree test of Raz & Safra (1997). Most importantly, we show how to save a quadratic blowup in this phase that would be incurred by a direct use of the parallelization step in Arora *et al.* (1998).

The first composition lemma we use is an off-the-shelf product due to Arora & Sudan (1997b). Similar lemmas are implicit in the works of Bellare *et al.* (1993) and Raz & Safra (1997).

LEMMA 2.5 (Arora & Sudan 1997b). *For every $\epsilon > 0$ and $p < \infty$, there exist constants c_1, c_2, c_3 such that for every $r, a : \mathbb{Z}^+ \to \mathbb{Z}^+$,*

$$MIP_{1,\epsilon}[p, r, a] \subseteq MIP_{1,\epsilon^{1/(2p+2)}}[p+3, r + c_1 \log a, c_2(\log a)^{c_3}].$$

The next lemma shows how to truncate the recursion. This lemma is proved in Section 4 using a "Fourier-analysis" based proof, as in Håstad (1997b). This is the first time that this style of analysis has been applied to MIPs with more than 2 provers. All previous analyses seem to have focused on composition with canonical 2-prover proof systems at the outer level. Our analysis reveals surprising complications (see Section 4 for details) and forces us to use a large number (seven) of extra bits to effect the truncation.

LEMMA 2.6. *For every $\epsilon > 0$ and $p < \infty$, there exists a $\gamma > 0$ such that for every $r, a : \mathbb{Z}^+ \to \mathbb{Z}^+$,*

$$MIP_{1,\gamma}[p, r, a] \subseteq PCP_{1,\frac{1}{2}+\epsilon}[r + O\left(2^{pa}\right), p+7].$$

Proof of Theorem 2.2. The proof is straightforward given the above lemmas. We first apply Lemma 2.4 to get a 3-prover MIP for SAT, then apply Lemma 2.5 twice to get a 6- and then a 9-prover MIP for SAT. The answer size in the final stage is poly $\log\log\log n$. Applying Lemma 2.6 at this stage we obtain a 16-query PCP for SAT; and the total randomness in all stages remains $(3+\varepsilon)\log n$. □

2.2. Organization of the paper. In Section 3, we prove Lemma 2.4. For this purpose, we present the Polynomial Constraint Satisfaction problem in Section 3.3 and discuss its hardness. We then discuss the Low degree Test in Section 3.5. Most aspects of the proofs in Section 3 are drawn from previous works of Arora *et al.* (1998); Arora & Sudan (1997a); Polishchuk & Spielman (1994); Raz & Safra (1997). In Section 4, we present the proof of Lemma 2.6. In Section 5 we suggest possible approaches for improvements in the joint size-query complexity of PCPs.

3. A randomness efficient MIP for SAT

In this section, we use the term "length-preserving reductions", to refer to reductions in which the length of the target instance of the reduction is nearly-linear ($O(n^\beta)$ for β arbitrarily close to 1) in the length of the source instance. More precisely, for $\beta > 1$, an β-length-preserving reduction is a reduction that runs in polynomial time and produces target instances of size at most $O(n^\beta)$.

To prove membership in SAT, we first transform SAT into an algebraic problem. This transformation comes in two phases. First we transform it to an algebraic problem (that we call AP for lack of a better name) in which the constraints can be enumerated compactly. Then we transform it to a promise problem on polynomials, called Polynomial Constraint Satisfaction (PCS), with a large associated gap. We then show how to provide an MIP verifier for the PCS problem.

Though most of these results are implicit in the literature, we find that abstracting them cleanly significantly improves the exposition of PCPs. The first problem, AP, could be proved to be NP-hard almost immediately, if one did not require length-preserving reductions. We show how the results of Polishchuk & Spielman (1994) imply a length preserving reduction from SAT to this problem. We then reduce this problem to PCS. This step mimics the sum-check protocol of Lund *et al.* (1990). The technical importance of this intermediate step is the fact that it does *not* refer to "low-degree" tests in its analysis. Low-degree tests are primitives used to test if the function described by a given oracle is close to some (unknown) multivariate polynomial of low-degree. Low-degree tests have played a central role in the constructions of PCPs. Here we separate (to a large extent) their role from other algebraic manipulations used to obtain PCPs/MIPs for SAT .

In the final step, we show how to translate the use of state-of-the-art low-degree tests, in particular the test of Raz & Safra (1997), in conjunction with the hardness of PCS to obtain a 3-prover MIP for SAT. This part follows a proof of Arora *et al.* (1998) (their parallelization step); however a direct implementation would involve $6 \log n$ randomness, or an n^6 blow up in the size of the proof. Part of this is a cubic blow up due to the use of the low-degree test and we are unable to get around this part. Direct use of the parallelization also results in a quadratic blowup of the resulting proof. We save on this by creating a variant of the parallelization step of Arora *et al.* (1998) that uses higher dimensional varieties instead of 1-dimensional ones.

3.1. A compactly described algebraic NP-hard problem.

Definition 3.1. *For functions $m, h : \mathbb{Z}^+ \to \mathbb{Z}^+$, the problem $\mathrm{AP}_{m,h}$ has as its instances $(1^n, H, T, \psi, \rho_1, \ldots, \rho_6)$ where: H is a field of size $h(n)$, $\psi : H^7 \to H$ is a constant degree polynomial, T is an arbitrary function from H^m to H and the ρ_i's are linear maps from H^m to H^m, for $m = m(n)$. (T is specified by a table of values, and ρ_i's by $m \times m$ matrices.) $(1^n, H, T, \psi, \rho_1, \ldots, \rho_6) \in \mathrm{AP}_{m,h}$ if there exists an assignment $A : H^m \to H$ such that for every $x \in H^m$, $\psi(T(x), A(\rho_1(x)), \ldots, A(\rho_6(x))) = 0$.*

The above problem is just a simple variant of standard constraint satisfaction problems, the only difference being that its variables and constraints are now indexed by elements of H^m. The only algebra in the above problem is in the fact that the functions ρ_i, which dictate which variables participate in which constraint, are linear functions. The following statement, abstracted from Polishchuk & Spielman (1994), gives the desired hardness of AP.

Lemma 3.2. *There exists a constant c such that for every $\beta > 1$ and any pair of functions $m, h : \mathbb{Z}^+ \to \mathbb{Z}^+$ satisfying $h(n)^{m(n)-c} \geq n$ and $h(n)^{m(n)} = O(n^\beta)$,* SAT *reduces to $\mathrm{AP}_{m,h}$ under β-length-preserving reductions.*

Lemma 3.2 is a reformulation of the result proved in Polishchuk & Spielman (1994) and Spielman (1995) in a manner that is convenient for us to work with. We prove this lemma in Section 3.2. We note that Szegedy (1999) has given an alternate abstraction of the result of Polishchuk & Spielman (1994); Spielman (1995). His abstraction focuses on some different aspects of the result of Polishchuk & Spielman (1994) and Spielman (1995) and does not suffice for our purposes.

3.2. Hardness of AP **problem.** The proof of Lemma 3.2 is along the lines of Polishchuk & Spielman (1994) and Spielman (1995). In the following two subsections, we (re-)present the machinery required to prove the lemma and finally provide a proof of the lemma in Section 3.2.3.

3.2.1. De Bruijn Graph Coloring Problem.

DEFINITION 3.3. *The de Bruijn graph B_n is a directed graph on 2^n vertices in which each vertex is represented by a n-bit binary string. The vertex represented by $(x_1, \dots, x_n)$ has edges pointing to the vertices represented by $(x_2, \dots, x_n, x_1)$ and $(x_2, \dots, x_n, x_1 \oplus 1)$, where $a \oplus b$ denotes the sum of a and b modulo 2.*

We then define a *wrapped de Bruijn graph* to be the product of a de Bruijn graph and a cycle.

DEFINITION 3.4. *The wrapped de Bruijn graph $\mathcal{B}_n$ is a directed graph on $5n \cdot 2^n$ vertices in which each vertex is represented by a pair consisting of an n-bit binary string and a number modulo $5n$. The vertex represented by $((x_1, \dots, x_n), a)$ has edges pointing to the vertices $((x_2, \dots, x_n, x_1), a+1)$ and $((x_2, \dots, x_n, x_1 \oplus 1), a+1)$, where the addition $a+1$ is performed modulo $5n$.*

Similarly, one can define the *extended de Bruijn graph* (on $(5n+1)\cdot 2^n$ vertices) to be the product of the de Bruijn graph (on 2^n vertices) and a line graph (on $5n+1$ vertices). For ease of notation, let us define for any vertex v, $\varrho_1(v)$ and $\varrho_2(v)$ to be the two neighbors of v in the wrapped de Bruijn graph. Polishchuk & Spielman (1994) and Spielman (1995) show how to reduce SAT to the following coloring problem on the wrapped de Bruijn graph using standard packet routing techniques (see Leighton (1992)).

DEFINITION 3.5. *The problem* DE-BRUIJN-GRAPH-COLOR *has as its instances $(\mathcal{B}_n, T)$ where $\mathcal{B}_n$ is a wrapped de Bruijn graph on $5n \cdot 2^n$ vertices and $T : V(\mathcal{B}_n) \to C_1$ is a coloring of the vertices of $\mathcal{B}_n$ (T is specified by a table of values). $(\mathcal{B}_n, T) \in$* DE-BRUIJN-GRAPH-COLOR *if there exists another coloring $A : V(\mathcal{B}_n) \to C_2$ such that for all vertices $v \in V(\mathcal{B}_n)$,*

$$\varphi(T(v), A(v), A(\varrho_1(v)), A(\varrho_2(v))) = 0$$

where C_1, C_2 are two sets of colors independent of n and $\varphi : C_1 \times C_2^3 \to \mathbb{Z}^+$ is a function independent of n.

Similar to length-preserving reductions, we can define the term "length-efficient reductions", to refer to reductions in which the length of the target instance of the reduction is at most an extra logarithmic factor off the length of the source instance (i.e., $O(n \log n)$). Spielman (1995) proves the following statement regarding the hardness of the above problem.

PROPOSITION 3.6 (Spielman 1995, Remark 4.3.3). SAT *reduces to* DE-BRUIJN-GRAPH-COLOR *under length-efficient reductions.*

3.2.2. Algebraic Description of De Bruijn Graphs. In this section, we shall give a very simple algebraic description of the de Bruijn graphs.

DEFINITION 3.7. *A Galois graph G_n is a directed graph on 2^n vertices in which each vertex is node is identified with an element of $GF(2^n)$. Let α be a generator*[1] *of $GF(2^n)$. The vertex represented by $\gamma \in GF(2^n)$ has edges pointing to the vertices represented by $\alpha\gamma$ and $\alpha\gamma + 1$.*

CLAIM 3.8. *The Galois graph G_n is isomorphic to the de Bruijn graph B_n.*

A proof of this claim can be found in (Spielman 1995, Lemma 4.3.5).

CLAIM 3.9. *Let m divide n and α be a generator of $GF(2^{n/m})$. Then the graph on*

$$\underbrace{GF(2^{n/m}) \times GF(2^{n/m}) \times \ldots \times GF(2^{n/m})}_{m \text{ times}}$$

in which the vertex represented by $(\sigma_1, \ldots, \sigma_m)$ has edges pointing to the vertices represented by

$$(\sigma_2, \ldots, \sigma_m, \alpha\sigma_1) \text{ and } (\sigma_2, \ldots, \sigma_m, \alpha\sigma_1 + 1)$$

is isomorphic to the de Bruijn graph B_n.

PROOF. By Claim 3.8, the given graph is isomorphic to the graph on binary strings of length n in which the vertex

$$(b_1, \ldots, b_{\frac{n}{m}}, b_{\frac{n}{m}+1}, \ldots, b_{2\frac{n}{m}}, \ldots, b_{(m-1)\frac{n}{m}+1}, \ldots, b_n)$$

[1] A generator of $GF(2^n)$ is an element $\alpha \in GF(2^n)$ such that $\alpha^{2^n-1} = 1$ and $\alpha^k \neq 1$ for any $1 \leq k < 2^n - 1$. Every element in $GF(2^n)$ can be represented by a unique polynomial in α of degree at most $n-1$ with coefficients from $\{0, 1\}$.

has edges pointing to the vertices given by

$$(b_{\frac{n}{m}+1}, \ldots, b_{2\frac{n}{m}}, \ldots, b_{(m-1)\frac{n}{m}+1}, \ldots, b_n, b_2, \ldots, b_{\frac{n}{m}}, b_1)$$

and

$$(b_{\frac{n}{m}+1}, \ldots, b_{2\frac{n}{m}}, \ldots, b_{(m-1)\frac{n}{m}+1}, \ldots, b_n, b_2, \ldots, b_{\frac{n}{m}}, b_1 \oplus 1)$$

Shuffling the order of b_i's, we observe that this graph is isomorphic to the graph in which the vertex represented by

$$(b_1, b_{\frac{n}{m}+1}, \ldots, b_{(m-1)\frac{n}{m}+1}, b_2, b_{\frac{n}{m}+2}, \ldots, b_{(m-1)\frac{n}{m}+2}, \ldots, b_m, b_{2m}, \ldots, b_n)$$

has edges pointed towards the vertices

$$(b_{\frac{n}{m}+1}, \ldots, b_{(m-1)\frac{n}{m}+1}, b_2, b_{\frac{n}{m}+2}, \ldots, b_{(m-1)\frac{n}{m}+2}, \ldots, b_m, b_{2m}, \ldots, b_n, b_1)$$

and

$$(b_{\frac{n}{m}+1}, \ldots, b_{(m-1)\frac{n}{m}+1}, b_2, b_{\frac{n}{m}+2}, \ldots, b_{(m-1)\frac{n}{m}+2}, \ldots, b_m, b_{2m}, \ldots, b_n, b_1 \oplus 1)$$

which is identical to the de Bruijn graph. □

Using the above result, we can now give a simple algebraic description of the extended de Bruijn graphs.

PROPOSITION 3.10. *Let m divide n and α be a generator of $H = GF(2^{n/m})$. Let $\mathcal{C} = \{1, \alpha, \ldots, \alpha^{5n}\}$ and $\mathcal{C}' = \{1, \alpha, \ldots, \alpha^{5n-1}\}$. Then the extended de Bruijn graph on $(5n+1) \cdot 2^n$ vertices is isomorphic to the graph on $H^m \times \mathcal{C}$ in which each vertex in $(x_1, \ldots, x_m, y) \in H^m \times \mathcal{C}'$ has edges pointed towards the vertices*

$$(x_2, \ldots, x_m, \alpha x_1, \alpha y)$$

and

$$(x_2, \ldots, x_m, \alpha x_1 + 1, \alpha y)$$

For ease of notation, if $v \in H^m \times \mathcal{C}$, then let $\varrho_1(v)$ and $\varrho_2(v)$ denote the two neighbors of v. Or even more generally, for any $v = (x_1, \ldots, x_m, y) \in H^{m+1}$, define

$$\varrho_1(x_1, \ldots, x_m, y) \mapsto (x_2, \ldots, x_m, \alpha x_1, \alpha y) \tag{3.11}$$

$$\varrho_2(x_1, \ldots, x_m, y) \mapsto (x_2, \ldots, x_m, \alpha x_1 + 1, \alpha y) \tag{3.12}$$

3.2.3. Proof of Lemma 3.2. Instead of showing that SAT is reducible to $\mathrm{AP}_{m,h}$, we shall show that SAT is reducible under length preserving reductions to another problem $\mathrm{AP'}_{m,h}$. It would then follow from the definition of AP and AP′ that SAT is reducible to $\mathrm{AP}_{m,h}$ under length preserving reductions.

