RESULTS ON PROBABILISTICALLY CHECKABLE PROOFS OF PROXIMITY

ALLEN LIN, LEO GAGNON, AND PRATYUSH VENKATAKRISHNAN

Abstract. We survey a few papers on probabilistically checkable proofs (**PCP**s). We introduce a notion of probabilistically checkable proofs of proximity (**PCPP**s) that rely on string distance, usually Hamming distance. We give a few results on the comparison to **NP** languages, composing and constructing more efficient **PCP**s, and results on **PCP**s of quasilinear length.

1. Introduction

A probabilistically checkable proof system is given by a probabilistic verifier V with random access to a proof string π. It then uses some random bits and queries some bits of π to either accept if π is a correct proof or reject if π is an incorrect proof, with high probabilities. Formallly, for a language $L \subseteq \{0,1\}^*$, functions $q, r \colon \mathbb{N} \to \mathbb{N}$, and constants $0 \le s \le c \le 1$, we say L has a $(r(n), q(n))$-**PCP** verifier with completeness c and soundness s if there exists a polynomial time probabilistic verifier V such that

- (*completeness*) for any $x \in L$, there exists π such that $\Pr[V(x,\pi) = 1] \ge c$; and
- (*soundness*) for any $x \notin L$ and for any π, $\Pr[V(x,\pi) = 1] \le s$.

If such a V exists, we say that $L \in \mathbf{PCP}_{c,s}(r(n), q(n))$. If $c = 1$, we say that the verifier has *perfect completeness*.

The celebrated **PCP** theorem [1] states that $\mathbf{NP} = \mathbf{PCP}_{1,1/2}[O(\log n), O(1)]$. In other words, any **NP** language has a **PCP**-verifier that reads a constant number of bits from a supposed proof π and decides to accept or reject with perfect completeness and soundness $1/2$.

We introduce a new notion that adapts standard **PCP**s to a concept of "distance" (specifically Hamming distance) as follows.

Definition 1.1. We define the *relative Hamming distance* between $u, v \in \{0,1\}^\ell$ by $\Delta(u,v) \coloneqq \#\{i \mid u_i \ne v_i\}/\ell$.

Definition 1.2. A *pair language* L splits strings into two parts, $L = \{(x,y)\}$. The verifier gets access to x but only oracle access to y, where queries to y are counted in query complexity. Define $L(x) \coloneqq \{y \mid (x,y) \in L\}$.

Definition 1.3 (**PCP**s of Proximity, Ben-Sasson and Sudan [3, 4])**.** For functions $s, \delta \colon \mathbb{N} \to [0,1]$, a verifier V is a **PCP** *of proximity* **PCPP** *verifier* for pair language L with proximity δ and soundness s if for every (x,y),

Date: May 14, 2024.

- (*completeness*) if $(x, y) \in L$, then there exists some π such that $V(x)$ accepts with oracle y and π with probability 1;
- (*soundness*) if y is $\delta(|x|)$-far from $L(x)$, then for every π, $V(x)$ accepts with oracle y and π with probability less than $s(|x|)$.

As compared to **PCP**s, **PCPP**s have a weaker condition on soundness—only inputs that are far (in relative Hamming distance) from all members of the language need to be rejected with high probability, but the query complexity condition is stronger since V has only oracle access to y, meaning that queries to y are counted in the query complexity.

In section 2, we summarize a result on short **PCPP**s to **NP** languages. In section 3, we introduce the notion of robust soundness and use it, along with a highly efficient **PCPP** construct, to compose **PCP**s more efficiently. Finally, in section 4, we prove a result on **PCP**s of quasilinear length. Specifically we prove the following theorem.

Theorem 1.4 (Ben-Sasson and Sudan [6])**.** For any complexity function $t\colon \mathbb{N} \to \mathbb{N}$,

$$\mathbf{NTIME}(t(n)) \subseteq \mathbf{PCP}_{1,1/2}[\log(t(n) \cdot \operatorname{polylog}(t(n))), \operatorname{polylog}(t(n))]$$

holds.

2. PCPPs for NP languages

The main result stated in Ben-Sasson and Sudan [3] is the following:

Theorem 2.1 (Ben-Sasson and Sudan [3])**.** There is a **PCP** of Proximity to every **NP** language.

This is derived from the following result, which gives a **PCPP** for any language $L \in \mathbf{NTIME}(T(n))$ with $\operatorname{polylog}(T(n))$ verification time for proofs of length $T(n) \cdot \operatorname{polylog}(T(n))$; the query complexity and proof length are comparable to previous results, such as Ben-Sasson and Sudan [5].

Theorem 2.2 (Efficient **PCPP**s with short proofs, Ben-Sasson and Sudan [3])**.** Fix a pair language $L \in \mathbf{NTIME}(t)$ where $t\colon \mathbb{N} \to \mathbb{N}$ is a non-decreasing function. Then, for every constant $s > 0$, we have $L \in \mathbf{PCPP}_{s,\delta}[r, q, t]$, for

- *proximity parameter* $\delta(m) = 1/\operatorname{polylog}(t(m))$,
- *randomness complexity* $r(m) = \log_2(t(m)) + O(\log\log t(m))$,
- *query complexity* $q(m) = \operatorname{polylog}(t(m))$, and
- *verification time* $\tau(n, K) = \operatorname{poly}(n, \log K, \log(t(n + K)))$.

The proof of this theorem crucially relies on a reduction from arbitrary $\mathbf{NTIME}(t(n))$ languages to a particular constraint satisfaction problem on extended de Bruijn graphs, which can be stated as a sort of generalized graph coloring problem. In this problem, the color of each node only needs to be "consistent" with the colors of neighboring nodes, where the notion of consistency is given by a list of legal values for each neighborhood in the graph—this can be expressed as an $\mathbf{NC}_1$ circuit. We now state the relevant definitions.