DEFINITION 3.13. *For functions $m, h : \mathbb{Z}^+ \to \mathbb{Z}^+$, the problem $\mathrm{AP'}_{m,h}$ has as its instances $(1^n, H, T, \psi, \rho_1, \ldots, \rho_5, \rho)$ where: H is a field of size $h(n)$, $\psi : H^7 \to H$ is a constant degree polynomial, T is an arbitrary function from H^{m-1} to H, the ρ_i's are linear maps from H^m to H^{m-1} and $\rho : H^m \to H$ is a linear map for $m = m(n)$. (T is specified by a table of values, ρ_i's by $m\times(m-1)$ matrices and ρ by a $m\times 1$ matrix.) $(1^n, H, T, \psi, \rho_1, \ldots, \rho) \in \mathrm{AP'}_{m,h}$ if there exists an assignment $A : H^{m-1} \to H$ such that for every $x \in H^m$, $\psi(T(\rho_1(x)), A(\rho_1(x)), \ldots, A(\rho_5(x)), \rho(x)) = 0$.*

PROPOSITION 3.14. *For every $\beta > 1$ and any pair of functions $m, h : \mathbb{Z}^+ \to \mathbb{Z}^+$ satisfying $h(n)^{m(n)-2} \geq n$ and $h(n)^{m(n)} = O\left(n^\beta\right)$,* SAT *reduces to* $\mathrm{AP'}_{m,h}$ *under β-length-preserving reductions.*

PROOF. Let ϕ be any instance of SAT of size n. By Proposition 3.6, we have that ϕ can be reduced to an instance $(\mathcal{B}_{n'}, T)$ of DE-BRUIJN-GRAPH-COLOR . As the reduction is length-efficient, we have that $5n' \cdot 2^{n'} = O(n \log n)$ or $N \approx n$ where $N = 2^{n'}$. Let $\beta > 1$ and m, h be any two functions satisfying the requisites of Proposition 3.14. Let $m'(n) = m(n) - 2$. Let α be a generator of the field $GF(2^{n/m'})$. Now as $h(n)^{m(n)-2} \geq n$, there exists a field H of size $h(n)$ such that the field $GF(2^{n/m'})$ can be embedded in H. Now, as seen from Section 3.2.2, we can view the graph $B_{n'}$ as a graph on $H^{m'}$ and the graph $\mathcal{B}_{n'}$ as a graph on $H^{m'} \times \mathcal{C}$ where $\mathcal{C} = \{1, \alpha, \ldots, \alpha^{5n}\}$. As $\mathcal{C} \subseteq GF(2^{n/m'}) \subseteq H$, we can further view $\mathcal{B}_{n'}$ as a graph on $H^{m'+1}$, where the neighborhood functions ϱ_1, ϱ_2 are as defined in (3.11) and (3.12). We can also view the set of colors C_1 and C_2 as embedded in the field H. With such an embedding, we can consider the map $T : V(\mathcal{B}_{n'}) \to C_1$ as a map $T : H^{m'+1} \to H$.

Consider the following choice of linear transformations $\rho_i : H^m \to H^{m'+1}$ (recall $m' = m - 2$) For any $(\bar{x}, y, z) \in H^m$ where $\bar{x} \in H^{m'}, y, z \in H$

- $\rho_1 : (\bar{x}, y, z) \mapsto (\bar{x}, y)$.
- $\rho_2 : (\bar{x}, y, z) \mapsto \varrho_1(\bar{x}, y)$.
- $\rho_3 : (\bar{x}, y, z) \mapsto \varrho_2(\bar{x}, y)$.
- $\rho_4 : (\bar{x}, y, z) \mapsto (\bar{x}, 1)$.

- $\rho_5 : (\bar{x}, y, z) \mapsto (\bar{x}, \alpha^{5n})$.

Also define $\rho : H^m \to H$ such that $\rho : (\bar{x}, y, z) \mapsto z$. Note each of the ρ_i's are linear transformations. Now consider the polynomials defined as follows:

- $\varphi_1 : H^4 \to H$ satisfying $\varphi_1|_{C_1 \times C_2^3} = \varphi$. i.e., the restriction of φ_1 on the subset $C_1 \times C_2^3$ of the domain is the same as the function φ in the definition of DE-BRUIJN-GRAPH-COLOR .

- $\varphi_2 : H^2 \to H$ such that $\varphi_2(a, b) = 0$ iff $a = b$. (i.e., φ_2 checks if its two inputs are equal.)

- $\varphi_3 : H \to H$ such that φ_3 evaluates to 0 iff its input belongs to the set C_2.
  (i.e., $\varphi_3(x) = \prod_{c \in C_2}(x - c)$)

- $\varphi_4 : H \to H$ such that φ_4 evaluates to 0 iff its input belongs to the set C_1.
  (i.e., $\varphi_4(x) = \prod_{c \in C_1}(x - c)$)

Clearly, φ_i's can be defined such that they are all of constant degree where the degree depends only on the cardinality of the sets C_1 and C_2.

Now consider the polynomial $\psi : H^7 \to H$ defined as follows

$$\psi(a, b, c, d, e, f, t) = \varphi_1(a, b, c, d) + \varphi_2(e, f)t + \varphi_3(b)t^2 + \varphi_4(a)t^3$$

Note that ψ is also a constant degree polynomial. By construction of ψ, we have that
$\psi(T(\rho_1(z)), A(\rho_1(z)), A(\rho_2(z)), A(\rho_3(z)), A(\rho_4(z)), A(\rho_5(z)), \rho(z)) = 0, \forall z \in H^m$ iff the corresponding instance $(\mathcal{B}_{n'}, T) \in$ DE-BRUIJN-GRAPH-COLOR , which happens iff $\phi \in$ SAT. Note

(1) φ_1 checks if the condition φ is satisfied by vertices of the graph.

(2) φ_2 checks if the first and last column of the extended graph is the same (and hence the graph can be viewed as a wrapped graph).

(3) Finally, φ_3 and φ_4 checks if the colors assigned by the function A and T are indeed valid colors. (i.e., $T(v) \in C_1$ and $A(v) \in C_2$.)

We have thus shown that $(1^n, H, T, \psi, \rho_1, \ldots, \rho_5, \rho) \in \mathrm{AP}'_{m,h} \iff \phi \in$ SAT. Moreover the above reduction is β-length-preserving (since $h^m = O\left(n^\beta\right)$). Thus, proved. $\square$

3.3. Polynomial constraint satisfaction. We next present an instance of an algebraic constraint satisfaction problem. This differs from the previous one in that its constraints are "wider", the relationship between constraints and variables that appear in it is arbitrary (and not linear), and the hardness is not established for arbitrary assignment functions, but only for low-degree functions. All the above changes only make the problem harder, so we ought to gain something – and we gain in the gap of the hardness. The problem is shown to be hard even if the goal is only to separate satisfiable instances from instances in which only ϵ fraction of the constraints are satisfiable. We define this gap version of the problem first.

DEFINITION 3.15. *For $\epsilon : \mathbb{Z}^+ \to \mathbb{R}^+$, and $m, b, q : \mathbb{Z}^+ \to \mathbb{Z}^+$ the promise problem* $\mathrm{GapPCS}_{\epsilon,m,b,q}$ *has as instances $(1^n, d, k, s, \mathbb{F}; C_1, \ldots, C_t)$, where $d, k, s \leq b(n)$ are integers and $\mathbb{F}$ is a field of size $q(n)$ and $C_j = (A_j; x_1^{(j)}, \ldots, x_k^{(j)})$ is an algebraic constraint, given by an algebraic circuit A_j of size s on k inputs and $x_1^{(j)}, \ldots, x_k^{(j)} \in \mathbb{F}^m$, for $m = m(n)$.*

- *[Completeness] $(1^n, d, k, s, \mathbb{F}; C_1, \ldots, C_t)$ is a YES instance if there exists a polynomial $p : \mathbb{F}^m \to \mathbb{F}$ of degree at most d such that for every $j \in \{1, \ldots, t\}$, the constraint C_j is satisfied by p, i.e., $A_j(p(x_1^{(j)}), \ldots, p(x_k^{(j)})) = 0$.*

- *[Soundness] $(1^n, d, k, s, \mathbb{F}; C_1, \ldots, C_t)$ is a NO instance if for every polynomial $p : \mathbb{F}^m \to \mathbb{F}$ of degree at most d it is the case that at most $\epsilon(n) \cdot t$ of the constraints C_j are satisfied.*

LEMMA 3.16. *There exist constants c_1, c_2 such that for every $\beta > 1$ and every choice of functions ϵ, m, b, q satisfying $(b(n)/m(n))^{m(n)-c_1} \geq n$, $q(n)^{m(n)} = O\left(n^\beta\right)$ and $q(n) \geq c_2 b(n)/\epsilon(n)$,* SAT *reduces to* $\mathrm{GapPCS}_{\epsilon,m,b,q}$ *under β-length-preserving reductions.*

(The problem $\mathrm{AP}_{m,h}$ is used as an intermediate problem in the reduction. However we don't mention this in the lemma, since the choice of parameters m, h may confuse the statement further.) A proof of Lemma 3.16 can be found in Section 3.4. This proof is inspired by the sum-check protocol used in Lund *et al.* (1990), which was also used in Babai *et al.* (1991). The specific steps in our proof follow the proof in Sudan (1992).

3.4. Hardness of Polynomial Constraint Satisfaction. In this section, we prove Lemma 3.16. In order to prove the hardness of $\mathrm{GapPCS}_{\epsilon,m,b,q}$, we shall use another related problem *Polynomial Evolution* (PE) as an intermediary problem between AP and GapPCS. In Section 3.4.1, we describe the problem Polynomial Evolution and analyze its hardness. Finally, in Section 3.4.2, we prove Lemma 3.16.

3.4.1. Polynomial Evolution.

DEFINITION 3.17. *A polynomial construction rule R over a field $\mathbb{F}$ on m variables is a circuit which takes an oracle for a polynomial $p : \mathbb{F}^m \to \mathbb{F}$ and returns a new polynomial $q : \mathbb{F}^m \to \mathbb{F}$, defined by $q \triangleq R^p(x)$.*

Polynomial Evolution involves checking whether there exists a polynomial $p : \mathbb{F}^m \to \mathbb{F}$ such that when a given sequence of construction rules are composed on this polynomial, the resulting polynomial is identically zero. More formally,

DEFINITION 3.18. *For functions $b, m, q : \mathbb{Z}^+ \to \mathbb{Z}^+$, the problem $\mathrm{PE}_{m,b,q}$ has as instances $(1^n, d, \mathbb{F}; R_1, \dots, R_l)$ where $d \leq b(n)$ are integers, $\mathbb{F}$ is a finite field of size $q(n)$ and the R_i's are polynomial construction rules over $\mathbb{F}$ on m variables. $(1^n, d, \mathbb{F}; R_1, \dots, R_l) \in \mathrm{PE}_{m,b,q}$ if there exists a polynomial $p_0 : \mathbb{F}^m \to \mathbb{F}$ of degree at most d such that the sequence of polynomials p_i defined by $p_i \triangleq R^{p_{i-1}}$ for $i = 1 \dots l$ satisfies $p_l \equiv 0$ (i.e., p_l is identically zero.)*

If q^m is polynomial in the description of the instance, then clearly $\mathrm{PE}_{m,b,q} \in \mathrm{NP}$. We shall prove the following statement regarding the hardness of $\mathrm{PE}_{m,b,q}$.

LEMMA 3.19. *There exists a constant $c \in \mathbb{Z}^+$ such that for every $\beta > 1$ and functions $m, h, q : \mathbb{Z}^+ \to \mathbb{Z}^+$ satisfying $q \geq cmh$ and $q^m = O\left(h^{\beta m}\right)$, $\mathrm{AP}_{m,h}$ reduces to $\mathrm{PE}_{m,mh,q}$ under β-length-preserving reductions.*

Let $(1^n, H, T, \psi, \rho_1, \dots, \rho_6)$ be an instance of $\mathrm{AP}_{m,h}$. Suppose $\beta > 1$ and $\mathbb{F}$ be a field of size $q(n)$ where q and β satisfy the requirements of Lemma 3.19 such that $H \subseteq \mathbb{F}$. Let c be the degree of the polynomial $\psi : H^7 \to H$. (Recall that by definition of $\mathrm{AP}_{m,h}$, c is a constant.)

Any assignment $S : H^m \to H$ can be interpolated to obtain a polynomial $\hat{S} : \mathbb{F}^m \to \mathbb{F}$ of degree at most $|H|$ in each variable (and hence a total degree of at most $m|H|$) such that $\hat{S}|_{H^m} = S$. (i.e., the restriction of $\hat{S}$ to H^m coincides with the function S.) Conversely, any polynomial $\hat{S} : \mathbb{F}^m \to \mathbb{F}$ can be interpreted as an assignment from H^m to $\mathbb{F}$ by considering the function restricted to the sub-domain H^m.

Based on the instance $(1^n, H, T, \psi, \rho_1, \ldots, \rho_6)$, we will construct a sequence of $(m+1)$ polynomial construction rules which transform a polynomial p_0 to the zero polynomial iff the assignment given by $A = p_0|_{H^m}$ satisfies the instance $(1^n, H, T, \psi, \rho_1, \ldots, \rho_6)$. The first rule takes as input a polynomial $p_o : \mathbb{F}^m \to \mathbb{F}$ of degree mh and outputs a polynomial $p_1 : \mathbb{F}^m \to \mathbb{F}$ of degree cmh which is 0 on H^m iff the corresponding assignment $p_0|_{H^m}$ satisfies the instance $(1^n, H, T, \psi, \rho_1, \ldots, \rho_6)$. The remaining m rules follow the sum-check protocol of Lund *et al.* (1990) and "amplify" the zero-set of the polynomial p_1 so that the resulting polynomials are zero on larger and larger sets. The final polynomial $p_{m+1} : \mathbb{F}^m \to \mathbb{F}$ will be identically zero iff the original polynomial p_1 was zero on H^m and hence, iff $(1^n, H, T, \psi, \rho_1, \ldots, \rho_6) \in \mathrm{AP}_{m,h}$.

The first polynomial construction rule R_1 encodes the polynomial $\psi : H^7 \to H$ of constant degree c, the function $T : H^m \to H$ and the linear transformations $\rho_i : H^m \to H^m$. Let $\hat{T} : \mathbb{F}^m \to \mathbb{F}$ be interpolation of T such that the restriction coincides with the function T. Also let $\hat{\psi} : \mathbb{F}^7 \to \mathbb{F}$ be the extension of the polynomial ψ to the domain $\mathbb{F}^7$. (i.e., If $\psi : H^7 \to H$ is given by $\psi(x_1, \ldots, x_7) = \sum a_{i_1,\ldots,i_7} x_1^{i_1} \ldots x_7^{i_7}$, then $\hat{\psi} : \mathbb{F}^7 \to \mathbb{F}$ is the same polynomial $\psi(x_1, \ldots, x_7) = \sum a_{i_1,\ldots,i_7} x_1^{i_1} \ldots x_7^{i_7}$.) Note $\hat{\psi}$ is also of degree c. Also let $\hat{\rho_i} : \mathbb{F}^m \to \mathbb{F}^m$ represent the extension of the linear transformation $\rho_i : H^m \to H^m$ to the domain $\mathbb{F}^m$ (i.e., if ρ_i is the linear map given by $\bar{x} \mapsto A\bar{x}$ where $\bar{x} \in H^m$ and A is a $m \times m$ matrix with elements from H, then $\hat{\rho_i}$ is the linear map given by $\bar{x} \mapsto A\bar{x}$ where $\bar{x} \in \mathbb{F}^m$) The rule R_1 is defined as follows:

$$p_1(x_1, .., x_m) \triangleq \hat{\psi}(\hat{T}(x_1, .., x_m), p_0(\hat{\rho_1}(x_1, .., x_m)), \ldots, p_0(\hat{\rho_6}(x_1, .., x_m)))$$

When $p_0 = \hat{A}$ for some assignment $A : H^m \to H$, then for $(x_1, \ldots, x_m) \in H^m$,

$$p_1(x_1, .., x_m) = \psi(T(x_1, .., x_m), A(\rho_1(x_1, .., x_m)), \ldots, A(\rho_6(x_1, .., x_m)))$$

Thus, $p_1|_{H^m} \equiv 0$ iff the polynomial p_0 represents an assignment A that satisfies the instance $(1^n, H, T, \psi, \rho_1, \ldots, \rho_6)$. Note that if p_0 is a polynomial of degree mh, then p_1 is a polynomial of degree at most cmh where c is the degree of the polynomial ψ.

Now to the remaining rules. It is to be noted that only rule R_1 actually depends on the instance, the other rules are generic rules which follow the sum-check protocol in Lund *et al.* (1990). As mentioned earlier, these rules make the zero-set of the polynomials larger and larger.

For starters, let us first work on a univariate polynomial, $p : \mathbb{F} \to \mathbb{F}$. Let $H = \{h_1, \ldots, h_{|H|}\}$ be an enumeration of the elements in H. Consider the construction rule that works as follows:

$$q(r) \triangleq \sum_{j=1}^{|H|} p(h_j) r^j$$

Clearly, if $p(h) = 0$ for all $h \in H$, then $q \equiv 0$ on $\mathbb{F}$. Conversely, if $\exists h \in H, p(h) \neq 0$, then q is a non-zero polynomial and hence is not identically zero.

Now, for multivariate polynomials, we shall mimic the above construction. Consider the sequence of polynomials construction rules defined as follows. For $i = 1, \ldots, m$, rule R_{i+1} works as follows:

$$p_{i+1}\left(\underbrace{\longleftarrow \bar{r} \longrightarrow}_{i-1 \text{ variables}}, r_i, \underbrace{\longleftarrow \bar{x} \longrightarrow}_{m-i \text{ variables}}\right) \triangleq \sum_{j=1}^{|H|} p_i\left(\underbrace{\longleftarrow \bar{r} \longrightarrow}, h_j, \underbrace{\longleftarrow \bar{x} \longrightarrow}\right) r_i^j$$

By the same reasoning as in the univariate case, we have that

$$p_{i+1}|_{\mathbb{F}^i \times H^{m-i}} \equiv 0 \iff p_i|_{\mathbb{F}^{i-1} \times H^{m-i+1}} \equiv 0$$

Thus, $p_{m+1} \equiv 0$ iff $p_1|_{H^m} = 0$. But $p_1|_{H^m} \equiv 0$ iff $p_0|_{H^m}$ satisfies $(1^n, H, T, \psi, \rho_1, \ldots, \rho_6)$. Thus, the rules we have constructed satisfy

$$(1^n, mh, \mathbb{F}; R_1, \ldots, R_{m+1}) \in \mathrm{PE}_{m,mh,q} \iff (1^n, H, T, \psi, \rho_1, \ldots, \rho_6) \in \mathrm{AP}_{m,h}$$

Since $q^m = O((h^m)^\beta))$, the above reduction is β-length-preserving. Thus, Lemma 3.19 is proved.

We can in fact prove a stronger statement regarding the hardness of the PE instance, we have created.

PROPOSITION 3.20. *Suppose, we have an instance $(1^n, d, \mathbb{F}; R_1, \ldots, R_{m+1})$ of $\mathrm{PE}_{m,mh,q}$ constructed from an instance $(1^n, H, T, \psi, \rho_1, \ldots, \rho_6)$ of $\mathrm{AP}_{m,h}$ as mentioned above.*

> *[Completeness] If $(1^n, H, T, \psi, \rho_1, \ldots, \rho_6) \in \mathrm{AP}_{m,h}$, then there exists a polynomial $p_0 : \mathbb{F}^m \to \mathbb{F}$ of degree at most mh such that the sequence of polynomials constructed by applying the rules $R_1, \ldots, R_{m+1}$ (i.e., $p_i = R^{p_{i-1}}$ for $i = 1 \ldots m+1$) satisfy $p_{m+1} \equiv 0$. Moreover, each of the polynomials $p_1, \ldots, p_{m+1}$ are of degree at most cmh.*
>
> *[Soundness] If there exists a polynomial $p_0 : \mathbb{F}^m \to \mathbb{F}$ of degree at most mh and polynomials $p_1, \ldots, p_{m+1}$ of degree at most cmh each, such that*
>
> $$\Pr_{\bar{x} \in \mathbb{F}^m}[p_i(\bar{x}) = R^{p_{i-1}}] > \frac{(c+1)mh}{q}, i = 1, \ldots, m+1$$
>
> $$\Pr_{\bar{x} \in \mathbb{F}^m}[p_{m+1}(\bar{x}) = 0] > \frac{(c+1)mh}{q}$$

then, $(1^n, H, T, \psi, \rho_1, \ldots, \rho_6) \in \mathrm{AP}_{m,h}$.

For the proof of this proposition, we shall need Schwartz's Lemma.

LEMMA 3.21 (Schwartz 1980). *For any finite field* $\mathbb{F}$*, if* $p, q : \mathbb{F}^m \to \mathbb{F}$ *are two distinct polynomials of degree at most* d *each, then*

$$\Pr_{\bar{x} \in \mathbb{F}^m} [p(\bar{x}) = q(\bar{x})] < \frac{d}{|\mathbb{F}|}$$

PROOF OF PROPOSITION 3.20. The proof for the Completeness part of the proposition directly follows from the manner in which the rules are constructed.

For the soundness part, we note that the rule R_1 increases the degree of the polynomial by at most a factor of c and each of the other rules R_i has the effect of changing the degree with respect to the $(i-1)^{th}$ variable to at most h and not increasing the degree with respect to any of the other variables. This implies that each of the polynomials $R_i^{p_{i-1}}$ have degree at most $(c+1)mh$. By Schwartz's Lemma, it now follows that $p_i \equiv R_i^{p_{i-1}}$ for $i = 1, \ldots, m+1$ and $p_{m+1} \equiv 0$. But this implies that $p_0|_{H^m}$ satisfies $(1^n, H, T, \psi, \rho_1, \ldots, \rho_6)$. Thus, proved. $\square$

3.4.2. Hardness of Gap PCS**.** We first reduce AP to GapPCS

LEMMA 3.22. *There exists a constant* c *such that every* $\beta > 1$ *and all functions* $q, m, h, b, \epsilon : \mathbb{Z}^+ \to \mathbb{Z}^+$ *satisfying* $q(n) \geq b(n)/\epsilon(n)$, $b(n) \geq 2cm(n)h(n)$ *and* $q(n)^{m(n)+1} = O\left(h(n)^{\beta m(n)}\right)$, $\mathrm{AP}_{m,h}$ *reduces to* $\mathrm{GapPCS}_{\epsilon,m+1,b,q}$ *under* β*-length-preserving reductions.*

PROOF. Let $(1^n, H, T, \psi, \rho_1, \ldots, \rho_6)$ be any instance of $\mathrm{AP}_{m,h}$. Using the reduction in the proof of Lemma 3.19, we obtain the instance $(1^n, d, \mathbb{F}; R_1, \ldots, R_{m+1})$. From this instance, we shall build an instance $(1^n, d, k, s, \mathbb{F}; C_1, \ldots, C_t)$ of $\mathrm{GapPCS}_{\epsilon,m+1,b,q}$ as specified below.

Let c be the same constant that appears in Lemma 3.19. Let p_0 be the polynomial of degree at most mh that occurs in the proof of the statement "$(1^n, d, \mathbb{F}; R_1, \ldots, R_{m+1}) \in \mathrm{PE}_{m,b,q}$". Also let $p_1, \ldots, p_{m+1}$ be the polynomials defined by the rules $R_1, \ldots, R_{m+1}$ (i.e, $p_i = R_i^{p_{i-1}}$). Note that the p_i's are of degree at most cmh. We first bundle together the polynomials $p_0, \ldots, p_{m+1}$ into a single polynomial $p : \mathbb{F}^{m+1} \to \mathbb{F}$. Let $\{f_0, \ldots, f_{q-1}\}$ be an enumeration of the elements in $\mathbb{F}$. Let $F_{m+1} = \{f_0, \ldots, f_{m+1}\}$. For each $i = 0, \ldots, m+1$, let $\delta_i : \mathbb{F} \to \mathbb{F}$ be the unique polynomial of degree at most $m+1$ satisfying

$$\delta_i(x) = \begin{cases} 1 & \text{if } x = f_i \\ 0 & \text{if } x \in F_{m+1} - f_i \end{cases}$$

Polynomial $p : \mathbb{F}^{m+1} \to \mathbb{F}$ is defined as follows: For $(v, \bar{x}) \in \mathbb{F}^{m+1}$ where $v \in \mathbb{F}$ and $\bar{x} \in \mathbb{F}^m$,

$$p(v, \bar{x}) = \sum_{i=0}^{m+1} \delta_i(v) p_i(\bar{x})$$

Since each of the polynomials $p_0, \ldots, p_{m+1}$ is of degree at most cmh, the polynomial p is of degree at most $cmh + m \leq 2cmh \leq b$.

For each $x \in \mathbb{F}^m$, construct constraint C_x as follows:

$$C_x = \Big(p_{m+1}(x) = 0\Big) \wedge \bigwedge_{i=1}^{m+1} \left(p_i(x) = R_i^{p_{i-1}}(x)\right)$$

(This constraint is to be thought of as a constraint on the single polynomial p.)

The circuit associated with each constraint C_x checks the polynomial p at $k \approx (m+2)(h+1) \leq b$ points and has size s which is of the same order as k. Since p is of degree d which is at most b, we have constructed an instance $(1^n, d, k, s, \mathbb{F}; C_1, \ldots, C_t)$ of $\mathrm{GapPCS}_{\epsilon, m+1, b, q}$ where $d, k, s \leq b$ and $t = q^m$. It follows from Proposition 3.20, that this instance $(1^n, d, k, s, \mathbb{F}; C_1, \ldots, C_t)$ satisfies the following lemma.

Proposition 3.23. *Suppose, we have an instance $(1^n, d, k, s, \mathbb{F}; C_1, \ldots, C_t)$ of $\mathrm{GapPCS}_{\epsilon, m+1, b, q}$ constructed from an instance $(1^n, H, T, \psi, \rho_1, \ldots, \rho_6)$ of $\mathrm{AP}_{m,h}$ as mentioned above.*

- *[Completeness] If $(1^n, H, T, \psi, \rho_1, \ldots, \rho_6) \in \mathrm{AP}_{m,h}$, then there exists a polynomial $p : \mathbb{F}^{m+1} \to \mathbb{F}$ of degree at most d such that p satisfies all the constraints C_i (i.e., $A_i(p(x_1^{(i)}, \ldots, p(x_k^{(i)}) = 0$)*
- *[Soundness] If there exist polynomial $p : \mathbb{F}^{m+1} \to \mathbb{F}$ of degree at most d which satisfies at least ϵ fraction of the constraints, then $(1^n, H, T, \psi, \rho_1, \ldots, \rho_6) \in \mathrm{AP}_{m,h}$.*

The completeness part of this proposition is clear by construction. For the soundness part, it is to be noted that if at least $(c+1)mh/q$ fraction of the constraints are satisfied, then the soundness condition in Proposition 3.20 implies that $(1^n, H, T, \psi, \rho_1, \ldots, \rho_6) \in \mathrm{AP}_{m,h}$. The only observation to be made is that $\epsilon \geq b/q \geq 2cmh/q \geq (c+1)mh/q$.

This proposition completes the proof of the lemma.

□

Lemma 3.16 now follows from Lemma 3.2 and Lemma 3.22.

3.5. Low-degree tests. Using GapPCS it is easy to produce a simple probabilistically checkable proof for SAT. Given an instance of SAT, reduce it to an instance $\mathcal{I}$ of GapPCS ; and provide as proof the polynomial $p : \mathbb{F}^m \to \mathbb{F}$ as a table of values. To verify correctness a verifier first "checks" that p is close to some polynomial and then verifies that a random constraint C_j is satisfied by p. Low-degree tests are procedures designed to address the first part of this verification step – i.e., to verify that an arbitrary function $f : \mathbb{F}^m \to \mathbb{F}$ is close to some (unknown) polynomial p of degree d.

Low-degree tests have been a subject of much research in the context of program checking and PCPs. For our purposes, we need tests that have very low probability of error. Two such tests with analyses are known, one due to Raz & Safra (1997) and another due to Rubinfeld & Sudan (1996) (with low-error analysis by Arora & Sudan (1997b)) For our purposes the test of Raz and Safra is more efficient. We describe their results first and then compare its utility with the result in Arora & Sudan (1997b).

A plane in $\mathbb{F}^m$ is a collection of points parametrized by two variables. Specifically, given $a, b, c \in \mathbb{F}^m$ the plane $\wp_{a,b,c} = \{\wp_{a,b,c}(t_1, t_2) = a+t_1 b+t_2 c | t_1, t_2 \in \mathbb{F}\}$. Several parameterizations are possible for a given plane. We assume some canonical one is fixed for every plane, and thus the plane is equivalent to the set of points it contains. The low-degree test uses the fact that for any polynomial $p : \mathbb{F}^m \to \mathbb{F}$ of degree d, the function $p_\wp : \mathbb{F}^2 \to \mathbb{F}$ given by $p_\wp(t_1, t_2) = p(\wp(t_1, t_2))$ is a bivariate polynomial of degree d. The verifier tests this property for a function f by picking a random plane through $\mathbb{F}^m$ and verifying that there *exists* a bivariate polynomial that has good agreement with f restricted to this plane. The verifier expects an auxiliary oracle f_{planes} that gives such a bivariate polynomial for every plane. This motivates the test below.

Low-Degree Test (Plane-Point Test)

Input: A function $f : \mathbb{F}^m \to \mathbb{F}$ and an oracle f_{planes}, which for each plane in $\mathbb{F}^m$ gives a bivariate degree d polynomial.

1. Choose a random point in the space $x \in_R \mathbb{F}^m$.
2. Choose a random plane $\wp$ passing through x in $\mathbb{F}^m$.
3. Query f_{planes} on $\wp$ to obtain the polynomial $h_\wp$. Query f on x.
4. Accept iff the value of the polynomial $h_\wp$ at x agrees with $f(x)$.

It is clear that if f is a degree d polynomial, then there exists an oracle f_{planes} such that the above test accepts with probability 1. It is non-trivial to prove any converse and Raz & Safra (1997) give a strikingly strong converse. Below we work their statement into a form that is convenient for us.

First some more notation. Let $\mathrm{LDT}^{f,f_{\text{planes}}}(x,\wp)$ denote the outcome of the above test on oracle access to f and f_{planes}. Let $f,g:\mathbb{F}^m\to\mathbb{F}$ have agreement δ if $\Pr_{x\in\mathbb{F}^m}[f(x)=g(x)]=\delta$.

Theorem 3.24. *There exist constants c_0,c_1 such that for every positive real δ, integers m,d and field $\mathbb{F}$ satisfying $|\mathbb{F}|\geq c_0d(m/\delta)^{c_1}$, the following holds: Fix $f:\mathbb{F}^m\to\mathbb{F}$ and f_{planes}. Let $\{P_1,\ldots,P_l\}$ be the set of all m-variate polynomials of degree d that have agreement at least $\delta/2$ with the function $f:\mathbb{F}^m\to\mathbb{F}$. Then*

$$\Pr_{x,\wp}[f(x)\notin\{P_1(x),\ldots,P_l(x)\}\text{ and }\mathrm{LDT}^{f,f_{\text{planes}}}(x,\wp)=\mathsf{accept}]\leq\delta.$$

Remarks:

1. The actual theorem statement of Raz & Safra (1997) differs in a few aspects. The main difference being that the exact bound on the agreement probability described is different; and the fact that the claim may only say that if the low-degree test passes with probability greater than δ, then there exists some polynomial that agrees with f in some fraction of the points. A proof of reduction of the above theorem from the statement of Raz & Safra (1997) can be found in Section 3.5.1.

2. The cubic blowup in our proof size occurs from the oracle f_{planes} which has size cubic in the size of the oracle f. A possible way to make the proof shorter would be to use an oracle for f restricted only to lines. (i.e., an analogous line-point test to the above test) The analysis of Arora & Sudan (1997b) does apply to such a test. However they require the field size to be (at least) a fourth power of the degree; and this results in a blowup in the proof to (at least) an eighth power. Note that the above theorem only needs a linear relationship between the degree and the field size.

3.5.1. Reduction of Theorem 3.24 from Raz and Safra. The statement of Raz & Safra (1997) regarding the Plane-point low-degree test is as follows:

Theorem 3.25 (Raz & Safra 1997, Theorem 7). *There exist constants c_0, c_1, c_2 and c_3 such that for every $\delta>0$, integers m,d and field $\mathbb{F}$ satisfying $|\mathbb{F}|\geq c_0d(m/\delta)^{c_1}$, the following holds: Let $f:\mathbb{F}^m\to\mathbb{F}$ be any function. If there* *exists an oracle* f_{planes} *satisfying* $\Pr_{x,\wp}[\text{LDT}^{f,f_{\text{planes}}}(x,\wp) = \mathsf{accept}] \geq \delta$*, then there exists a polynomial* $p : \mathbb{F}^m \to \mathbb{F}$ *of degree at most* d *such that* p *and* f *agree on at least* δ^{c_2}/c_3 *fraction of the points.*

The above theorem statement of Raz & Safra (1997) relates the probability of a function f passing the low degree test with the agreement of f with some polynomial of low degree. The form of the statement which will be most convenient for us to work with is one which states that the probability of the low degree test passing on points at which f does not agree with any of the polynomials it has high agreement with is very low. By now transformations between these two forms of the low-degree test are standard (cf. Arora & Sudan (1997b); Raz & Safra (1997)). Below we follow the standard steps which go through a sequence of stronger forms culminating in Theorem 3.24.