Definition 2.3. An n-dimensional *de Bruijn graph* of m symbols is a directed graph whose vertices are all strings of length n over m symbols,

$$V = \underbrace{\{s_1, \ldots, s_m\}}_{S}^{n},$$

and edges

$$E = \{((r_1, r_2, \ldots, r_n), (r_2, \ldots, r_n, r_{n+1})) \mid r_1, \ldots, r_{n+1} \in S\}.$$

The paper uses a related family of graphs, known as *extended de Bruijn graphs*:

Definition 2.4. The *extended de Bruijn graph* $\mathcal{DB}_{k,\ell}$ is a directed graph with ℓ layers, each of which contains 2^k nodes represented by k-bit strings. (In other words, $V = \{0, \ldots, \ell - 1\} \times \{0,1\}^k$.)

The node $v = (i, (b_0, \ldots, b_{i^*}, \ldots, b_{k-1}))$ then has outgoing edges to the two nodes

$$\Gamma_{i,0}(v) = (i+1, (b_0, \ldots, b_{i^*}, \ldots, b_{k-1})),$$
$$\Gamma_{i,1}(v) = (i+1, (b_0, \ldots, b_{i^*} \oplus 1, \ldots, b_{k-1})),$$

where $i^* \coloneqq i \bmod k$ and $\oplus$ denotes bitwise xor.

We can now state the constraint satisfaction problem for de Bruijn graphs.

Definition 2.5 (Generalized de Bruijn Graph Coloring, [3])**.** The problem is defined with respect to the infinite family of extended de Bruijn graphs and is parametrized by five (fixed) finite objects:

(1) a finite color-set $\mathcal{C}$,
(2) an extraction function $f_{\mathsf{extract}} : \mathcal{C} \to \{0,1\} \cup \{\perp\}$,
(3) a finite vertex-type set $\mathcal{V}$,
(4) a type-coloring constraint $f_{\mathsf{color}} : \mathcal{V} \times \mathcal{C}^3 \to \{0,1\}$, and
(5) a uniform $\mathbf{NC}_1$ family of type-assignments $f_{\mathsf{v\text{-}type}} = \{f^{(t)}_{\mathsf{v\text{-}type}} : \mathcal{V}_t \to \mathcal{V}\}_{t \in \mathbb{N}}$ (that is, a logspace machine that on input 1^t, outputs a boolean formula containing $f^{(t)}_{\mathsf{v\text{-}type}}$).

Given an input $y \in \{0,1\}^*$, the problem is to determine whether, for $t = \lceil \log_2 |y| \rceil$, there exists a coloring $C \colon V_t \to \mathcal{C}$ such that the following two conditions hold.

(1) For all vertices v in $\mathcal{G}_t$, $f_{\mathsf{color}}(f_{\mathsf{v\text{-}type}}(v), C(v), C(\Gamma_{i,0}(v)), C(\Gamma_{i,1}(v))) = 0$.
    In such a case, we say that C *satisfies the coloring constraints*, which are induced by $f_{\mathsf{v\text{-}type}}$ and f_{color}.
(2) For all $1 \le i \le K = |y|$, $y_i = f_{\mathsf{extract}}(C(v_i))$, where v_i is vertex number $2 \cdot 2^t + i$ in layer 0 of $\mathcal{G}_t$.
    In such a case, we say that C is *consistent with the input* (or *with* y).

This graph-coloring problem turns out to be universal. In other words, we can reduce to it from the bounded-halting problem. We now state and prove the reduction.

Definition 2.6. For a Turing machine M, the *bounded halting problem* BH_M is defined as the set of tuples (y, t) such that the machine M accepts y within 2^t steps (where t is written in binary).

Theorem 2.7 (Universality of Generalized de Bruijn Graph Coloring, [3])**.** For any Turing machine M, there exist finite parameters $\mathcal{C}$, f_{extract}, $\mathcal{V}$, f_{color}, and $f_{\mathsf{v\text{-}type}}$, such that the bounded-halting problem for M can be reduced via the identity mapping to the corresponding Generalized de Bruijn Graph Coloring problem, and that for every $y \in \{0,1\}^K$, where $K < 2^t$, machine M halts on y within 2^t steps if and only if there exists a coloring $C : V_t \to \mathcal{C}$ that satisfies the given constraints while being consistent with y.

The proof adapts a nearly linear-time oblivious Turing machine simulation from Pippenger and Fischer [8] to give a reduction.

While the proof is quite technical, at a high level it works by first reducing to a particular constraint satisfaction problem and then reducing that to de Bruijn coloring. The constraint satisfaction problem is formulated in terms of functions $\mathbb{T}_0, \ldots, \mathbb{T}_t$ and $\mathbb{H}_0, \ldots, \mathbb{H}_t$, where the $\mathbb{T}_i$ represent partial configurations in the course of computation and the $\mathbb{H}_i$ represent the relations between the $\mathbb{T}_i$. In particular, $\mathbb{T}_i(j, \cdot)$ represents a tape window of length 2^{t-i} after tape position $(j+k) \cdot 2^{t-i}$ for all $k \geq 0$.

Given this universality result, we can reduce from an $\mathbf{NTIME}(t(n))$ language to an extended de Bruijn graph of size $t(n) \cdot \log^{O(1)}(t(n))$. The paper then goes on to describe an arithmetization technique to complete the proof of Theorem 2.1.

3. PCP Composition

In this paper, Ben-Sasson et al. [4] introduce a new notion of robust soundness. This, along with a construct of a certain efficient **PCPP** (**PCP** of Proximity), allows for more seamless composition of **PCP**s, leading to more efficient **PCP**s than other methods that require the use of parallelization.

First, in order to more easily work with composing PCPs, it is useful to talk about a **PCP** as having a verifier V such that $V(x)$ generates query positions I and a decision circuit D. Then V accepts if $D(\pi|_I) = 1$. In other words, a verifier can be viewed as generating a predicate on the query bits. This is useful later because we can now work with the circuit D and positions I that are generated.

We begin with some definitions.

Definition 3.1. We define $\mathsf{CIRCUIT\text{-}VALUE} \coloneqq \{(C, w) \mid C(w) = 1\}$ for circuits C and inputs w.

Essentially, we check whether a circuit C on some input w evaluates to 1. Observe that we use $\mathsf{CIRCUIT\text{-}VALUE}$ as a pair language. Clearly, $\mathsf{CIRCUIT\text{-}VALUE} \in \mathbf{P}$, but using some more clever tricks we construct more efficient verifiers.

Definition 3.2. For $s, \rho \colon \mathbb{N} \to [0,1]$, a **PCP** verifier V for L has robust-soundness error s with robustness parameter ρ if for every $x \notin L$ and every oracle π, the bits read by V are ρ-close to being accepted with probability less than s. Formally, for all $x \notin L$ and for all proofs π,

$$\Pr[\text{there exists } a \text{ such that } V^a(x) = 1 \text{ and } \Delta(a, \pi) \leq \rho] < s(|x|).$$

Observe that the robust soundness condition and **PCPP** soundness condition naturally work well together. We see that when some soundness conditions hold, a robust **PCP** and a **PCPP** compose nicely together.

Using these definitions, we state the composition theorem.

Theorem 3.3 (Composition Theorem [4])**.** If

(1) L has a robust **PCP** verifier V_{out} with randomness complexity r_{out}, decision complexity d_{out}, robust-soundness error $1 - \varepsilon_{\text{out}}$ and robustness ρ_{out};
(2) CIRCUIT-VALUE has a **PCPP** verifier V_{in} with randomness complexity r_{in}, query complexity q_{in}, decision complexity d_{in}, proximity parameter δ_{in} and soundness error $1 - \varepsilon_{\text{in}}$; and
(3) $\delta_{\text{in}}(d_{\text{out}}(n)) \leq \rho_{\text{out}}(n)$ for all $n \in \mathbb{N}$,

then L has a **PCP** with randomness $r_{\text{out}}(n) + r_{\text{in}}(d_{\text{out}}(n))$, query $q_{\text{in}}(d_{\text{out}}(n))$, decision $d_{\text{in}}(d_{\text{out}}(n))$ and soundness error $1 - \varepsilon_{\text{out}}(n)\varepsilon_{\text{in}}(d_{\text{out}}(n))$.

Proof. From the definitions of **PCPP**s and robust soundness, it is very natural to compose the two. In particular, the requirement that $\delta_{\text{in}}(d_{\text{out}}(n)) \leq \rho_{\text{out}}(n)$ for all n is exactly what we need to ensure the soundness agrees in the composition.

We denote oracle queries to the ith position of the outer verifiers proof oracle as (out, i) and queries to the jth position in the Rth possible proof oracle for the inner verifier as (R, j). The composition is done by constructing a verifier that does the following.

(1) Let $R \in \{0,1\}^{r_{\text{out}}}$ be chosen randomly.
(2) Run $V_{\text{out}}(x; R)$ to get $I_{\text{out}} = (i_1, \ldots, i_{q_{\text{out}}})$ and D_{out}.
(3) Run $V_{\text{in}}(D_{\text{out}})$ to get $I_{\text{in}} = (j_1, \ldots j_{q_{\text{in}}})$ and D_{in}.
(4) For each l, we determine the queries of the composed verifier.
    (a) If j_l is a query to the input oracle, set $k_l = (\text{out}, i_{j_l})$. Thus, V_{in}'s queries to the input oracle are redirected to bits of the proof oracle of V_{out}.
    (b) If j_l is a query to the proof oracle, set $k_l = (R, j_l)$. Thus, V_{in}'s queries to the Rth possible proof are redirected to locations in V_{out}'s proof.
(5) Output D_{in} and $I_{\text{comp}} = (k_1, \ldots, k_{q_{\text{in}}})$.

First, we see by inspection that the randomness complexity, decision complexity, and query complexity are as desired.

Next, we analyze the completeness. By the completeness of V_{out} and V_{in}, for any $x \in L$, there exist proofs for both that make them accept with probability 1. By inspection of the redirection procedure, we see that these proofs can be combined and there exists a proof that makes the combined machine accept with probability 1.

Finally, we analyze the soundness. Let $x \notin L$ and π be any oracle. By the robust soundness of V_{out}, with probability greater than ε_{out}, $\pi_{\text{out}}|_{I_{\text{out}}}$ is ρ_{out}-far from satisfying D_{out}. The **PCPP** soundness of V_{in} then gives that $V_{\text{in}}(D_{\text{out}})$ rejects the oracle with probability greater than ε_{in}. Therefore, we reject with probability at least $\varepsilon_{\text{out}}\varepsilon_{\text{in}}$. □

The composition theorem is useful for complexity results because it gives better bounds when composing, compared to other composition techniques that require parallelization.

In order to make use of the composition theorem, we require a good **PCPP** verifier for CIRCUIT-VALUE. The main construct of the paper is the following **PCPP** that can be used for composition and is highly efficient.

Theorem 3.4. There exists a universal constant c such that for all $n, m \in \mathbb{N}, 0 < \delta, \gamma < 1/2$ satisfying $n^{1/m} \geq m^{cm}/(y\delta)^3$ and $\delta \leq \gamma/c$, CIRCUIT-VALUE has a robust **PCP** of proximity (for circuits of size n) with

- *randomness* $\left(1 - \frac{1}{m}\right) \log n + O(m \log m) + O(\log \log n) + O(\log(1/\delta))$;
- *decision complexity* $n^{1/m} \cdot \text{poly}(\log n, 1/\delta)$, which also upper bounds the query complexity;
- *perfect completeness*; and
- for *proximity parameter* δ, the verifier has robust-soundness error $\Omega(\gamma)$ with robustness parameter $(1 - \gamma)\delta$.

This main construct of deciding CIRCUIT-VALUE is abstracted into the following problem (by modifying a construct from a previous paper). Let $\mathbb{F}$ be a field. Given oracles $F_i \colon \mathbb{F}^m \to \mathbb{F}, i \in \{1, \ldots, t = \text{polylog}\, n\}$, the verifier must test that each of the F_is are close to an m-variate polynomial of low degree and the polynomials satisfy some consistency properties that F_i is locally consistent with F_{i-1}. As it is highly technical, we omit the construction, but we refer the reader to [4].