LEMMA 3.26. *Let* c_0, c_1, c_2, c_3 *be the constants that appear in Theorem 3.25. For every positive real* δ*, integers* m, d *and field* $\mathbb{F}$ *satisfying* $|\mathbb{F}| \geq c_0 d(m/\delta)^{c_1}$*, the following holds: Fix* $f : \mathbb{F}^m \to \mathbb{F}$ *and* f_{planes}*. Let* $\{P_1, \ldots, P_l\}$ *be the set of all* m*-variate polynomials of degree* d *that have agreement at least* $\delta^{c_2}/2c_3$ *with the function* $f : \mathbb{F}^m \to \mathbb{F}$*. Then*

$$\Pr_{x,\wp}[f(x) \notin \{P_1(x), \ldots, P_l(x)\} \textit{ and } \text{LDT}^{f,f_{\text{planes}}}(x,\wp) = \text{accept}] \leq \delta.$$

PROOF. Suppose, $\Pr_{x,\wp}[f(x) \notin \{P_1(x), \ldots, P_l(x)\}$ and $\text{LDT}^{f,f_{\text{planes}}}(x,\wp) = \text{accept}] > \delta$. Let $S \subseteq \mathbb{F}^m$ be the set of all points in $\mathbb{F}^m$ at which f does not agree with any of $P_1, \ldots, P_l$. Then by our hypothesis, $f|_S$ passes the low-degree test (Plane-point test) with probability at least δ. We can now extend $f|_S$ to a function $g : \mathbb{F}^m \to \mathbb{F}$ on the entire domain $\mathbb{F}^m$ by setting the value of g at points not in S randomly. As g passes the low degree test with probability at least δ, by Theorem 3.25, we have that there exists a polynomial $P : \mathbb{F}^m \to \mathbb{F}$ of degree at most d that agrees with g on at least δ^{c_2}/c_3 fraction of the points in $\mathbb{F}^m$. The points of agreement of P with g must be concentrated in S as the value of g at points in $\mathbb{F}^m - S$ is random. Note the a random function has agreement approximately $1/|\mathbb{F}|$ with every degree d polynomial. Thus, P agrees with $f|_S$ on at least $\frac{\delta^{c_2}}{2c_3}|\mathbb{F}^m|$ points in S. As f is different from each of $P_1, \ldots, P_l$ in S, this polynomial P must be different from $P_1, \ldots, P_l$. Thus, we have a polynomial other than $P_1, \ldots, P_l$ that agrees with f on $\delta^{c_2}/2c_3$ fraction of points in $\mathbb{F}^m$. But this is a contradiction as $\{P_1, \ldots, P_l\}$ is the set of all polynomial that have at least $\delta^{c_2}/2c_3$ agreement with f. □

Now, for some more notation. Fix $f : \mathbb{F}^m \to \mathbb{F}$ and an oracle f_{planes}. Let the success probability of a point $x \in \mathbb{F}^m$ be defined as the fraction of planes $\wp$

passing through x such that the value of the polynomial $f_{\text{planes}}(\wp)$ at x agrees with $f(x)$. The success probability of a plane $\wp$ is defined to be the fraction of points x on the plane $\wp$ such that $f_{\text{planes}}(\wp)$ at x agrees with $f(x)$. Note, by this definition

$$\begin{aligned} E_{x\in\mathbb{F}^m}[\text{ Success probability of } x\ ] &= E_{\wp-\text{plane}}[\text{ Success probability of } \wp\ ] \\ &= \Pr_{x,\wp}[\mathrm{LDT}^{f,f_{\text{planes}}} = \mathsf{accept}] \end{aligned}$$

We are now ready to prove the next stronger form of Theorem 3.25.

LEMMA 3.27. *There exist constants c, c' such that for every positive real δ, integers m, d and field $\mathbb{F}$ satisfying $|\mathbb{F}| \geq cd(m/\delta)^{c'}$, the following holds: Let $f : \mathbb{F}^m \to \mathbb{F}$ be any function. If there exists a oracle f_{planes} satisfying*

$$\Pr_{x,\wp}[\mathrm{LDT}^{f,f_{\text{planes}}}(x,\wp) = \mathsf{accept}] \geq \delta$$

then there exists a polynomial $p : \mathbb{F}^m \to \mathbb{F}$ of degree at most d such that p and f agree on at least $3\delta/4$ fraction of the points.

PROOF. Let $\wp$ be a random plane. Since $E_{\wp-\text{plane}}$ [Success probability of $\wp$] is at least δ, it follows by an averaging argument that with probability at least $\delta/8$, the success probability of $\wp$ is at least $7\delta/8$. In other words, if for a random plane $\wp$, $E(\wp)$ denotes the event that there exists a bivariate polynomial $g_\wp : \mathbb{F}^2 \to \mathbb{F}$ of degree at most d that agrees with f on at least $7\delta/8$ fraction of the points on $\wp$, then

$$\Pr_{\wp}[E(\wp)] \geq \frac{\delta}{8} \tag{3.28}$$

Let c_0, c_1, c_2, c_3 be the constants that appear in Theorem 3.25. Let $P_1, \ldots, P_l$ be all the polynomials of degree at most d that agree with f on at least $\frac{1}{2c_3}\left(\frac{\delta^2}{20}\right)^{c_2}$ fraction of the points of $\mathbb{F}^m$. Note that $l \leq 4c_3\left(\frac{20}{\delta^2}\right)^{c_2}$. Define $\rho_1, \ldots, \rho_l$ such that $\rho_i = \Pr_{x\in\mathbb{F}^m}[P_i(x) = f(x)]$ (i.e., agreement of P_i and f). If we show that there exists an i such that $\rho_i \geq 3\delta/4$, we would be done. We will assume the contrary and obtain a contradiction to (3.28).

Suppose for all $i = 1, \ldots, l$, $\rho_i < 3\delta/4$. Let $\wp$ be any plane such that the event $E(\wp)$ occurs. Then, the bivariate polynomial $g_\wp$ that is described in the event $E(\wp)$ should satisfy one of the following.

CASE (*i*): $g_\wp \notin \{P_1|_\wp, \ldots, P_l|_\wp\}$. (i.e., $g_\wp$ is not the restriction of any of the P_i's to the plane $\wp$.)

CASE (ii)**:** $g_\wp \in \{P_1|_\wp, \ldots, P_l|_\wp\}$. (i.e., $g_\wp$ is the restriction of one of the P_i's to the plane $\wp$.)

In case (i), we have that $\wp$ is a plane whose success probability is at least $7\delta/8$ and moreover, on at least $7\delta/8 - ld/|\mathbb{F}|$ fraction of the points on $\wp$, the polynomial $g_\wp$ agrees with f but not with any of $P_1, \ldots, P_l$. By Lemma 3.26, if $|\mathbb{F}| \geq c_0 d(20m/\delta^2)^{c_1}$, then at most $\delta^2/20$ fraction of the points in $\mathbb{F}^m$ are such that f does not agree with $P_1, \ldots, P_l$ but the low degree test passes at that point. Thus, by an averaging argument it follows that

$$\Pr_\wp[\text{ Case } (i) \text{ occurs }] \leq \frac{\delta^2}{20(\frac{7\delta}{8} - \frac{ld}{|\mathbb{F}|})}$$

If $|\mathbb{F}| > 2^{2c_2+5}5^{c_2+1}c_3 d/3\delta^{c_2+1}$, then $|\mathbb{F}| > 40ld/3\delta$ and the above probability is less than $\delta/16$. Thus, if $\mathbb{F}$ is chosen in such a manner, the probability of case(i) happening is less than $\delta/16$.

In case (ii), for $i = 1, \ldots, l$, define the random variable γ_i to denote the fraction of points on the random plane $\wp$ at which P_i agrees with f. We have that for each i, $E_\wp[\gamma_i] = \rho_i$. An application of Chebyshev's inequality tells us that for each $i = 1, \ldots, l$,

$$\Pr_\wp\left[\gamma_i - \rho_i > \frac{\delta}{8}\right] \leq \frac{64\rho_i}{\delta^2|\mathbb{F}|^2}$$

As we have by our assumption that $\rho_i < 3\delta/4$, we have that

$$\Pr_\wp\left[\exists i, \gamma_i > \frac{7\delta}{8}\right] \leq l \times \frac{64\rho_i}{\delta^2|\mathbb{F}|^2} \leq \frac{2^{2c_2+8}5^{c_2}c_3}{|\mathbb{F}|^2\delta^{2c_2+1}}$$

If we choose $\mathbb{F}$ such that $|\mathbb{F}| \geq 2^{c_2+6}5^{c_2/2}\sqrt{c_3}/\delta^{c_2+1}$, then the above probability is less than $\delta/16$. Note that the probability on the LHS is an upper bound on the $\Pr_\wp[$ Case (ii) occurs $]$. Thus, case (ii) happens with probability less than $\delta/16$.

Let c, c' be sufficiently large constants such that $|\mathbb{F}| \geq cd(m/\delta)^{c'}$ implies the three inequalities $|\mathbb{F}| \geq c_0 d(20m/\delta^2)^{c_1}$, $|\mathbb{F}| > 2^{2c_2+5}5^{c_2+1}c_3 d/3\delta^{c_2+1}$ and $|\mathbb{F}| \geq 2^{c_2+6}5^{c_2/2}\sqrt{c_3}/\delta^{c_2+1}$. In this case we have that $\Pr_\wp[E(\wp)] = \Pr_\wp[$ Case (i) $] + \Pr_\wp[$ Case (i) $] < \delta/16 + \delta/16 = \delta/8$. This contradicts (3.28). Hence, there does exist a i such that $\rho_i \geq 3\delta/4$. Thus, for this i, the polynomial P_i and f agree on at least $3\delta/4$ fraction of the points in $\mathbb{F}^m$. $\square$

Theorem 3.24 is then obtained from Lemma 3.27 by mimicking the proof of Lemma 3.26 from Theorem 3.25.

3.6. Putting them together. As pointed out earlier a simple PCP for GapPCS can be constructed based on the low-degree test. A proof would be an oracle f representing the polynomial and the auxiliary oracle f_{planes}. The verifier performs a low-degree test on f and then picks a random constraint C_j and verifies that C_j is satisfied by the assignment f. But the naive implementation would make k queries to the oracle f and this is too many queries. The same problem was faced by Arora *et al.* (1998) who solved it by running a curve through the k points and then asking a new oracle f_{curves} to return the value of f restricted to this curve. This solution cuts down the number of queries to 3, but the analysis of correctness works only if $|\mathbb{F}| \geq kd$. In our case, this would impose an additional quadratic blowup in the proof size and we would like to avoid this. We do so by picking r-dimensional varieties (algebraic surfaces) that pass through the given k points. This cuts down the degree to $rk^{1/r}$. However some additional complications arise: The variety needs to pass through many random points, but not at the expense of too much randomness. We deal with these issues below.

A variety $\mathcal{V} : \mathbb{F}^r \to \mathbb{F}^m$ is a collection of m functions, $\mathcal{V} = \langle \mathcal{V}_1, \ldots, \mathcal{V}_m \rangle$, $\mathcal{V}_i : \mathbb{F}^r \to \mathbb{F}$. A variety is of degree D if all the functions $\mathcal{V}_1, \ldots, \mathcal{V}_m$ are polynomials of degree D. For a variety $\mathcal{V}$ and function $f : \mathbb{F}^m \to \mathbb{F}$, the restriction of f to $\mathcal{V}$ is the function $f|_{\mathcal{V}} : \mathbb{F}^r \to \mathbb{F}$ given by $f|_{\mathcal{V}}(a_1, \ldots, a_r) = f(\mathcal{V}(a_1, \ldots, a_r))$. Note that the restriction of a degree d polynomial $p : \mathbb{F}^m \to \mathbb{F}$ to an r-dimensional variety $\mathcal{V}$ of degree D is an r-variate polynomial of degree Dd.

Let $S \subseteq \mathbb{F}$ be of cardinality $k^{1/r}$. Let $z_1, \ldots, z_k$ be some canonical ordering of the points in S^r. Let $\mathcal{V}^{(0)}_{S,x_1,\ldots,x_k} : \mathbb{F}^r \to \mathbb{F}^m$ denote a canonical variety of degree $r|S|$ that satisfies $\mathcal{V}^{(0)}_{S,x_1,\ldots,x_k}(z_i) = x_i$ for every $i \in \{1, \ldots, k\}$. Let $Z_S : \mathbb{F}^r \to \mathbb{F}$ be the function given by $Z_S(y_1, \ldots, y_r) = \prod_{i=1}^r \prod_{a \in S}(y_i - a)$; i.e. $Z_S(z_i) = 0$. Let $\alpha = \langle \alpha_1, \ldots, \alpha_m \rangle \in \mathbb{F}^m$. Let $\mathcal{V}^{(1)}_{S,\alpha}$ be the variety $\langle \alpha_1 Z_S, \ldots, \alpha_m Z_S \rangle$. We will let $\mathcal{V}_{S,\alpha,x_1,\ldots,x_k}$ be the variety $\mathcal{V}^{(0)}_{S,x_1,\ldots,x_k} + \mathcal{V}^{(1)}_{S,\alpha}$. Note that if α is chosen at random, $\mathcal{V}_{S,\alpha,x_1,\ldots,x_k}(z_i) = x_i$ for $z_i \in S^r$ and $\mathcal{V}_{S,\alpha,x_1,\ldots,x_k}(z)$ is distributed uniformly over $\mathbb{F}^m$ if $z \in (\mathbb{F} - S)^r$. These varieties will replace the role of the curves of Arora *et al.* (1998). We note that Dinur et al. also use higher dimensional varieties in the proof of PCP-related theorems (Dinur *et al.* 1999). (They call these structures manifolds instead of varieties.) Their use of varieties is for purposes quite different from ours.

We are now ready to describe the MIP verifier for $\text{GapPCS}_{\epsilon,m,b,q}$. (Henceforth, we shall assume that t, the number of constraints in $\text{GapPCS}_{\epsilon,m,b,q}$ instance is at most q^{2m}. In fact, for our reduction from SAT (Lemma 3.16), t is exactly equal to q^m.)

MIP VERIFIER$^{f,f_{\text{planes}},f_{\text{varieties}}}(1^n, d, k, s, \mathbb{F}; C_1, \ldots, C_t)$.

Notation: r is a parameter to be specified. Let $S \subseteq \mathbb{F}$ be such that $|S| = k^{1/r}$.

1. Pick $a, b, c \in \mathbb{F}^m$ and $z \in (\mathbb{F} - S)^r$ at random.
2. Let $\wp = \wp_{a,b,c}$. Use b, c to compute $j \in \{1, \ldots, t\}$ at random (i.e., j is fixed given b, c, but is distributed uniformly when b and c are random.) Compute α such that $\mathcal{V}(z) = a$ for $\mathcal{V} = \mathcal{V}_{S,\alpha,x_1^{(j)},\ldots,x_k^{(j)}}$.
3. Query $f(a)$, $f_{\text{planes}}(\wp)$ and $f_{\text{varieties}}(\mathcal{V})$. Let $g = f_{\text{planes}}(\wp)$ and $h = f_{\text{varieties}}(\mathcal{V})$.
4. Accept if all the conditions below are true:
   (a) g and f agree at a.
   (b) h and f agree at a.
   (c) A_j accepts the inputs $h(z_1), \ldots, h(z_k)$.

Complexity: Clearly the verifier V makes exactly 3 queries. Also, exactly $3m \log q + r \log q$ random bits are used by the verifier. The answer sizes are no more than $O((drk^{1/r} + r)^r \log q)$ bits.

Now to prove the correctness of the verifier. Clearly, if the input instance is a YES instance then there exists a polynomial P of degree d that satisfies all the constraints of the input instance. Choosing $f = P$ and constructing f_{planes} and $f_{\text{varieties}}$ to be restrictions of P to the respective planes and varieties, we notice that the MIP verifier accepts with probability one.

To prove the soundness of the verifier, we first need to bound the number of polynomials of degree d that have a fairly large agreement with a function $f : \mathbb{F}^m \to \mathbb{F}$.

CLAIM 3.29. *Let $f : \mathbb{F}^m \to \mathbb{F}$ be any function. Suppose integer $d > 0$ and fraction δ are such that $\delta > 2\sqrt{\frac{d}{q}}$ where $q = |\mathbb{F}|$. Then there are at most $\frac{2}{\delta}$ polynomials of degree d that have agreement at least δ with f.*

A proof of this claim can be found in (Arora & Sudan 1997b, Proposition 7). We are now ready to bound the soundness of the verifier.

CLAIM 3.30. *Let δ be any constant that satisfies the conditions of Theorem 3.24 and $\delta \geq 4\sqrt{\frac{d}{q}}$ where $q = |\mathbb{F}|$. Then the soundness of the MIP Verifier is at most*

$$\delta + \frac{4\epsilon}{\delta} + \frac{4rk^{\frac{1}{r}}d}{\delta(q - k^{\frac{1}{r}})}.$$

PROOF. Let $P_1, \dots, P_l$ be all the polynomials of degree d that have agreement at least $\delta/2$ with f. Note that as $\delta/2 \geq 2\sqrt{d/q}$, we have from Claim 3.29 that $l \leq 4/\delta$. Now suppose, the MIP Verifier had accepted a NO instance. Then one of the following events must have taken place.

EVENT 1: $f(a) \notin \{P_1(a), \dots, P_l(a)\}$ and $\mathrm{LDT}^{f, f_{\mathrm{planes}}}(a, \wp) = \text{accept}$.
We have from Theorem 3.24, that Event 1 could have happened with probability at most δ.

EVENT 2: There exists an $i \in \{1, \dots, l\}$, such that constraint C_j is satisfiable with respect to polynomial P_i. (i.e., $A_j(P_i(x_1^{(j)}), \dots, P_i(x_k^{(j)})) = 0$).
As the input instance is a NO instance of $\mathrm{GapPCS}_{\epsilon,m,b,q}$, this events happens with probability at most $l\epsilon \leq 4\epsilon/\delta$.