4. Quasilinear **PCP**s

4.1. **Univariate algebraic constraint satisfaction problem.** We now consider **PCP**s with quasilinear length. We say that a complexity function $t \colon \mathbb{N} \to \mathbb{N}$ is quasilinear if $t(n) = n \cdot \text{polylog}(n)$. Ben-Sasson and Sudan [6] proved Theorem 1.4, that any decision problem decidable in nondeterministic time $O(t(n))$ also has a **PCP** with perfect completeness and soundness $1/2$ that uses $\log(t(n) \cdot \text{polylog}(t(n)))$ random bits and queries $\text{polylog}(t(n))$ bits of a proof. The proof of this theorem is twofold: first a reduction to a **NTIME**-complete language ALGEBRAIC-CSP$_{k,d}$ introduced in Definition 4.2 (Proposition 4.3), and second a **PCP** verifier for ALGEBRAIC-CSP$_{k,d}$ with the required completeness and soundness (Proposition 4.12).

Definition 4.1. Let $\mathbb{F}$ be a field, and let $H \subseteq \mathbb{F}$. Let $\{f_1, \ldots, f_{k'}\}$ be a collection of k' affine maps, where $f_i(x) = a_i x + b_i$ for $a_i, b_i \in \mathbb{F}$. Finally, let $C \colon \mathbb{F}^{k'+1} \to \mathbb{F}$ be a polynomial of degree at most $|H|$ in the first variable. We say the tuple

$$(\mathbb{F}, \{f_1, \ldots, f_{k'}\}, H, C)$$

is an instance of ALGEBRAIC-CSP if and only if there exists a polynomial $P \in \mathbb{F}[x]$ such that $\deg(P) \leq |H| - 1$ and

$$C(x, P(f_1(x)), \ldots, P(f_{k'}(x))) = 0$$

for all $x \in H$.

Definition 4.2. Let $k, d \in \mathbb{N}$. Let the restriction ALGEBRAIC-CSP$_{k,d}$ be the collection of all instances of ALGEBRAIC-CSP where $k' \leq k$ and the degree of C in all but the first variable is at most d.

We now show that ALGEBRAIC-CSP is **NTIME**-complete.

Proposition 4.3. There exists $k, d \in \mathbb{N}$ such that for any complexity function $t\colon \mathbb{N} \to \mathbb{N}$ and any language $L \in \mathbf{NTIME}(t(n))$, L is reducible to $\mathsf{ALGEBRAIC\text{-}CSP}_{k,d}$ in time $\mathrm{poly}(t(n))$.

The proof of Proposition 4.3 uses a reduction to a graph coloring problem.

4.2. **de Bruijn graphs and coloring.** For ease of notation, we define the cyclic permutation operator $\sigma\colon \{0,1\}^k \to \{0,1\}^k$ as follows. If $w = (w_1, \ldots, w_k) \in \{0,1\}^k$, let $\sigma(w) \coloneqq (w_k, w_1, \ldots, w_{k-1})$. We also let $\oplus$ define the bitwise xor operator, and we let $e_i \in \{0,1\}^k$ be the sequence with 1 at the ith coordinate and 0 everywhere else.

In Section 3 we give the definition of the de Bruijn graph. Here we define a similar notion called the wrapped de Bruijn graph, first introduced by Spielman [9].

Definition 4.4. Let $k \in \mathbb{N}$, and let m be the smallest power of 2 such that $m > 5k$. The *k-dimensional wrapped de Bruijn graph* is a directed graph B_k with vertices

$$V = \left\{(w, i) \mid w \in \{0,1\}^k, i \in \{0, \ldots, m-1\}\right\}$$

and edges $(\sigma(w), (i{+}1 \bmod m))$ and $(\sigma(w) \oplus e_1, (i{+}1 \bmod m))$ for any $v = (w, i) \in V$.

Note that each vertex $v = (w, i) \in V$ is connected to two neighbors. We denote by $N_1(v) = (\sigma(w), (i+1 \bmod m))$ and $N_2(v) = (\sigma(w) \oplus e_1, (i+1 \bmod m))$ the first and second neighbors of v, respectively. We now define the wrapped de Bruijn graph coloring problem.

Definition 4.5. Let $B_k = (V, E)$ be a k-dimensional wrapped de Bruijn graph, and let $\widehat{C} = \left\{\widehat{C}_v \mid v \in V\right\}$ be a collection of constraints on the vertices V, where $\widehat{C}_v\colon \{0,1\}^{12} \to \{0,1\}$.

We say that the pair comprising of the graph and the constraints $\left(B_k, \widehat{C}\right)$ is an instance of DE-BRUIJN-COLORING if there exists some assignment $\widehat{A}\colon V \to \{0,1\}^4$ such that

$$\widehat{C}_v\left(\widehat{A}(v), \widehat{A}(N_1(v)), \widehat{A}(N_2(v))\right) = 1$$

for all $v \in V$.

Intuitively, the constraint $\widehat{C}_v$ of a vertex $v \in V$ takes in three arguments, namely v and its neighbors $N_1(v)$ and $N_2(v)$, and is satisfied ($\widehat{C}_v = 1$) if and only if the coloring given by $\widehat{A}$ satisfies the *coloring problem*. That is, v and $N_1(v)$ have distinct colors and v and $N_2(v)$ have distinct colors. If this is possible for all vertices v, then the assignment $\widehat{A}$ is a solution to $\widehat{C}$ for the graph B_k.

One of the most important properties of DE-BRUIJN-COLORING is that it is **NP**-complete. The proof involves two reductions. The first is a $\mathrm{poly}(t(n))$-time reduction from a language $L \in \mathbf{NTIME}(t(n))$ to CKTSAT, the circuit satisfiability problem. The second is a $\mathrm{poly}(t(n))$-time reduction from CKTSAT to DE-BRUIJN-COLORING. We refer the reader to [9] for the details. In particular, we have the following.

Lemma 4.6. For any complexity function $t\colon \mathbb{N} \to \mathbb{N}$, a instance of $L \in \mathbf{NTIME}(t(n))$ is $\mathrm{poly}(t(n))$-time reducible to an instance $\left(B_k, \widehat{C}\right) \in$ DE-BRUIJN-COLORING, where $k = \lceil \log(t(n) \cdot O\left(\log(t(n))^2\right)) \rceil$.

We remark that k is our desired number of random bits.

4.3. **Reducing from DE-BRUIJN-COLORING to ALGEBRAIC-CSP.** Recall that an instance of ALGEBRAIC-CSP involves a collection of affine maps $\{f_1, \ldots, f_{k'}\}$. Affine graphs are natural encodings of affine maps over a field $\mathbb{F}$ into a graph-like object.