EVENT 3: For all $i \in \{1, \dots, l\}$, $P_i|_{\mathcal{V}} \neq h$, but the value of h at a is contained in $\{P_1(a), \dots, P_l(a)\}$.
To bound the probability of this event happening, we reinterpret the randomness of the MIP verifier. First pick $b, c, \alpha \in \mathbb{F}^m$. From this we generate the constraint C_j and this defines the variety $\mathcal{V} = \mathcal{V}_{S,\alpha,x_1^{(j)},\dots,x_k^{(j)}}$. Now we pick $z \in (\mathbb{F} - S)^r$ at random and this defines $a = \mathcal{V}(z)$. We can bound the probability of the event in consideration after we have chosen $\mathcal{V}$, as purely a function of the random variable z as follows. Fix any i and $\mathcal{V}$ such that $P_i|_{\mathcal{V}} \neq h$. Note that the value of h at a equals $h(z)$ (by definition. of a, z and $\mathcal{V}$). Further $P_i(a) = P_i|_{\mathcal{V}}(z)$. But z is chosen at random from $(\mathbb{F} - S)^r$. By Schwartz's lemma (Lemma 3.21), the probability of agreement on this domain is at most $rk^{1/r}d/(|\mathbb{F}| - |S|)$. Using the union bound over the i's we get that this event happens with probability at most $lrk^{1/r}d/(|\mathbb{F}| - |S|) \leq 4rk^{\frac{1}{r}}d/\delta(q - k^{\frac{1}{r}})$.

We thus have that the probability of one of the above events occurring is at most $\delta + 4\epsilon/\delta + 4rk^{\frac{1}{r}}d/\delta(q - k^{\frac{1}{r}})$.

We would be done if we show that if none of the three events occur, then the MIP verifier rejects. Suppose none of the three events took place. In other words, all the following happened

- $f(a) \in \{P_1(a), \dots, P_l(a)\}$ or $\mathrm{LDT}^{f, f_{\mathrm{planes}}}(a, \wp) = \text{reject}$. We could as well assume that $f(a) \in \{P_1(a), \dots, P_l(a)\}$ for in the other case (i.e., LDT rejects), the verifier also rejects.

- For all i, $A_j\big(P_i(x_1^{(j)}, \dots, P_i(x_k^{(j)})\big) \neq 0$.

- $\exists i$, $P_i|_{\mathcal{V}} = h$ or the value of h at a is not contained in $\{P_1(a), \ldots, P_l(a)\}$.

If h at a is not one of $P_1(a), \ldots, P_l(a)$, then the MIP verifier rejects as $f(a) \in \{P_1(a), \ldots, P_l(a)\}$. So, if the MIP verifier had accepted, it should be the case that $\exists i$, $P_i|_{\mathcal{V}} = h$. But as $\forall i$, $A_j(P_i(x_1^{(j)}, \ldots, P_i(x_k^{(j)}) \neq 0$, the verifier is bound to reject in this case too. Thus, if none of the the three events occurred, then the verifier should have rejected. □

We can now complete the construction of a 3-prover MIP for SAT and give the proof of Lemma 2.4.

Proof of Lemma 2.4. Choose $\delta = \frac{\mu}{3}$. Let c_0, c_1 be the constants that appear in Theorem 3.24. Choose $\varepsilon' = \varepsilon/2$ where ε is the soundness of the MIP, we wish to prove. Choose $\epsilon = \min\{\delta\mu/12, \varepsilon'/3(9+c_1), (5+c_1)/4\}$. Let n be the size of the SAT instance. Let $m = \epsilon \log n / \log\log n$, $b = (\log n)^{3+\frac{1}{\epsilon}}$ and $q = (\log n)^{9+c_1+\frac{1}{\epsilon}}$. Note that this choice of parameters satisfies the requirements of Lemma 3.16 for $\beta = 1 + (9 + c_1)\epsilon \leq (1 + \varepsilon'/3)$. Hence, SAT reduces to $\mathrm{GapPCS}_{\epsilon,m,b,q}$ under $(1 + \varepsilon'/3)$-length-preserving reductions. Combining this reduction with the MIP verifier for GapPCS, we have a MIP verifier for SAT. Also δ satisfies the requirements of Claim 3.30. Thus, this MIP verifier has soundness as given by Claim 3.30. Recall that $k, d \leq b(n) = (\log n)^{3+\frac{1}{\epsilon}}$ from the definition of $\mathrm{GapPCS}_{\epsilon,m,b,q}$. Setting $r = \frac{1}{\epsilon}$, we have that for sufficiently large n,

$$4rk^{\frac{1}{r}}d/\delta(q - k^{\frac{1}{r}}) \leq 8rk^{\frac{1}{r}}d/q\delta \leq 8b^{1+\epsilon}/q\delta\epsilon \leq 8/\delta\epsilon(\log n)^{5+c_1-3\epsilon} \leq \mu/3$$

Hence, the soundness of the MIP verifier is at most $\delta + 4\epsilon/\delta + \mu/3 \leq \mu$. The randomness used is exactly $3m\log q + r\log q$ which with the present choice of parameters is $(3 + \varepsilon')\log n + \mathrm{poly}\log n \leq (3 + \varepsilon)\log n$. The answer size is $O((brb^{1/r} + r)^r)\log q)$ bits which for our choice of parameters is $O((9 + c_1 + \frac{1}{\epsilon})\left(\frac{1}{\epsilon}\right)^{\frac{1}{\epsilon}} \log^{2/\epsilon^3} n)$ (i.e., $\mathrm{poly}\log n$). Thus, $\mathrm{SAT} \in \mathrm{MIP}_{1,\mu}[(3+\varepsilon)\log n, \mathrm{poly}\log n]$. □

4. Constant query inner verifier for MIPs

In this section, we truncate the recursion by constructing a constant query "inner verifier" for a p-prover interactive proof system. An inner verifier is a subroutine designed to simplify the task of an MIP verifier. Say an MIP verifier V_{out}, on input x and random string R, generated queries $q_1, \ldots, q_p$ and a linear sized circuit C. In the standard protocol the verifier would send query q_i to prover Π_i and receive some answer a_i. The verifier accepts if $C(a_1, \dots, a_p) =$ true. An inner verifier reduces the answer size complexity of this protocol by accessing oracles $A_1, \dots, A_p$, which are supposedly encodings of the responses $a_1, \dots, a_p$, and an auxiliary oracle B, and probabilistically verifying that the A_i's really correspond to some commitment to strings $a_1, \dots, a_p$ that satisfy the circuit C. The hope is to get the inner verifier to do all this with very few queries to the oracles $A_1, \dots, A_p$ and B and we do so with one (bit) query each to the A_i's and seven queries to B. For encoding the responses $a_1, \dots, a_p$, we use the *long code* of Bellare *et al.* (1998). We then adapt the techniques of Håstad (1996, 1997b) to develop and analyze a protocol for the inner verifier.

4.1. Long Code. In this section, we represent all Boolean values by $\{-1, 1\}$, with -1 representing true and 1 representing false. This is done so that the Boolean xor operation becomes integer multiplication. For any finite set $\mathcal{U}$, let $\mathcal{F}_{\mathcal{U}}$ denote the set of all functions $f : \mathcal{U} \to \{-1, 1\}$. The long code of a string $a \in \mathcal{U}$ is the string $E_a^{\mathcal{U}}$ of length $2^{|\mathcal{U}|}$, whose entries are indexed by the functions $f \in \mathcal{F}_{\mathcal{U}}$, such that $E_a^{\mathcal{U}}(f) = f(a)$. For indexing purposes, a fixed (but arbitrary) ordering of the functions in $\mathcal{F}_{\mathcal{U}}$ is used. With this association in mind, we use the words "function", "string", "table" and "oracle" interchangeably. We say that a string A indexed by functions $f \in \mathcal{F}_{\mathcal{U}}$ is *folded* if $A(f) = -A(-f)$ for every $f \in \mathcal{F}_{\mathcal{U}}$. Long codes are folded. We shall assume that all strings are folded. This can be done if we employ the following access mechanism suggested in Bellare *et al.* (1998). Let u be some fixed (but arbitrary) string in the set $\mathcal{U}$. Now $\mathcal{F}_{\mathcal{U}}$ can be divided into 2 sets $\mathcal{F}_{\mathcal{U}}^1$ and $\mathcal{F}_{\mathcal{U}}^2$ as follows:

$$
\begin{aligned}
\mathcal{F}_{\mathcal{U}}^1 &= \{f \in \mathcal{F}_{\mathcal{U}} | f(u) = 1\} \\
\mathcal{F}_{\mathcal{U}}^2 &= \{f \in \mathcal{F}_{\mathcal{U}} | f(u) = -1\}
\end{aligned}
$$

Note that $\mathcal{F}_{\mathcal{U}}$ is the disjoint union of $\mathcal{F}_{\mathcal{U}}^1$ and $\mathcal{F}_{\mathcal{U}}^2$. These sets satisfy the nice property that for any function $f \in \mathcal{F}_{\mathcal{U}}$ either $f \in \mathcal{F}_{\mathcal{U}}^1$ or $-f \in \mathcal{F}_{\mathcal{U}}^1$ but not both. Given any string A which is the truth-table of a function $A : \mathcal{F}_{\mathcal{U}} \to \{-1, 1\}$, to find the value of $A(f)$ for any $f \in \mathcal{F}_{\mathcal{U}}$, we do the following. If $f \in \mathcal{F}_{\mathcal{U}}^1$, then we lookup $A(f)$ in the string A. Otherwise, we lookup $A(-f)$ and infer the value of $A(f)$ by negating $A(-f)$. (i.e., $A(f) = -A(-f)$)

In what follows, we will be using the long code to encode members of two sets $\mathcal{A}$ and $\mathcal{B}$ (defined below). In practice, to use a long code $f = E_a^{\mathcal{U}}$ in a protocol, we should be able to generate a random function in $\mathcal{F}_{\mathcal{U}}$ (i.e., in $\mathcal{F}_{\mathcal{A}}$ and $\mathcal{F}_{\mathcal{B}}$ in our case). For this purpose, we assume that the set $\mathcal{U}$ is a subset of $\{0, 1\}^l$ for some l and that the elements of $\mathcal{U}$ can be enumerated in time exponential in l. The sets $\mathcal{A}$ and $\mathcal{B}$ that we would be using in the protocol will satisfy these properties.

4.2. Details of the Inner Verifier. We now return to the description of our inner verifier. We start with some notation. Let $\mathcal{A} = \{+1, -1\}^a$ and $\mathcal{B} = \{(a_1, \dots, a_p) | C(a_1, \dots, a_p) = -1\}$. Let π_i be the projection function $\pi_i : \mathcal{B} \to \mathcal{A}$ which maps $(a_1, \dots, a_p)$ to a_i. By abuse of notation, for $\beta \subseteq \mathcal{B}$, let $\pi_i(\beta)$ denote $\{\pi_i(x) | x \in \beta\}$. Queries to the oracle A_i will be functions $f \in \mathcal{F}_{\mathcal{A}}$. Queries to the oracle B will be functions $g \in \mathcal{F}_{\mathcal{B}}$. The inner verifier expects the oracles to provide the long codes of the strings $a_1, \dots, a_p$, i.e., $A_i = E^{\mathcal{A}}_{a_i}$ and $B = E^{\mathcal{B}}_{a_1,\dots,a_p}$. Of course, we can not assume these properties; they need to be verified explicitly by the inner verifier. We will however assume that the encodings are folded. We are now ready to specify the inner verifier.

$V_{\text{inner}}{}^{A_1,\dots,A_p,B}(\mathcal{A}, \mathcal{B}, \pi_1, \dots, \pi_p)$.

1. For each each $i \in \{1, \dots, p\}$, choose $f_i \in \mathcal{F}_{\mathcal{A}}$ at random.
2. Choose $f, g_1, g_2, h_1, h_2 \in \mathcal{F}_{\mathcal{B}}$ at random and independently.
3. Let $g = f\,(g_1 \wedge g_2)\,(\prod f_i \circ \pi_i))$ and $h = f\,(h_1 \wedge h_2)\,(\prod f_i \circ \pi_i))$.
4. Read the following bits from the oracles $A_1, \dots, A_p, B$

   $y_i = A_i(f_i)$, for each $i \in \{1, \dots, p\}$.
   $w = B(f)$.
   $u_1 = B(g_1); u_2 = B(g_2)$
   $v_1 = B(h_1); v_2 = B(h_2)$
   $z_1 = B(g); z_2 = B(h)$
5. Accept iff
$$w \prod_{i=1}^{p} y_i = (u_1 \wedge u_2) z_1 = (v_1 \wedge v_2) z_2$$

4.3. Analysis of Inner Verifier. Suppose the strings $a_1, \dots, a_p$ are such that $C(a_1, \dots, a_p) = -1$. Let the tables A_i be the long codes of the strings $a_1, \dots, a_p \in \mathcal{A}$ for each $i = 1, \dots, p$. Also let table B be the long code of $(a_1, \dots, a_p) \in \mathcal{B}$. It is clear that the inner verifier V_{inner} accepts these tables with probability 1. This proves the completeness part of V_{inner}. We shall prove the soundness of V_{inner} by showing that if the acceptance probability of the inner verifier is sufficiently high then the tables $A_1, \dots, A_p$ are non-trivially close to the encoding of strings $a_1, \dots, a_p$ that satisfy $C(a_1, \dots, a_p) = -1$. For this purpose, we would need some machinery from Fourier analysis.

4.3.1. Fourier Transforms. In this section, we present linear functions, Fourier transforms as introduced by Håstad (1996, 1997b). These are tools that come handy in the analysis of long codes. A function $A : \mathcal{F}_{\mathcal{U}} \to \{-1, 1\}$ is said to be *linear* iff $A(f)A(g) = A(fg)$ for all $f, g \in \mathcal{F}_{\mathcal{U}}$. There are $2^{|\mathcal{U}|}$ linear functions, one corresponding to each set $\alpha \subseteq \mathcal{U}$, defined as follows.

$$\chi_\alpha(f) = \prod_{a \in \alpha} f(a)$$

(By convention, a product ranging over an empty set is 1.) At this point it is worthwhile noting that if the string A is the the long code of $a \in \mathcal{U}$, i.e., $A = E_a^{\mathcal{U}}$, then $A = \chi_{\{a\}}$. In other words, long codes are precisely the linear functions corresponding to singleton sets.

The function $A : \mathcal{F}_{\mathcal{U}} \to \{-1, 1\}$ can be viewed as a real-valued function $A : \mathcal{F}_{\mathcal{U}} \to \mathbb{R}$. The set of all real-valued functions of the form $A : \mathcal{F}_{\mathcal{U}} \to \mathbb{R}$ form a vector space (over the reals) of dimension $2^{|\mathcal{U}|}$. We could define the following inner product between functions A, A' in this space.

$$\langle A, A' \rangle \frac{1}{2^{|\mathcal{U}|}} \sum_{f \in \mathcal{F}_{\mathcal{U}}} A(f)A'(f) = E_{f \in \mathcal{F}_{\mathcal{U}}} [A(f)A'(f)]$$

The set of linear functions, i.e., the set $\{\chi_\alpha : \alpha \subseteq \mathcal{U}\}$, form a complete orthonormal basis for this space under the above inner product. Thus any function $A : \mathcal{F}_{\mathcal{A}} \to \mathbb{R}$ in this space has the following *Fourier expansion.*

$$A(f) = \sum_{\alpha \subseteq \mathcal{U}} \hat{A}_\alpha \chi_\alpha(f)$$

where $\hat{A}_\alpha = \langle A, \chi_\alpha \rangle$ is the *Fourier coefficient* of A with respect to α. Parseval's identity tells us that $\langle A, A \rangle = \sum_\alpha \hat{A}_\alpha^2$. Thus, for every function $A : \mathcal{F}_{\mathcal{U}} \to \{-1, 1\}$, we have that $\sum_\alpha \hat{A}_\alpha^2 = 1$.

For working with Fourier coefficients and linear functions, the following three standard properties come pretty handy.

$$\chi_\alpha(f)\chi_\alpha(g) = \chi_\alpha(fg) \tag{4.1}$$

$$\chi_\alpha(f)\chi_{\alpha'}(f) = \chi_{\alpha \triangle \alpha'}(f) \tag{4.2}$$

$$E_{f \in \mathcal{F}_{\mathcal{U}}} [\chi_\alpha(f)] = \begin{cases} 1 & \text{if } \alpha = \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

where $\alpha \triangle \alpha'$ represents the symmetric difference of the sets α and α' which is the set of elements contained in one of the sets α, α' but not both. (Proofs of these properties can be found in Håstad (1997b))

If a function $A : \mathcal{F}_{\mathcal{U}} \to \{-1, 1\}$ is folded, then $\hat{A}_\alpha = 0$ for every $\alpha \subseteq \mathcal{U}$ such that $|\alpha|$ is even. (A proof of this fact can be found in Håstad (1997b).) In particular, $\hat{A}_\alpha = 0$, if $\alpha = \emptyset$. We would like to mention here that the only property of folding that is usually used is $\hat{A}_\emptyset = 0$, but in this paper we would be making essential use of the fact that $\hat{A}_\alpha = 0$ for all α such that $|\alpha|$ is even.

4.3.2. Soundness of Inner verifier. In what follows, we let $\hat{A}_{i,\alpha}$ denote the Fourier coefficient of the table A_i with respect to the set α. The following lemma lays out the precise soundness condition in terms of the Fourier coefficients of the oracles $A_1, \ldots, A_p$.

Claim 4.4. *For every $\epsilon > 0$, there exists a $\delta > 0$ such that if the inner verifier ${V_{\text{inner}}}^{A_1,\ldots,A_p,B}(\mathcal{A}, \mathcal{B}, \pi_1, \ldots, \pi_p)$ accepts with probability at least $\frac{1}{2} + \epsilon$, then there exist $a_1, \ldots, a_p \in \mathcal{A}$ such that $C(a_1, \ldots, a_p) = -1$ and $|\hat{A}_{i,\{a_i\}}| \geq \delta$ for every $i \in \{1, \ldots, p\}$.*

Proof. Let δ be some constant dependent on ϵ(to be decided later.) Assume that there do not exist $a_1, \ldots, a_p \in \mathcal{A}$ such that $C(a_1, \ldots, a_p) = -1$ and $|\hat{A}_{i,\{a_i\}}| \geq \delta$ for every $i \in \{1, \ldots, p\}$. On restating this assumption, we get that for every $\beta \subseteq \mathcal{B}$ such that $|\beta| = 1$, there exists a $i \in \{1, \ldots, p\}$ such that $|\hat{A}_{i,\pi_i(\beta)}| < \delta$. To prove the lemma, it is sufficient if we show that for every choice of ϵ there exists a particular choice of δ, such that this assumption implies that the acceptance probability of V_{inner} is less than $\frac{1}{2} + \epsilon$.

Let ACC be the indicator random variable denoting the acceptance condition of the inner verifier. Hence, $E[ACC]$ denotes the acceptance probability of the inner verifier. We shall divide the task of proving this claim into several phases, given by Sub-Claims 4.5—4.10.

Sub-Claim 4.5.

$$E[ACC] = \frac{1}{4} + \frac{1}{2}T_1 + \frac{1}{4}T_2$$

where

$$T_1 = E\left[w\left(u_1 \wedge u_2\right) z_1 \prod_{i=1}^{p} y_i\right]; \quad T_2 = E\left[\left(u_1 \wedge u_2\right)\left(v_1 \wedge v_2\right) z_1 z_2\right]$$

Proof of Sub-Claim 4.5. The acceptance condition of the verifier V_{inner} is given by the following expression.

$$ACC = \frac{1}{4}\left(1 + w(u_1 \wedge u_2) z_1 \prod_{i=1}^{p} y_i\right)\left(1 + w(v_1 \wedge v_2) z_2 \prod_{i=1}^{p} y_i\right)$$

The acceptance probability of V_{inner} is thus exactly equal to the following expression.

$$E[ACC] = E\left[\frac{1}{4}\left(1 + w\,(u_1 \wedge u_2)\,z_1 \prod_{i=1}^{p} y_i\right)\left(1 + w\,(v_1 \wedge v_2)\,z_2 \prod_{i=1}^{p} y_i\right)\right]$$

where the expectation is taken over the random choices of the functions f_i, f, g_1, g_2, h_1 and h_2. By linearity of expectation, this simplifies to

$$\begin{aligned} E[ACC] = \frac{1}{4} + \frac{1}{4}E\left[w\,(u_1 \wedge u_2)\,z_1 \prod_{i=1}^{p} y_i\right] + \frac{1}{4}E\left[w\,(v_1 \wedge v_2)\,z_2 \prod_{i=1}^{p} y_i\right] \\ + \frac{1}{4}E\Big[(u_1 \wedge u_2)\,(v_1 \wedge v_2)\,z_1 z_2\Big] \end{aligned}$$

Recall that $y_i = A_i(f_i)$, $w = B(f)$, $u_1 = B(g_1)$, $u_2 = B(g_2)$, $v_1 = B(h_1)$, $v_2 = B(h_2)$, $z_1 = B(g)$, $z_2 = B(h)$. We thus note that

$$E\left[w\,(u_1 \wedge u_2)\,z_1 \prod_{i=1}^{p} y_i\right] = E\left[w\,(v_1 \wedge v_2)\,z_2 \prod_{i=1}^{p} y_i\right]$$

Thus, the acceptance probability is given as follows.

$$E[ACC] = \frac{1}{4} + \frac{1}{2}E\left[w\,(u_1 \wedge u_2)\,z_1 \prod_{i=1}^{p} y_i\right] + \frac{1}{4}E\Big[(u_1 \wedge u_2)\,(v_1 \wedge v_2)\,z_1 z_2\Big]$$

□

We shall now simplify each of the terms T_1 and T_2 individually and obtain the following bounds for T_1 and T_2.

Sub-Claim 4.6.

$$T_1 \;\leq\; \frac{1}{2}\sum_{\beta \subseteq \mathcal{B}} \hat{B}_\beta^2 \prod_{i=1}^{p} |\hat{A}_{i,\pi_i(\beta)}| \left(\frac{1}{2}\right)^{|\beta|} \left(1 + \sum_{\beta_1 \subseteq \beta} |\hat{B}_{\beta_1}|\right)^2$$

SUB-CLAIM 4.7.

$$T_2 \;\leq\; \frac{1}{4}\sum_{\beta\subseteq\mathcal{B}} \hat{B}_\beta^2 \left(\frac{1}{4}\right)^{|\beta|}\left(1+\sum_{\beta_1\subseteq\beta}|\hat{B}_{\beta_1}|\right)^4$$

PROOF OF SUB-CLAIM 4.6. Using the fact that $a\wedge b=(1+a+b-ab)/2$, we expand T_1 as follows:

$$T_1 \;=\; \frac{1}{2}E\left[wz_1\prod_{i=1}^{p}y_i\right]+E\left[wu_1z_1\prod_{i=1}^{p}y_i\right]-\frac{1}{2}E\left[wu_1u_2z_1\prod_{i=1}^{p}y_i\right]$$

The expression for T_1 is of the form $\frac{1}{2}T_{11}+T_{12}-\frac{1}{2}T_{13}$. We shall simplify each of the terms T_{11},T_{12} and T_{13} individually. Using Fourier expansion, T_{11} can be expanded as follows.

$$\begin{aligned} T_{11} &= E\left[B(f)B(g)\prod_{i=1}^{p}A_i(f_i)\right] \\ &= E\left[\left(\sum_{\beta_1\subseteq\mathcal{B}}\hat{B}_{\beta_1}\chi_{\beta_1}(f)\right)\left(\sum_{\beta_2\subseteq\mathcal{B}}\hat{B}_{\beta_2}\chi_{\beta_2}(g)\right)\prod_{i=1}^{p}\sum_{\alpha_i\subseteq\mathcal{A}}\hat{A}_{i,\alpha_i}\chi_{\alpha_i}(f_i)\right]\end{aligned}$$

By linearity of expectation, we obtain,

$$T_{11} \;=\; \sum_{\beta_1,\beta_2,\alpha_i}\hat{B}_{\beta_1}\hat{B}_{\beta_2}\left(\prod_{i=1}^{p}\hat{A}_{i,\alpha_i}\right)E\left[\chi_{\beta_1}(f)\chi_{\beta_2}(g)\prod_{i=1}^{p}\chi_{\alpha_i}(f_i)\right]$$

Recalling that $g=f(g_1\wedge g_2)\prod f_i\circ\pi_i$, we have

$$T_{11}=\sum_{\beta_1,\beta_2,\alpha_i}\hat{B}_{\beta_1}\hat{B}_{\beta_2}\left(\prod_{i=1}^{p}\hat{A}_{i,\alpha_i}\right)E\left[\chi_{\beta_1}(f)\chi_{\beta_2}\left(f(g_1\wedge g_2)\prod_{i=1}^{p}f_i\circ\pi_i\right)\right.$$
$$\left.\prod_{i=1}^{p}\chi_{\alpha_i}(f_i)\right]$$

Using property (4.1) that $\chi_\alpha(fg)=\chi_\alpha(f)\chi_\alpha(g)$, we have

$$T_{11}=\sum_{\beta_1,\beta_2,\alpha_i}\hat{B}_{\beta_1}\hat{B}_{\beta_2}\left(\prod_{i=1}^{p}\hat{A}_{i,\alpha_i}\right)E\left[\chi_{\beta_1}(f)\chi_{\beta_2}(f)\chi_{\beta_2}(g_1\wedge g_2)\right.$$
$$\left.\prod_{i=1}^{p}\chi_{\beta_2}(f_i\circ\pi_i)\prod_{i=1}^{p}\chi_{\alpha_i}(f_i)\right]$$

The function f, g_1, g_2 and f_i's are all chosen independently. Hence,

$$T_{11} = \sum_{\beta_1,\beta_2,\alpha_i} \hat{B}_{\beta_1}\hat{B}_{\beta_2}\left(\prod_{i=1}^{p}\hat{A}_{i,\alpha_i}\right)\left(E\Big[\chi_{\beta_1}(f)\chi_{\beta_2}(f)\Big]E\Big[\chi_{\beta_2}(g_1\wedge g_2)\Big]\right.$$
$$\left.\prod_{i=1}^{p}E\Big[\chi_{\beta_2}\left(f_i\circ\pi_i\right)\chi_{\alpha_i}(f_i)\Big]\right)$$

From properties (4.2),(4.3), we conclude that $E\left[\chi_{\beta_1}(f)\chi_{\beta_2}(f)\right]$ is 0 if $\beta_1 \neq \beta_2$ and 1 otherwise (i.e., when $\beta_1 = \beta_2 = \beta$). Thus,

$$T_{11} = \sum_{\beta,\alpha_i}\hat{B}_\beta^2\left(\prod_{i=1}^{p}\hat{A}_{i,\alpha_i}\right)\left(E\Big[\chi_\beta(g_1\wedge g_2)\Big]\prod_{i=1}^{p}E\Big[\chi_\beta\left(f_i\circ\pi_i\right)\chi_{\alpha_i}(f_i)\Big]\right)$$

Since g_1 and g_2 are chosen at random from $\mathcal{F}_\mathcal{B}$, the expected value of $g_1 \wedge g_2$ on any element in $\mathcal{B}$ is $\frac{1}{2}$. Hence, $E\left[\chi_\beta(g_1\wedge g_2)\right] = E\left[\prod_{x\in\beta}(g_1\wedge g_2)(x)\right] = \prod_{x\in\beta}E\left[(g_1\wedge g_2)(x)\right] = \left(\frac{1}{2}\right)^{|\beta|}$. We thus obtain,

$$T_{11} = \sum_{\beta,\alpha_i}\hat{B}_\beta^2\left(\frac{1}{2}\right)^{|\beta|}\left(\prod_{i=1}^{p}\hat{A}_{i,\alpha_i}\right)\prod_{i=1}^{p}E\Big[\chi_\beta\left(f_i\circ\pi_i\right)\chi_{\alpha_i}(f_i)\Big]$$

Using properties (4.2),(4.3), as before we conclude that $E\left[\chi_\beta\left(f_i\circ\pi_i\right)\chi_{\alpha_i}(f_i)\right]$ is 0 if $\alpha_i \neq \pi_i(\beta)$ and is 1 otherwise. We thus have the following expression for T_{11}.

$$T_{11} = \sum_\beta \hat{B}_\beta^2\prod_{i=1}^{p}\hat{A}_{i,\pi_i(\beta)}\left(\frac{1}{2}\right)^{|\beta|}$$

Analogously, T_{12} can be simplified to the following expression.

$$\begin{aligned} T_{12} &= E\left[B(f)B(g)B(g_1)\prod_{i=1}^{p}A_i(f_i)\right] \\ &= \sum_{\beta,\beta_1}\hat{B}_\beta^2\hat{B}_{\beta_1}\left(\prod_{i=1}^{p}\hat{A}_{i,\pi_i(\beta)}\right)E\Big[\chi_\beta(g_1\wedge g_2)\chi_{\beta_1}(g_1)\Big] \end{aligned}$$

Now, let us analyze the expression $E\Big[\chi_\beta(g_1 \wedge g_2)\chi_{\beta_1}(g_1)\Big]$.

$$
\begin{aligned}
E\Big[\chi_\beta(g_1 \wedge g_2)\chi_{\beta_1}(g_1)\Big] &= E\left[\prod_{x\in\beta}(g_1 \wedge g_2)(x)\prod_{x\in\beta_1} g_1(x)\right] \\
&= \prod_{x\in\beta\setminus\beta_1} E\left[(g_1 \wedge g_2)(x)\right] \prod_{x\in\beta_1\cap\beta} E\left[(g_1 \wedge g_2)g_1(x)\right] \\
&\qquad \prod_{x\in\beta_1\setminus\beta} E\left[g_1(x)\right] \\
&= \left(\frac{1}{2}\right)^{|\beta\setminus\beta_1|}\left(\frac{1}{2}\right)^{|\beta\cap\beta_1|} \prod_{x\in\beta_1\setminus\beta} E\left[g_1(x)\right] \\
&= \left(\frac{1}{2}\right)^{|\beta|} \prod_{x\in\beta_1\setminus\beta} E\left[g_1(x)\right]
\end{aligned}
$$

The step before the last one follows from the fact that for any element $x \in \mathcal{B}$, $E\left[(g_1 \wedge g_2)(x)\right] = \frac{1}{2}$ and $E\left[(g_1 \wedge g_2)g_1(x)\right] = \frac{1}{2}$. Now for any element $x \in \mathcal{B}$, $E\left[g_1(x)\right] = 0$. Hence, if $\beta_1\setminus\beta \neq \emptyset$, $E\left[\chi_\beta(g_1 \wedge g_2)\chi_{\beta_1}(g_1)\right] = 0$. Thus, T_{12} reduces to the following expression.

$$
T_{12} = \sum_{\beta}\sum_{\beta_1\subseteq\beta} \hat{B}_\beta^2\hat{B}_{\beta_1}\prod_{i=1}^{p}\hat{A}_{i,\pi_i(\beta)}\left(\frac{1}{2}\right)^{|\beta|}
$$

Using a similar analysis, T_{13} can be simplified to the following expression.

$$
T_{13} = \sum_{\beta\subseteq\mathcal{B}}\sum_{\beta_1,\beta_2\subseteq\beta} \hat{B}_\beta^2\hat{B}_{\beta_1}\hat{B}_{\beta_2}\prod_{i=1}^{p}\hat{A}_{i,\pi_i(\beta)}(-1)^{|\beta_1\cap\beta_2|}\left(\frac{1}{2}\right)^{|\beta|}
$$

Recalling that $T_1 = \frac{1}{2}T_{11} + T_{12} - \frac{1}{2}T_{13}$, we have the following.

$$
\begin{aligned}
T_1 \;=\; & \frac{1}{2}\sum_{\beta\subseteq\mathcal{B}}\hat{B}_\beta^2\prod_{i=1}^{p}\hat{A}_{i,\pi(\beta)}\left(\frac{1}{2}\right)^{|\beta|} + \sum_{\beta\subseteq\mathcal{B}}\sum_{\beta_1\subseteq\beta}\hat{B}_\beta^2\hat{B}_{\beta_1}\prod_{i=1}^{p}\hat{A}_{i,\pi_i(\beta)}\left(\frac{1}{2}\right)^{|\beta|} \\
& -\frac{1}{2}\sum_{\beta\subseteq\mathcal{B}}\sum_{\beta_1,\beta_2\subseteq\beta}\hat{B}_\beta^2\hat{B}_{\beta_1}\hat{B}_{\beta_2}\prod_{i=1}^{p}\hat{A}_{i,\pi_i(\beta)}(-1)^{|\beta_1\cap\beta_2|}\left(\frac{1}{2}\right)^{|\beta|}
\end{aligned}
$$

Upper bounding each term by its absolute value, we obtain the following inequality.

$$\begin{aligned}
T_1 &\leq \frac{1}{2}\sum_{\beta\subseteq\mathcal{B}} \hat{B}_\beta^2 \prod_{i=1}^{p} |\hat{A}_{i,\pi(\beta)}| \left(\frac{1}{2}\right)^{|\beta|} + \sum_{\beta\subseteq\mathcal{B}}\sum_{\beta_1\subseteq\beta} \hat{B}_\beta^2 |\hat{B}_{\beta_1}| \prod_{i=1}^{p} |\hat{A}_{i,\pi_i(\beta)}| \left(\frac{1}{2}\right)^{|\beta|} \\
&\quad + \frac{1}{2}\sum_{\beta\subseteq\mathcal{B}}\sum_{\beta_1,\beta_2\subseteq\beta} \hat{B}_\beta^2 |\hat{B}_{\beta_1}\hat{B}_{\beta_2}| \prod_{i=1}^{p} |\hat{A}_{i,\pi_i(\beta)}| \left(\frac{1}{2}\right)^{|\beta|} \\
&= \frac{1}{2}\sum_{\beta\subseteq\mathcal{B}} \hat{B}_\beta^2 \prod_{i=1}^{p} |\hat{A}_{i,\pi_i(\beta)}| \left(\frac{1}{2}\right)^{|\beta|} \left(1 + \sum_{\beta_1\subseteq\beta} |\hat{B}_{\beta_1}|\right)^2
\end{aligned}$$

□

The upper bound for T_2 (in Sub-Claim 4.7) is obtained by an analogous analysis. Recalling that $E[ACC] = \frac{1}{4} + \frac{1}{2}T_1 + \frac{1}{4}T_2$, we have,

$$E[ACC] \leq \frac{1}{4} + \frac{1}{4}\sum_{\beta} \hat{B}_\beta^2 \left(\prod_{i=1}^{p} |\hat{A}_{i,\pi_i(\beta)}| \frac{(1+\gamma_\beta)^2}{2^{|\beta|}} + \frac{1}{4}\frac{(1+\gamma_\beta)^4}{4^{|\beta|}}\right)$$

where γ_β denotes $\sum_{\beta_1\subseteq\beta} |\hat{B}_{\beta_1}|$.