Definition 4.7. Let $\mathbb{F}$ be a finite field, and let $\mathcal{A}$ be the set of all affine maps over $\mathbb{F}$. The *affine graph* $G(\mathbb{F}, \mathcal{A})$ over $\mathbb{F}$ is the directed graph with vertices $V = \mathbb{F}$ and edges $(v, f(v))$ for all $v \in V$ and all $f \in \mathcal{A}$.

Recall from field and Galois theory that we may write any field $\mathrm{GF}\left(2^\ell\right)$ as the quotient ring $\mathrm{GF}(2)[x]/(q(x))$, where $(q(x))$ is a maximal ideal generated by an irreducible polynomial q of degree ℓ. Furthermore, recall that a polynomial $p \in \mathrm{GF}(2)[x]$ of degree ℓ is primitive if it is the minimal polynomial over $\mathrm{GF}(2)$ of a primitive element of $\mathrm{GF}\left(2^\ell\right)$. It follows that the polynomials $p_i(x) \coloneqq x^i \pmod{p(x)}$ are such that $\deg(p_i) < \ell$ and are pairwise distinct for all $i \in \left\{1, \ldots, 2^\ell\right\}$. For an overview of these facts, we refer the reader to Artin §15.7 [2] and Lidl and Niederreiter §3.1 [7].

We now embed a de Bruijn graph B_k into an affine graph over the field $\mathrm{GF}\left(2^\ell\right)$ for any integer $\ell > k + \log(5k) + 2$.

Proposition 4.8. Let $k \in \mathbb{N}$, let m be the smallest power of 2 such that $m > 5k$, and let $\ell > k + \log(5k) + 2$ be some integer. Let $\mathrm{GF}\left(2^\ell\right) = \mathrm{GF}(2)[x]/(q(x))$ for some irreducible polynomial q of degree ℓ. Let $p(x) \in \mathrm{GF}(2)[x]$ be a primitive polynomial of degree $s = \log m \in \mathbb{N}$, and let $p_i(x) \coloneqq x^i \pmod{p(x)}$. For $(w, i) \in V$, $w \in \{0,1\}^k$, and $i \in \{0, \ldots, m-1\}$, the map $f\colon V(B_k) \to \mathrm{GF}\left(2^\ell\right)$ given by

$$f(w, i) = x^s \cdot \sum_{j=1}^{k} w_j x^j + p_i(x) \tag{1}$$

is an injective graph homomorphism from B_k to $G\left(\mathrm{GF}\left(2^\ell\right), \mathcal{A}\right)$, where

$$\mathcal{A} = \left\{ f_{a,b,c,d}(x) \coloneqq ax + bp(x) + cx^{s+1} + dx^{s+k+1} \;\middle|\; a \in \mathrm{GF}\left(2^\ell\right)[x]; b, c, d \in \{0,1\} \right\}.$$

Proof. We first show that f is injective. Recall that $\deg(p_i) < s$ for all $i \in \{1, \ldots, m\}$, but taking $j = 1$ in the definition of $f(w, i)$ in (1) gives $\deg(f) \geq s + 1$. Therefore, in f the second polynomial p_i contributes terms of degree $< s$, whereas the first polynomial contributes terms of degree $\geq s + 1$. To prove injectiveness, it suffices to show that each component of f is injective. The first polynomial is injective by definition (since if $w \neq w'$ then the coefficients w_j differ), and the second polynomial p_i is injective by our earlier discussion that the p_is are pairwise distinct.

To show that f is a graph homomorphism, we need only show that the edges are mapped correctly by f. Recall that a vertex $v = (w, i) \in V$ of B_k shares edges with $N_1(v)$ and $N_2(v)$. By definition of an affine graph, we need to show that if (v, v') is an

edge of B_k, then $f(v') = f_{f(v),b,c,d}$ for some $b, c, d \in \{0,1\}$. We proceed by casework on the two vertices.

Case 1 Suppose $v' = N_1(v) = (\sigma(w), (i+1 \bmod m))$. Then

$$f(\sigma(w),(i+1 \bmod m)) = \begin{cases} x^{s+1} \cdot \sum_{j=1}^{k} w_j x^j & \text{if } w_k = 0, \\ x^{s+1} \cdot \sum_{j=1}^{k} w_j x^j + x^{s+k+1} + x^{s+1} & \text{if } w_k = 1, \end{cases} + \begin{cases} xp_i(x) & \text{if } \deg(p_i) < s-1, \\ xp_i(x) + p(x) & \text{if } \deg(p_i) = s-1. \end{cases}$$

Case 2 Suppose $v' = N_2(v) = (\sigma(w) \oplus e_1, (i+1 \bmod m))$. Then

$$f(\sigma(w)\oplus e_1,(i+1 \bmod m)) = \begin{cases} x^{s+1} \cdot \sum_{j=1}^{k} w_j x^j + x^{s+1} & \text{if } w_k = 0, \\ x^{s+1} \cdot \sum_{j=1}^{k} w_j x^j + x^{s+k+1} & \text{if } w_k = 1, \end{cases} + \begin{cases} xp_i(x) & \text{if } \deg(p_i) < s-1, \\ xp_i(x) + p(x) & \text{if } \deg(p_i) = s-1. \end{cases}$$

In both cases we have clear choices of b, c, and d. This completes the proof. □

Now we are ready to prove Proposition 4.3. This proof involves *arithmetizing* an instance of $\left(B_k, \widehat{C}\right)$. Our choice of $k = 10$ and $d = 16$ suffices. Let $G\left(\mathrm{GF}\left(2^\ell\right), \mathcal{A}\right)$ be the affine graph that is the image of our B_k under f.