We shall now show that $E[ACC] < \frac{1}{2} + \frac{\delta}{2}$ using the following three facts.

(i) If $|\beta| = 1$, then there exists an $i \in \{1, \dots, p\}$ such that $|\hat{A}_{i,\pi_i(\beta)}| < \delta$. (assumption made at the beginning of proof of claim)

(ii) If $|\beta|$ is even, then $|\hat{B}_\beta| = 0$. (due to folding.)

(iii) $\sum_\beta \hat{B}_\beta^2 = 1$ (Parseval's identity)

The following observations come useful in bounding $E[ACC]$.

- The contribution due to terms where $|\beta|$ is even is zero on account of *(ii)*.
- The contribution due to terms where $|\beta|$ is large is small due to the $2^{|\beta|}$ and $4^{|\beta|}$ in the denominator. (in our case, $|\beta| \geq 5$ is large enough for the analysis to go through.)
- The contribution due to terms where $|\beta| = 1$ is small since $|A_{i,\pi(\beta)}| < \delta$.

We handle the intermediate case of $|\beta| = 3$ explicitly. This intuition is made concrete in the following Sub-Claims. Define η_1, η_3, η_5 as follows.

$$\eta_1 = \sum_{|\beta|=1} \hat{B}_\beta^2 \qquad \eta_3 = \sum_{|\beta|=3} \hat{B}_\beta^2 \qquad \eta_5 = \sum_{|\beta|\geq 5} \hat{B}_\beta^2$$

Note that $\eta_1 + \eta_3 + \eta_5 = 1$ from *(ii)* and *(iii)*.

SUB-CLAIM 4.8. *For any $\beta \subseteq \mathcal{B}$, define*

$$\Omega_\beta \;=\; \hat{B}_\beta^2 \left(\prod_{i=1}^{p} |\hat{A}_{i,\pi_i(\beta)}| \frac{(1+\gamma_\beta)^2}{2^{|\beta|}} + \frac{1}{4} \frac{(1+\gamma_\beta)^4}{4^{|\beta|}} \right)$$

then facts (i), (ii) and (iii) imply

$$\begin{aligned}
\sum_{\beta:|\beta|=1} \Omega_\beta &< 2\eta_1\delta + \eta_1 \frac{(1+\sqrt{\eta_1})^4}{16} \\
\sum_{\beta:|\beta|=3} \Omega_\beta &\leq \eta_3 \frac{\left(1+\sqrt{1-\eta_1}+\sqrt{3\eta_1}\right)^2}{8} + \eta_3 \frac{\left(1+\sqrt{1-\eta_1}+\sqrt{3\eta_1}\right)^4}{256} \\
\sum_{\beta:|\beta|\geq 5} \Omega_\beta &\leq \frac{25}{32}\eta_5 + \frac{5^4}{4^6}\eta_5
\end{aligned}$$

PROOF OF SUB-CLAIM 4.8. We first make the following observations based on *(i)*, *(ii)* and *(iii)*.

- If $|\beta| = 1$, then $\gamma_\beta = |\hat{B}_\beta| \leq \sqrt{\eta_1}$. Hence, $(1+\gamma_\beta)^2/2^{|\beta|} \leq 2$ and $(1+\gamma_\beta)^4/4^{|\beta|} \leq (1+\sqrt{\eta_1})^4/4$.

- When $|\beta| = 3$, $\gamma_\beta = |\hat{B}_\beta| + \sum_{i=1}^{3} |\hat{B}_{\{u_i\}}|$ where $\beta = \{u_1, u_2, u_3\}$. Thus, γ_β is maximized when all the weight is concentrated on the four terms in the above expression (i.e., $\hat{B}_\beta^2 + \sum_{i=1}^{3} \hat{B}_{\{u_i\}}^2 = 1$) and when the weight is distributed equally across the singleton sets $\{u_i\}$'s. This happens when $|\hat{B}_\beta| = \sqrt{1-\eta_1}$ and $|\hat{B}_{\{u_i\}}| = \sqrt{\eta_1/3}$. Thus, in this case, $\gamma_\beta \leq 1 + \sqrt{1-\eta_1} + \sqrt{3\eta_1}$.

- Finally to the case when $|\beta| \geq 5$. We know from Cauchy's inequality that $(\sum_{i=1}^{m} p_i)^2 \leq m \sum_{i=1}^{m} p_i^2$. Thus, $\gamma_\beta = \sum_{\beta' \subseteq \beta} |\hat{B}_{\beta'}| \leq \sqrt{2^{|\beta|-1} \sum_{\beta' \subseteq \beta} \hat{B}_{\beta'}^2} \leq 2^{(|\beta|-1)/2}$. Note that we have only $2^{|\beta|-1}$ terms in the summation as $\hat{B}_{\beta'} = 0$ when $|\beta'|$ is even due to folding. Thus, $(1+\gamma_\beta)^2/2^{|\beta|} \leq (1 + 2^{(|\beta|-1)/2})^2/2^{|\beta|} = (2^{-|\beta|/2} + 2^{-1/2})^2 \leq 25/32$ since $|\beta| \geq 5$.

These observations lead to the bounds indicated in Sub-Claim 4.8. □

With Sub-Claim 4.8 and fact *(ii)*, we have that

$$
\begin{aligned}
E[ACC] &< \frac{1}{4} + \frac{1}{4}\left[2\eta_1\delta + \eta_1\frac{(1+\sqrt{\eta_1})^4}{16} + \eta_3\frac{\left(1+\sqrt{1-\eta_1}+\sqrt{3\eta_1}\right)^2}{8}\right. \\
&\qquad \left. + \eta_3\frac{\left(1+\sqrt{1-\eta_1}+\sqrt{3\eta_1}\right)^4}{256} + \left(\frac{25}{32}+\frac{5^4}{4^6}\right)\eta_5\right] \\
&\le \frac{1}{4} + \frac{\delta}{2} + \frac{1}{4}\left[\eta_1\frac{(1+\sqrt{\eta_1})^4}{16} + \eta_3\frac{\left(1+\sqrt{1-\eta_1}+\sqrt{3\eta_1}\right)^2}{8}\right. \\
&\qquad \left. + \eta_3\frac{\left(1+\sqrt{1-\eta_1}+\sqrt{3\eta_1}\right)^4}{256} + \left(\frac{25}{32}+\frac{5^4}{4^6}\right)\eta_5\right] \\
&\qquad\qquad (\text{ since } \eta_1 \le 1) \\
&= \frac{1}{4} + \frac{\delta}{2} + \frac{1}{4}\cdot\lambda(\eta_1,\eta_3,\eta_5)
\end{aligned}
$$

where $\lambda(\cdot,\cdot,\cdot)$ is defined suitably. We have thus reduced the upper bound of $E[ACC]$ to an expression involving just three parameters. Observe that $\lambda(\eta_1,\eta_3,\eta_5)$ is of the form $\lambda_1(\eta_1) + \eta_3\lambda_2(\eta_1) + C\eta_5$ where λ_1, λ_2 are the appropriate functions and C a constant. Since $\eta_1 + \eta_3 + \eta_5 = 1$, for any fixed η_1,

$$
\begin{aligned}
\lambda_2(\eta_1) < C &\implies \lambda(\eta_1,\eta_3,\eta_5) \le \lambda_1(\eta_1) + C(1-\eta_1) \\
\lambda_2(\eta_1) \ge C &\implies \lambda(\eta_1,\eta_3,\eta_5) \le \lambda_1(\eta_1) + (1-\eta_1)\lambda_2(\eta_1)
\end{aligned}
$$

Using the above observation and the fact that $\sqrt{1-\eta_1} \le 1 - \eta_1/2$ for $|\eta_1| \le 1$, we have the following bound for $\lambda(\eta_1,\eta_3,\eta_5)$.

SUB-CLAIM 4.9. *For any $\eta_1,\eta_3,\eta_5 \in \mathbb{R}^+$ such that $\eta_1+\eta_3+\eta_5 = 1$, we have*

$$
\lambda(\eta_1,\eta_3,\eta_5) \le \max_{x\in[0,1]} \begin{cases} p(x) \\ q(x) \end{cases}
$$

where p, q are polynomials defined as follows:

$$
\begin{aligned}
p(x) &= x^2\frac{(1+x)^4}{16} + \left(\frac{25}{32}+\frac{5^4}{4^6}\right)(1-x^2) \\
q(x) &= x^2\frac{(1+x)^4}{16} + (1-x^2)\left(\frac{(2-\frac{x^2}{2}+\sqrt{3}x)^2}{8} + \frac{(2-\frac{x^2}{2}+\sqrt{3}x)^4}{256}\right)
\end{aligned}
$$

Finally, we shall bound the value of polynomials p, q for $x \in [0,1]$ to obtain the following sub-claim

SUB-CLAIM 4.10. *For $x \in [0,1]$,*

$$p(x), q(x) \leq 1$$

where p, q are polynomials as defined in Sub-Claim 4.9

This Sub-Claim is proved in Section 4.3.3. From Sub-Claim 4.9 and Sub-Claim 4.10, we have $E[ACC] < \frac{1}{2} + \frac{\delta}{2}$. Thus choosing $\delta = 2\epsilon$, we have that the acceptance probability of V_{inner} is less than $\frac{1}{2} + \epsilon$, which is what we wanted to prove. □

4.3.3. Proof of Sub-Claim 4.10. Sub-Claim 4.10 can be checked numerically for the polynomials p and q. We however give an alternate proof employing Sturm sequences. Sturm sequences are used to calculate the number of distinct real zeroes of any polynomial between any two real numbers.

DEFINITION 4.11. *Given a polynomial $f \in \mathbb{R}[x]$, the sturm sequence of f, sturm-seq(f), is a sequence of polynomials $< f_0, f_1, \ldots, f_s >$ where the polynomials $f_0, f_1, \ldots, f_s \in \mathbb{R}[x]$ are defined as follows:*

$$\begin{aligned}
f_0 &= f & & \\
f_1 &= f' & & \text{where } f' \text{ is the derivative of } f \\
f_0 &= f_1 q_1 - f_2 & & \text{where } q_1 \in \mathbb{R}[x],\ \deg(f_2) < \deg(f_1) \\
&\vdots & & \vdots \\
f_{k-2} &= f_{k-1} q_{k-1} - f_k & & \text{where } q_{k-1} \in \mathbb{R}[x],\ \deg(f_k) < \deg(f_{k-1}) \\
&\vdots & & \vdots \\
f_{s-2} &= f_{s-1} q_{s-1} - f_s & & \text{where } q_{s-1} \in \mathbb{R}[x],\ \deg(f_s) < \deg(f_{s-1}) \\
f_{s-1} &= f_s q_s & & \text{where } q_s \in \mathbb{R}[x]
\end{aligned}$$

For any $a \in \mathbb{R}$, $\text{sturm}_a(f) =< a_1, a_2, \ldots, a_s >$ where

$$a_i = \begin{cases} `+' & \text{if} f_i(a) > 0 \\ `-' & \text{if} f_i(a) < 0 \\ 0 & \text{otherwise} \end{cases}$$

(i.e., a_i is the sign of $f_i(a)$).

For any $a \in \mathbb{R}$, $\#Var_a(f)$ is defined to be the number of sign changes in the sequence $\text{sturm}_a(f)$. If any 0's occur in the sequence $\text{sturm}_a(f)$, then we consider the abbreviated sequence discarding the 0's.

The following theorem gives the relationship between the number of real roots of a polynomial f between any two points a and b with $\#Var_a(f)$ and $\#Var_b(f)$.

THEOREM 4.12. *For any polynomial $f \in \mathbb{R}[x]$ and any two real numbers $a < b \in \mathbb{R}$, the number of distinct roots of the polynomial f in the range $(a, b]$*[2] *is given by $\#Var_a(f) - \#Var_b(f)$.*

A proof of this theorem can be found in Mishra (1993).

We are now ready to prove Sub-Claim 4.10

PROOF OF SUB-CLAIM 4.10. We need to show that $p(x), q(x) \leq 1$ for $x \in [0, 1]$.

Consider the first polynomial p. Let $p'(x) = p(x) - 1$. Note that $p'(1) = 0$. The sturm sign sequence for p' at the points $x = 0$ and $x = 1$ can be shown to be the following. (Unfortunately, the actual values of the p_i''s at $x = 0, 1$ are rational numbers involving too many digits to be printed here.)

$$\begin{aligned} \text{sturm}_0(p') &= < -, 0, +, +, -, -, + > \\ \text{sturm}_1(p') &= < 0, +, +, +, -, -, + > \end{aligned}$$

Hence $\#Var_0(p') = 3$ and $\#Var_1(p') = 2$. Hence the number of distinct real roots of p' in $(0, 1]$ is 1. As $p'(1) = 0$, 1 is the only real root in this range. Hence, $p'(x) \geq 0$ or $p'(x) \leq 0$ for all x such that $0 < x \leq 1$. Since $p'(0) < 0$, we have that $p'(x) \leq 0$ for all x in $[0, 1]$. Hence, $p(x) \leq 1, \forall x \in [0, 1]$.

Now for the other polynomial q. Define $q'(x) = q(x) - 1$. We have $q'(1) = 0$. The sturm sign sequence for q' can be shown to be the following:

$$\begin{aligned} \text{sturm}_0(q') &= < -, +, +, -, +, +, -, -, +, -, - > \\ \text{sturm}_1(q') &= < 0, +, -, -, +, +, -, -, -, +, - > \end{aligned}$$

Hence $\#Var_0(q') = 6$ and $\#Var_1(q') = 5$. Hence the number of distinct real roots of q' in $(0, 1]$ is 1. As $q'(1) = 0$, 1 is the only real root in this range. As $q'(0) < 0$, we have that $q'(x) \leq 0$ for all x such that $0 \leq x \leq 1$. Hence, $q(x) \leq 1, \forall x \in [0, 1]$.

Thus, both $p(x)$ and $q(x)$ are at most 1. This proves Sub-Claim 4.10. □

[2] $(a, b] = \{x \in \mathbb{R} : a < x \leq b\}$

4.4. Composed Verifier. There is a natural way to compose a p-prover MIP verifier V_{out} with an inner verifier such as V_{inner} described in the beginning of this section so as to preserve perfect completeness. In this section, we describe this composed verifier and thus, prove Lemma 2.6. We show that the number of queries issued by this composed verifier is exactly that of the inner verifier while the randomness is the sum of the randomness of the MIP verifier and the inner verifier. We then discuss the completeness and soundness of this composed verifier.

Recall the statement of Lemma 2.6. For every $\epsilon > 0$, we wish to show that there exists a $\gamma > 0$ such that

$$\text{MIP}_{1,\gamma}[p, r, a] \subseteq \text{PCP}_{1,\frac{1}{2}+\epsilon}[r + O(2^{pa}), p + 7] \tag{4.13}$$

Let $\epsilon > 0$ be an arbitrary number. Choose $\varepsilon = \epsilon/2$. By Claim 4.4, there exists a $\delta = \delta_\varepsilon$ such that the statement of Claim 4.4 holds. Choose $\gamma = \varepsilon\delta^{2p}$. For this choice of γ, we shall show that (4.13) holds good, thus proving Lemma 2.6.

Let $L \in \text{MIP}_{1,\gamma}[p, r, a]$. Let V_{out} be the corresponding MIP verifier for L. The action of the MIP verifier V_{out} is recalled below.

V_{out} interacts with p provers, $\Pi_1, \ldots, \Pi_p$. On an input string x of length n, V_{out} picks a $r(n)$-bit random string R and generates p queries $(1, q_1^{(R)}), \ldots, (p, q_p^{(R)})$ and a linear sized circuit C_R. It then issues query $(i, q_i^{(R)})$ to prover Π_i which responds with the answer $a_{i,q_i^{(R)}}$. V_{out} accepts iff $C_R(a_{1,q_1^{(R)}}, \ldots, a_{p,q_p^{(R)}}) = -1$.

Let Q be the set of all queries issued by V_{out} on input string x over all random strings R. (Note that $|Q| \leq p2^r$ since each random string R uniquely determines the queries V_{out} issues to the prover Π_i's) The p provers $\Pi_1, \ldots, \Pi_p$ that V_{out} interacts with can be thought of as p functions $\Pi_i : Q \to \{0,1\}^a$.

We shall now construct a $(r + O(2^{pa}), p + 7)$-restricted verifier V_{comp} for L by composing V_{out} with the inner verifier V_{inner} specified in Section 4.2. The proof (or oracle) that V_{comp} expects is of the form $\Gamma : \{0,1\}^* \to \{+1, -1\}$.

${V_{\text{comp}}}^{\Gamma}(x)$

1. Pick a random string $R \in \{0,1\}^{r(n)}$.
2. Generate queries $(1, q_1^{(R)}), \ldots, (p, q_p^{(R)})$ and circuit C_R as V_{out} would do on input x and random string R.
3. For each $i \in \{1, \ldots, p\}$, set $A_i(\cdot) \leftarrow \Gamma(i, q_i^{(R)}, \cdot)$.
4. Set $B \leftarrow \Gamma(p+1, R, \cdot)$.

5. Set $\mathcal{A} \leftarrow \{-1,-1\}^{a(n)}$.
6. Set $\mathcal{B} \leftarrow \{(a_1,\ldots,a_p) | C_R(a_1,\ldots,a_p) = -1\}$.
7. For each $i \in \{1,\ldots,p\}$, set the projection function $\pi_i : \mathcal{B} \to \mathcal{A}$ such that $(a_1,\ldots,a_p) \overset{\pi_i}{\longmapsto} a_i$.
8. Accept iff ${V_{\text{inner}}}^{A_1,\ldots,A_p,B}(\mathcal{A},\mathcal{B},\pi_1,\ldots,\pi_p)$ accepts.

Clearly the number of queries issued by V_{comp} is that of V_{inner} which is $p+7$, while the total randomness is the sum of the randomness of V_{out} and V_{inner} which is $r + O(2^{pa})$.

4.4.1. Completeness of Composed Verifier. It is easy to verify that V_{comp} has completeness 1. Suppose $x \in L$. By the completeness of V_{out}, there exist tables $\Pi_1,\ldots,\Pi_p$ such that $\Pr_R[{V_{\text{out}}}^{\Pi_1,\ldots,\Pi_p}(x,R) = \mathsf{accept}] = 1$. For each $R \in \{0,1\}^r$, let $(1, q_{j_1}^{(R)}),\ldots,(p, q_{j_p}^{(R)})$ be the queries issued by V_{out} on input string x and random string R. Construct another oracle $\Pi_{p+1} : \{0,1\}^r \to \{0,1\}^{ap}$ such that $\Pi_{p+1}(R) = (a_{1,q_1^{(R)}},\ldots,a_{p,q_p^{(R)}})$ where $a_{i,q_i^{(R)}} = \Pi_i(q_i^{(R)})$ (i.e., response of oracle Π_i on query $q_i^{(R)}$). Now if we construct Γ such that

- For each $i \in \{1,\ldots,p\}$, and $q \in Q$, $\Gamma(i,q,\cdot)$ is the long code of $\Pi_i(q)$.
- For each $R \in \{0,1\}^r$, $\Gamma(p+1,R,\cdot)$ is the long code of $\Pi_{p+1}(R)$.

We note that V_{comp} accepts on all random strings. Thus, the completeness is 1.

4.4.2. Soundness of Composed Verifier. The only thing that remains to be proved is that the soundness of V_{comp} is $\frac{1}{2}+\epsilon$. We prove this by showing that if V_{comp} accepts x with probability at least $\frac{1}{2}+\epsilon$, i.e.,

$$\Pr_{R'}[V^{\Gamma}(x;R') = \mathsf{accept}] \geq \frac{1}{2}+\epsilon$$

(where R' is the combined randomness of V_{out} and V_{inner}) then $x \in L$. By the soundness condition of the outer MIP verifier V_{out}, it is sufficient if we show that there exist provers $\Pi_1,\ldots,\Pi_p$ such that

$$\Pr_{R}[V^{\Pi_1,\ldots,\Pi_p}(x;R) = \text{ accept }] \geq \gamma$$

And the rest of the proof would be devoted to proving this fact.

Consider the following randomized strategy DECODE that takes as input a folded table A and returns an a-bit string x. A is an oracle whose input are functions of the form $f \in \mathcal{F}_{\mathcal{A}}$. Recall $\mathcal{A} = \{-1,1\}^a$.

$\textsc{Decode}(A)$

1. Choose $\alpha \subseteq \mathcal{A}$ with probability $\hat{A}_\alpha^2$.
2. Choose an $x \in \alpha$ uniformly at random.
3. Return x.

We remark that since $\sum_\alpha \hat{A}_\alpha^2 = 1$, the $\hat{A}_\alpha^2$'s determine a probability distribution and hence step 1 is legitimate. Moreover, the procedure will never get stuck in step 2 because of choosing $\alpha = \emptyset$ since $\hat{A}_\emptyset = 0$ (as A is folded.) We thus have that if $|\hat{A}_{\{a\}}| \geq \delta$, then $\Pr[\textsc{Decode}(A) = a] \geq \delta^2$.

Now imagine constructing the p provers $\Pi_1, \ldots, \Pi_p$ using the randomized strategy $\textsc{Decode}$ (on the proof Γ of the composed verifier V_{comp}) as follows:

For each $i \in \{1, \ldots, p\}$ do

For each $q \in Q$ do

Set $a_{i,q} \leftarrow \textsc{Decode}(\Gamma(i, q, \cdot))$.

Set prover $\Pi_i : Q \to \{0,1\}^a$ such that $\Pi_i(q) = a_{i,q}, \forall q \in Q$.

We shall now show that if V_{comp} accepts x on proof Γ with probability at least $\frac{1}{2} + \epsilon$, then V_{out} accepts x on interacting with the p provers $\Pi_1, \ldots, \Pi_p$ as constructed above with probability at least γ (over the random coin tosses of V_{out} and the $\textsc{Decode}$ strategy.)

Let $\mathcal{R}$ denote the set of random choices of the MIP verifier V_{out} that satisfy

$$\Pr_{R''}[{V_{\text{inner}}}^{A_1, \ldots, A_p, B}(x; R'') = \mathsf{accept}] \geq \frac{1}{2} + \frac{\epsilon}{2}$$

where each of $A_i(\cdot) = \Gamma(i, q_i^{(R)}, \cdot)$ and $B = \Gamma(p+1, R, \cdot)$ is as specified in the working of V_{comp} and the probability is taken over the coin tosses R'' of V_{inner}. By an averaging argument, it follows that $\Pr_R[R \in \mathcal{R}] \geq \epsilon/2$. Let $\varepsilon = \epsilon/2$ and $\delta = \delta_\varepsilon$ as mentioned in the beginning of the proof. By the soundness condition for the inner verifier V_{inner} (see Claim 4.4), we have that for each $R \in \mathcal{R}$, there exist $a_1^{(R)}, \ldots, a_p^{(R)}$ such that $C_R(a_1^{(R)}, \ldots, a_p^{(R)}) = -1$ and for each $l \in \{1, \ldots, p\}$, $|\hat{A}_{i,\{a_l^{(R)}\}}| \geq \delta$. Translating these conditions into the proof of the composed verifier V_{comp}, we have that for each $R \in \mathcal{R}$, there exist $a_1^{(R)}, \ldots, a_p^{(R)}$ such that $C_R(a_1^{(R)}, \ldots, a_p^{(R)}) = -1$ and for each $l \in \{1, \ldots, p\}$, $|(\hat{\Gamma}(i, q_i^{(R)}, \cdot)_{\{a_l^{(R)}\}}| \geq \delta$. We now use these facts to produce p provers $\Pi_1, \ldots, \Pi_p$ for V_{out} such that V_{out} accepts these p provers with probability at least γ.

Reiterating the soundness condition from the inner verifier V_{inner}, we have that for each $R \in \mathcal{R}$, there exist $a_1^{(R)}, \ldots, a_p^{(R)}$ such that $C_R(a_1^{(R)}, \ldots, a_p^{(R)}) = -1$ and for each $l \in \{1, \ldots, p\}$, $|(\hat{\Gamma}(i, q_i^{(R)}, \cdot))_{\{a_l^{(R)}\}}| \geq \delta$. Now, let us analyze the probability of the outer verifier accepting the provers $\Pi_1, \ldots, \Pi_p$ on input string x, where the provers Π_i are constructed from Γ as mentioned before.

$$\begin{aligned}
\Pr\Big[\quad {V_{\text{out}}}^{\Pi_1,\ldots,\Pi_p} \quad & (x; R) = \mathsf{accept}\Big] \\
= \quad & \Pr\left[C_r(a_{1,q_1^{(R)}}, \ldots, a_{p,q_p^{(R)}}) = -1\right] \\
\geq \quad & \Pr\left[\forall i, \Pi_i(q_i^{(R)}) = a_i^{(R)}\right] \\
\geq \quad & \Pr_R\left[R \in \mathcal{R}\right] \cdot \Pr\left[\forall i, \Pi_i(q_i^{(R)}) = a_i^{(R)} \middle| R \in \mathcal{R}\right] \\
= \quad & \Pr_R\left[R \in \mathcal{R}\right] \cdot \Pr\left[\forall i, \textsc{Decode}\left(\Gamma(i, q_i^{(R)}, \cdot)\right) = a_i^{(R)} \middle| R \in \mathcal{R}\right] \\
= \quad & \Pr_R\left[R \in \mathcal{R}\right] \cdot \prod_{i=1}^{p} \Pr\left[\textsc{Decode}\left(\Gamma(i, q_i^{(R)}, \cdot)\right) = a_i^{(R)} \middle| R \in \mathcal{R}\right] \\
\geq \quad & \varepsilon \delta^{2p} \\
= \quad & \gamma
\end{aligned}$$

(all the probabilities are over the random coins of both V_{out} and the DECODE procedure unless otherwise specified.) Thus, there exists provers $\Pi_1, \ldots, \Pi_p$ such that V_{out} accepts with probability at least γ, which in turn implies that $x \in L$. This completes the proof of the Lemma 2.6

5. Scope for Further Improvements

The following are a few approaches which would further reduce the size-query complexity in the construction of PCPs described in this paper.

1. An improved low-error analysis of the low-degree test of Rubinfeld & Sudan (1996) in the case when the field size is linear in the degree of the polynomial. (It is to be noted that the current best analysis (Arora & Sudan 1997b) requires the field size to be at least a fourth power of the degree.) Such an analysis would reduce the proof blowup to nearly quadratic.

2. It is known that for every $\epsilon, \delta > 0$, $\text{MIP}_{1,\epsilon}[1, 0, n] \subseteq \text{PCP}_{1-\delta,\frac{1}{2}}[c \log n, 3]$ from the results of Håstad (1997a). Traditionally, results of this nature

have led to the construction of inner verifiers for p-prover MIPs and thus showing that for every $\delta > 0$ and p there exists $\epsilon > 0$ and c such that

$$\mathrm{MIP}_{1,\epsilon}[p, r, a] \subseteq \mathrm{PCP}_{1-\delta,\frac{1}{2}}[r + c \log a, p + 3].$$

Proving a result of this nature would reduce the query complexity of the small PCPs constructed in this paper to 6 (when composed with Lemma 2.4).

Acknowledgements

Research supported in part by Sloan Foundation, NSF Career Award CCR-9875511 and CCR-9912342

References

Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan & Mario Szegedy (1998). Proof Verification and the Hardness of Approximation Problems. *Journal of the ACM* **45**(3), 501–555.

Sanjeev Arora & Shmuel Safra (1998). Probabilistic Checking of Proofs: A New Characterization of NP. *Journal of the ACM* **45**(1), 70–122.

Sanjeev Arora & Madhu Sudan (1997a). Improved Low Degree Testing and its Applications. In *Proc.* 29*th ACM Symp. on Theory of Computing*, 485–495. El Paso, Texas.

Sanjeev Arora & Madhu Sudan (1997b). Improved Low Degree Testing and its Applications. Technical Report TR97-003, Electronic Colloquium on Computational Complexity. URL `http://www.eccc.uni-trier.de/eccc/`.

László Babai, Lance Fortnow & Carsten Lund (1991). Non-Deterministic Exponential Time has Two-Prover Interactive Protocols. *Computational Complexity* **1**, 3–40.

Mihir Bellare, Oded Goldreich & Madhu Sudan (1998). Free Bits, PCPs, and Nonapproximability—Towards Tight Results. *SIAM Journal of Computing* **27**(3), 804–915.

Mihir Bellare, Shafi Goldwasser, Carsten Lund & Alexander Russell (1993). Efficient Probabilistically Checkable Proofs and Applications to Approximation. In *Proc.* 25*th ACM Symp. on Theory of Computing*, 294–304. San Diego, California.

STEPHEN A. COOK (1988). Short propositional formulas represent nondeterministic computations. *Information Processing Letters* **26**(5), 269–270.

IRIT DINUR, ELDAR FISCHER, GUY KINDLER, RAN RAZ & SHMUEL SAFRA (1999). PCP Characterizations of NP: Towards a Polynomially-Small Error-Probability. In *Proc. 31th ACM Symp. on Theory of Computing*, 29–40. Atlanta, Georgia.

KATALIN FRIEDL & MADHU SUDAN (1995). Some Improvements to Total Degree Tests. In *Proc. 3rd Israel Symposium on Theoretical and Computing Systems*, 190–198. Tel Aviv, Israel.

JOHAN HÅSTAD (1996). Clique is Hard to Approximate Within $n^{1-\epsilon}$. In *Proc. 37nd IEEE Symp. on Foundations of Comp. Science*, 627–636. Burlington, Vermont.

JOHAN HÅSTAD (1997a). Some optimal inapproximability results. In *Proc. 29th ACM Symp. on Theory of Computing*, 1–10. El Paso, Texas.

JOHAN HÅSTAD (1997b). Some optimal inapproximability results. Technical Report TR97-037, Electronic Colloquium on Computational Complexity. URL `http://www.eccc.uni-trier.de/eccc/`.

FRANK THOMSON LEIGHTON (1992). *Introduction to Parallel Algorithms and Architectures*. Morgan Kaufmann Publishers, Inc., San Mateo, CA.

CARSTEN LUND, LANCE FORTNOW, HOWARD KARLOFF & NOAM NISAN (1990). Algebraic Methods for Interactive Proof Systems. In *Proc. 31st IEEE Symp. on Foundations of Comp. Science*, 2–10. St. Louis, Missouri.

BHUBANESHWAR MISHRA (1993). *Algorithmic Algebra*. Springer Verlag, New York, NY.

ALEXANDER POLISHCHUK & DANIEL A. SPIELMAN (1994). Nearly-linear Size Holographic Proofs. In *Proc. 26th ACM Symp. on Theory of Computing*, 194–203. Montréal, Québec, Canada.

RAN RAZ & SHMUEL SAFRA (1997). A Sub-Constant Error-Probability Low-Degree Test, and a Sub-Constant Error-Probability PCP Characterization of NP. In *Proc. 29th ACM Symp. on Theory of Computing*, 475–484. El Paso, Texas.

RONITT RUBINFELD & MADHU SUDAN (1996). Robust Characterizations of Polynomials with Applications to Program Testing. *SIAM Journal of Computing* **25**(2), 252–271.

JACOB T. SCHWARTZ (1980). Fast Probabilistic Algorithms for Verification of Polynomial Identities. *Journal of the ACM* **27**(4), 701–717.

DANIEL SPIELMAN (1995). *Computationally Efficient Error-Correcting Codes and Holographic Proofs.* Ph.D. thesis, Massachusetts Institute of Technology.

MADHU SUDAN (1992). *Efficient Checking of Polynomials and Proofs and the Hardness of Approximation Problems.* Ph.D. thesis, University of California, Berkeley.

MARIO SZEGEDY (1999). Many-Valued Logics and Holographic Proofs. In *Automata, Languages and Programming, 26st International Colloquium*, JIRÍ WIEDERMANN, PETER VAN EMDE BOAS & MOGENS NIELSEN, editors, volume 1644 of *Lecture Notes in Computer Science*, 676–686. Springer-Verlag, Prague, Czech Republic.

Manuscript received October 31, 2000

PRAHLADH HARSHA
Laboratory for Computer Science,
Massachusetts Institute of Technology,
545 Technology Square,
Cambridge, MA 02139.
prahladh@mit.edu

MADHU SUDAN
Laboratory for Computer Science,
Massachusetts Institute of Technology,
545 Technology Square,
Cambridge, MA 02139.
madhu@mit.edu
http://theory.lcs.mit.edu/~madhu/