Recall that $\mathrm{GF}\left(2^\ell\right) = \mathrm{GF}(2)[x]/(q(x))$, where q is an irreducible polynomial of degree ℓ. We construct the instance

$$\phi = \left(\mathrm{GF}\left(2^\ell\right), \{f', f''\} \cup \mathcal{A}, H, C(x, y_0, y_1, z_{000}, \ldots, z_{111})\right)$$

as follows. Denote the set $I = \{f(v) \mid v \in V\} \subseteq \mathrm{GF}\left(2^\ell\right)$ for vertices V of B_k. Note there exists a $\zeta \in \mathrm{GF}\left(2^\ell\right)$ such that $(\zeta + I) \cap I = \varnothing$. Such a $\zeta \in \mathrm{GF}\left(2^\ell\right)$ exists from elementary field theory since interpreting $\mathrm{GF}\left(2^\ell\right) = \mathrm{GF}(2)[x]/(q(x))$ as polynomials with coefficients in $\mathrm{GF}(2)$ modulo $q(x)$, all elements in I have degree at most $k+s$.

Given ζ, we now define $f'(x) \coloneqq x$ and $f''(x) \coloneqq \zeta - x$ as part of our instance ϕ. Furthermore, we define $H \coloneqq (\zeta + I) \cup I$. All that we have left to define is the polynomial $C\colon \mathrm{GF}\left(2^\ell\right)^{k+1} \to \mathrm{GF}\left(2^\ell\right)$.

For each vertex v in G, we construct the polynomial $C_v(y, z_{b_0}, z_{b_1})$ of degree at most 15 in each variable that agrees with $\widehat{C}_v \in C$ on all elements in $\{0,1\}^{12}$, where $b_0, b_1 \in \{0,1\}^3$. This choice of $b_0 = (b,c,d)$ and $b_1 = (b',c',d')$ comes from the choices of coefficients $f(N_1(v)) = f_{f(v),b,c,d}$ and $f(N_2(v)) = f_{f(v),b',c',d'}$, respectively, as in the proof of Proposition 4.8.

Let $P_h(x)$ be the polynomial of degree $|H|-1$ such that $P_h(h) = 1$ and 0 elsewhere. Furthermore, let $P_{\{0,1\}^4}$ be the polynomial of degree $d = 16$ with roots $\{0,1\}^4$. Then define our polynomial $C\colon \mathrm{GF}\left(2^\ell\right)^{11} \to \mathrm{GF}\left(2^\ell\right)$ by

$$C(x, y_0, y_1, z_{000}, \ldots, z_{111}) = \sum_{v \in I} P_v(x) \cdot C_v(y_0, z_{b_0(v)}, z_{b_1(v)}) + \sum_{h \in H \setminus I} P_h(x) \cdot P_{\{0,1\}^4}(y_1).$$

Note that the first sum is 0 if all the coloring constraints $\widehat{C}$ are satisfied, and the second sum is 0 if all vertices $v \in V$ received a coloring assignment. Therefore, C is 0 if both are satisfied. Furthermore, if we denote by $\deg_{x_i}(C)$ the degree of the ith variable of C, then one readily checks that $\deg_x(C) \leq |H| - 1$ and $\deg_t(C) \leq d = 16$ for $t \in \{y_0, y_1, z_{000}, \ldots, z_{111}\}$.

This reduction is done in time poly $\left(2^\ell\right)$. One can readily check that the instance $\phi \in \mathsf{ALGEBRAIC\text{-}CSP}_{k,d}$ if and only if $\left(B_k, \widehat{C}\right) \in \mathsf{DE\text{-}BRUIJN\text{-}COLORING}$; our choice of polynomial P such that $C(x, P(f_1(x)), \ldots, P(f_k(x))) = 0$ for all $x \in H$ is precisely the coloring assignment $\widehat{A}\colon \mathrm{GF}\left(2^\ell\right) \to \{0,1\}^4$ of $\left(B_k, \widehat{C}\right) \in \mathsf{DE\text{-}BRUIJN\text{-}COLORING}$.

We remark that an adaption of this reduction for the pair language gives

$$100(kd+1)(|H|-1) < 2^\ell \leq 200(kd+1)(|H|-1). \tag{2}$$

We refer the reader to [4] for the proof.

4.4. **Constructing a PCP for** $\mathsf{ALGEBRAIC\text{-}CSP}$**.** With our reduction in-hand, to prove Theorem 1.4 it suffices to give a $(\log(t(n) \cdot \mathrm{polylog}(t(n))), \mathrm{polylog}(t(n)))$-**PCP** verifier with perfect completeness and soundness $1/2$ for $\mathsf{ALGEBRAIC\text{-}CSP}$. In particular, our reduction reduces some $x \in \{0,1\}^*$ of length $|x| = n$ to some tuple $\phi = \left(\mathrm{GF}\left(2^\ell\right), \{f_1, \ldots, f_k\}, H, C\right)$ of quasilinear length $n \cdot \mathrm{polylog}(n)$, and we wish to show that $\phi \in \mathsf{ALGEBRAIC\text{-}CSP}_{k,d}$ when $x \in L \in \mathbf{NTIME}(t(n))$. Recall from Definitions 4.1 and 4.2 that $\phi \in \mathsf{ALGEBRAIC\text{-}CSP}_{k,d}$ if there exists some polynomial $P \in \mathrm{GF}\left(2^\ell\right)[x]$ such that

(I) $\deg(P) \leq |H| - 1$ and
(II) $C(x, P(f_1(x)), \ldots, P(f_k(x))) = 0$ for all $x \in H$.

With this, the idea of our desired **PCP** verifier is as follows. The proof consists of two polynomials $p_1, p_2 \in \mathrm{GF}\left(2^\ell\right)[x]$. The **PCP** verifier needs to check that p_1 is our desired P that satisfy conditions (I) and (II). In other words, the **PCP** verifier needs to verify that

(i) $\deg(p_1) \leq |H| - 1$,
(ii) p_2 vanishes on H (i.e. $p_2(x) = 0$ for all $x \in H$), and
(iii) $p_2(x) = C(x, p_1(f_1(x)), \ldots, p_1(f_k(x)))$ for a randomly chosen $x \in \mathrm{GF}\left(2^\ell\right)$ (i.e. C and p_2 agree).

For the **PCP** verifier to perform checks on (i) and (ii), we introduce Reed-Solomon codes and their respective **PCP**s of proximity.

Definition 4.9. Let $P(x) \in \mathbb{F}[x]$ be a polynomial over a finite field $\mathbb{F}$. The *evaluation* of P over $S \subseteq \mathbb{F}$ is defined as the function $p\colon S \to \mathbb{F}$ given by $p(x) = P(x)$ for all $x \in S$. The *Reed-Solomon code of degree at most d over $\mathbb{F}$ at $S \subseteq \mathbb{F}$* is

$$\mathrm{RS}(\mathbb{F}, S, d) \coloneqq \{p\colon S \to \mathbb{F} \mid p \text{ is an evaluation over } S \text{ of some } p \in \mathbb{F}[x],\ \deg(p) \leq d\}.$$

Definition 4.10. Let $P(x) \in \mathbb{F}[x]$ be a polynomial over a finite field $\mathbb{F}$. Let $H, S \subseteq \mathbb{F}$ and $d \in \mathbb{N}$. The *H-vanishing Reed-Solomon code* is

$$\mathrm{VRS}(\mathbb{F}, S, H, d) \coloneqq \{p \in \mathrm{RS}(\mathbb{F}, S, d) \mid p \text{ vanishes on } H\}.$$

Evidently, Reed-Solomon codes are crucial in our design of the **PCP** verifier. More importantly, there exist **PCPP**-verifiers of quasilinear length for both RS and VRS when $F = \mathrm{GF}\left(2^\ell\right)$. Formally, Definition 1.3 defines **PCPP**s with pairs (x, y), so we need to adapt RS to VRS to its corresponding "pair" languages. We omit the details and refer to reader to [6].

Proposition 4.11. There exist

$$\mathbf{PCPP}\left(\log(n \cdot \mathrm{polylog}(n)), \mathrm{polylog}(n), \mathsf{HAMMING}_{\mathrm{GF}\left(2^\ell\right)}\right)$$

verifiers with soundness s and proximity δ for both RS and VRS.

The construction is technical, so we omit the proof of Proposition 4.11 for the sake of brevity and refer the reader to §6 of [6].

With our **PCPP**-verifiers, our construction of the **PCP** verifier for Theorem 1.4 becomes clear, as with our previous discussion.

Proposition 4.12. There exists a $(\log(t(n) \cdot \mathrm{polylog}(t(n))), \mathrm{polylog}(t(n)))$-verifier for $\mathsf{ALGEBRAIC\text{-}CSP}_{k,d}$ with perfect completeness and soundness $1/2$.

Proof. We construct our **PCP**-verifier. Recall that our quasilinear-length instance is

$$\phi = \left(\mathrm{GF}\left(2^\ell\right), \{f_1, \ldots, f_k\}, H, C\right).$$

The verifier receives a proof π comprising of

- a polynomial $p_1 \in \mathrm{GF}\left(2^\ell\right)[x]$ and a proof of proximity π_1 to the Reed-Solomon code $\mathrm{RS}\left(\mathrm{GF}\left(2^\ell\right), \mathrm{GF}\left(2^\ell\right), |H| - 1\right)$ and
- a polynomial $p_2 \in \mathrm{GF}\left(2^\ell\right)[x]$ and a proof of proximity π_2 to the vanishing Reed-Solomon code $\mathrm{VRS}\left(\mathrm{GF}\left(2^\ell\right), \mathrm{GF}\left(2^\ell\right), H, (kd+1)(|H|-1)\right)$.

Note that these **PCPP**-verifiers exist by Proposition 4.11, and the size of π_1 and π_2 are linear in $\left|\mathrm{GF}\left(2^\ell\right)\right|$, or quasilinear in $t(n)$. Hence the overall proof π is quasilinear in $t(n)$. As before, the **PCP**-verifier verifies (i) and (ii) as follows. The verifier tosses $\log(t(n) \cdot \mathrm{polylog}(t(n)))$ random coins and then

(1) verify that the **PCPP**-verifier for the Reed-Solomon code accepts p_1 on the arguments $\left(\mathrm{GF}\left(2^\ell\right), \mathrm{GF}\left(2^\ell\right), |H| - 1\right)$ with proof of proximity π_1 using proximity $\delta = 1/(10k)$ and soundness $s = 1/2$; and
(2) verify that the **PCPP**-verifier for the vanishing Reed-Solomon code accepts p_2 on the arguments $\left(\mathrm{GF}\left(2^\ell\right), \mathrm{GF}\left(2^\ell\right), H, (kd+1)(|H|-1)\right)$ with proof of proximity π_2 using proximity $\delta = 1/100$ and soundness $s = 1/2$.

Again, step 1 probabilistically verifies that p_1 is a polynomial over $\mathrm{GF}\left(2^\ell\right)$ of degree at most $|H| - 1$, which is exactly (i). Similarly, step 2 probabilistically verifies that p_2 is a polynomial over $\mathrm{GF}\left(2^\ell\right)$ of degree at most $|H| - 1$ that vanishes on H, which is exactly (ii). Finally, we

(3) select a random $x \in \mathrm{GF}\left(2^\ell\right)$; query $p_1(x), p_1(f_1(x)), \ldots, p_1(f_k(x)), p_2(x)$; and verify if

$$p_2(x) = C(x, p_1(f_1(x)), \ldots, p_1(f_k(x))),$$

which is exactly (iii). The **PCP**-verifer accepts if and only if steps 1, 2, and 3 all return true. This **PCP**-verifier has the necessary randomness complexity of $\log(t(n) \cdot \text{polylog}(t(n)))$ and query complexity of $\text{polylog}(t(n))$.

Next, we show that this **PCP**-verifier has perfect completeness. Suppose our instance satisfies $\phi \in \textsf{ALGEBRAIC-CSP}_{k,d}$. By definition, there exists a polynomial $P \in \text{GF}\left(2^\ell\right)[x]$ such that (I) and (II) hold. Let p_1 be the evaluation of P on $\text{GF}\left(2^\ell\right)$. By definition,

$$p_1 \in \text{RS}\left(\text{GF}\left(2^\ell\right), \text{GF}\left(2^\ell\right), |H| - 1\right). \tag{3}$$

Next, define

$$T(x) \coloneqq C(x, P(f_1(x)), \ldots, P(f_k(x))).$$

By definition, T vanishes on H. Denote by $\deg_{x_i}(C)$ the degree of C in variable x_i. Note that

$$\deg(T) \le \deg_{x_1}(C) + \sum_{i=1}^{k} \deg_{x_{i+1}}(C) \deg(P) \le (kd+1)(|H|-1)$$

because $\deg(P) \le |H| - 1$, $\deg_{x_1}(C) \le |H|$, and $\deg_{x_{i+1}}(C) \le d$ for all $i \in \{1, \ldots, k\}$, by definition. Let p_2 be the evaluation of T. Therefore,

$$p_2 \in \text{VRS}\left(\text{GF}\left(2^\ell\right), \text{GF}\left(2^\ell\right), H, (kd+1)(|H|-1)\right). \tag{4}$$

Both (3) and (4) imply the existence of proofs of proximity π_1 and π_2 to p_1 and p_2, which causes steps 1 and 2 of the verification process to return true by Proposition 4.11. Thus, it suffices to prove that for a random $x \in \text{GF}\left(2^\ell\right)$ the polynomials p_2 and T agree on x. However, this holds by construction, since for all $x \in \text{GF}\left(2^\ell\right)$ we have

$$p_2(x) = T(x) = C(x, P(f_1(x)), \ldots, P(f_k(x))) = C(x, p_1(f_1(x)), \ldots, p_1(f_k(x)))$$

because p_1 and p_2 are the evaluations of P and T, respectively. Thus, step 3 is verified as well, so our proof π causes the **PCP**-verifier to accept, as required.

Finally, we verify that the **PCP**-verifier has soundness $1/2$. Suppose our instance satisfies $\phi \notin \textsf{ALGEBRAIC-CSP}_{k,d}$. Then we show that our protocol fails in at least one of step 1, 2, and 3 with high probability.

Case 1 Suppose p_1 is $1/(10k)$-far from $\text{RS}\left(\text{GF}\left(2^\ell\right), \text{GF}\left(2^\ell\right), |H| - 1\right)$. Then Proposition 4.11 implies that the **PCPP**-verifier rejects with probability at most $1/2$ in step 1.

Case 2 Suppose p_2 is $1/100$-far from $\text{VRS}\left(\text{GF}\left(2^\ell\right), \text{GF}\left(2^\ell\right), H, (kd+1)(|H|-1)\right)$. Then Proposition 4.11 implies that the **PCPP**-verifier rejects with probability at most $1/2$ in step 2.

Case 3 Suppose steps 1 and 2 passed, so

$$p_1 \text{ is } 1/(10k)\text{-close to } \text{RS}\left(\text{GF}\left(2^\ell\right), \text{GF}\left(2^\ell\right), |H| - 1\right), \tag{5}$$

$$p_2 \text{ is } 1/100\text{-close to } \text{VRS}\left(\text{GF}\left(2^\ell\right), \text{GF}\left(2^\ell\right), H, (kd+1)(|H|-1)\right). \tag{6}$$

Let P^* be the polynomial such that $\deg(P^*) \leq |H| - 1$ that is closest to p_1. Define $T\colon \mathrm{GF}\big(2^\ell\big) \to \mathrm{GF}\big(2^\ell\big)$ by

$$T(x) \coloneqq C(x, P^*(f_1(x)), \ldots, P^*(f_k(x))).$$

Similarly, define $S\colon \mathrm{GF}\big(2^\ell\big) \to \mathrm{GF}\big(2^\ell\big)$ by

$$S(x) \coloneqq C(x, p_1(f_1(x)), \ldots, p_1(f_k(x))).$$

Finally, let T' be the polynomial closest to p_2. To calculate the probability that step 3 falsely accepts, we calculate how many entries T and T' agree on; therefore, we can determine the probability on a random $x \in \mathrm{GF}\big(2^\ell\big)$. Clearly $T \neq T'$ because if they agree, then T vanishes on H, which means that P^* is our desired polynomial, contrary to our initial assumption that $\phi \notin \mathsf{ALGEBRAIC\text{-}CSP}_{k,d}$.

Note that p_1 is $1/(10k)$-close to P^* by (5). The union bound implies that S is $1/10$-close to T. Furthermore, (6) implies that p_2 is $1/100$-close to T'. Finally, T and T' agree on at most $1/100$ of their entries because $\deg(T), \deg(T') \leq (kd+1)(|H|-1) < 2^\ell/100 = \left|\mathrm{GF}\big(2^\ell\big)\right|/100$, where the $1/100$ constant comes from (2). Therefore, T and T' agree on less than $1/100$ of their entries. The total probability of acceptance is thus $1/10 + 1/100 + 1/100 < 1/2$, which completes our soundness condition.

The proof is finished. □

This finishes the proof of Theorem 1.4.

References

[1] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*, Cambridge University Press, Cambridge, UK, 2012.

[2] M. Artin. *Algebra*, 2nd ed. Prentice Hall, Hoboken, NJ, 2010.

[3] E. Ben-Sasson, O. Goldreich, P. Harsha, M. Sudan, and S. Vadhan. *Short* **PCP***s Verifiable in Polylogarithmic Time*, Proceedings of the 20th Annual IEEE Conference on Computational Complexity (2005) 120–134.

[4] E. Ben-Sasson, O. Goldreich, P. Harsha, M. Sudan, and S. Vadhan. *Robust* **PCP***s of Proximity, Shorter* **PCP***s, and Applications to Coding*, SIAM Journal of Computing **36**(4) (2006), 889–974.

[5] E. Ben-Sasson and M. Sudan. *Simple PCPs with polylog rate and query complexity*, Proceedings of the 37th annual ACM symposium on Theory of computing, Baltimore, Maryland, 21–24 May 2005, pp. 266–275.

[6] E. Ben-Sasson and M. Sudan. *Short* **PCP***s with Polylog Query Complexity*, SIAM Journal of Computing **38**(2) (2008) 551–607.

[7] R. Lidl and H. Niederreiter. *Finite Fields*, 2nd ed., Cambridge University Press, Cambridge, UK, 1996.

[8] N. Pippenger and M. J. Fischer. *Relations among complexity measures*, Journal of the ACM, 26 (1979), pp. 361–381.

[9] D. Spielman. *Computationally Efficient Error-Correcting Codes and Holographic Proofs*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 1995.

Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, MA 02139-4301
Email address: allenees@mit.edu

Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, MA 02139-4301
Email address: leognon@mit.edu

Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, MA 02139-4301
Email address: psvenk@mit.edu