Nearly-linear Size Holographic Proofs

Alexander Polishchuk*
Moscow Independent University

Daniel A. Spielman†
MIT

Abstract

We show how to construct holographic (or transparent) proofs of size $n^{1+\epsilon}$ that can be checked by a verifier that is allowed to read only $O(1)$ bits of the proof and has access to $O(\log n)$ random bits, for all $\epsilon > 0$. In general, we construct proofs of size $n^{1+2^{-O(q(n))}}(\log n)^{O(q(n))}$ checkable by the query of $2^{O(q(n))}$ bits, for any $q(n) = O(\log\log n)$. An essential element of our construction is a proof that the low-degree test used by Arora and Safra [AS92] is effective on domains of size linear in the degree of the encoded polynomial.

1. Introduction

Babai, Fortnow, Levin and Szegedy [BFLS91] introduce the notion of a holographic (or transparent) proof: a proof whose validity can be probabilistically checked by examining a few randomly chosen bits of the proof. They describe a system whereby any length-n proof of a theorem can be converted into a length-$n^{1+\epsilon}$ holographic proof of the theorem that can be checked by examining $(\log n)^{1/\epsilon+O(1)}$ bits of the proof. Focusing on different parameters suggested by [FGL+91], Arora and Safra [AS92] define the class $PCP(r(n), q(n))$ of languages that have proofs of membership which can be probabilistically checked by a polynomial-time verifier that has access to $O(r(n))$ random bits and is allowed to examine $O(q(n))$ bits of the proof.

In their celebrated paper, Arora *et al.* [ALM+92] show that $NP = PCP(\log n, 1)$, which means that there exist holographic proofs of membership in any NP language that can be checked by examining only a constant number of bits of the proof. However, their process of converting a proof into a holographic proof produces a super-quadratic increase in size. We construct holographic proofs that combine the advantages of small size with constant-query checkability. We prove that $NP = PCP(\log n, 1)$ with proofs of size $n^{1+\epsilon}$ for any $\epsilon > 0$. In general, we construct proofs of size $n^{1+2^{-O(q(n))}}(\log n)^{O(q(n))}$ checkable by the query of $2^{O(q(n))}$ bits, for any $q(n) = O(\log\log n)$. For simplicity of exposition, we will explain our results in terms of proofs of language membership, and discuss the framework of [BFLS91] at the end of the paper.

Requiring that a proof be holographic can be viewed as requiring that a set of data have a special format such that one can probabalistically check whether the data has been properly formatted by examining it in only few places. Recent results showing that low-degree polynomials are efficiently checkable, self-test/correctable, and randomly-self-reducible [BLR90, Rub90, GLR+91, GS92, Lip91, BF90, Sud92] demonstrate that multivariate polynomials have presentations that naturally possess such a format. Thus, it is not surprising that the checkability of holographic proofs has rested on the checkability of multivariate polynomials.

Arora and Safra [AS92] achieved a major breakthrough when they demonstrated that one could efficiently test and self-correct presentations of multivariate polynomials that were of size cubic in the degree of the polynomials. Sudan [Sud92] obtained a quadratic bound. The improvement of this bound to linear in Section 5 is an important element of our construction of nearly-linear size holographic proofs. One of the advantages of our techniques is that they enable us to present an elementary, self-contained explanation of why this maximum-degree test works as well as it does.

Another key ingredient of our construction is the use in Section 6 of a reduction from the problem of veri-

*polisch@math.utah.edu.

†Dept. of Applied Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139. spielman@math.mit.edu.
Partially supported by the Fannie and John Hertz Foundation, Air Force Contract F49620-92-J-0125 and NSF grant 9212184CCR.

fying that a circuit has an accepting input to a problem of coloring a graph related to a de Bruijn graph. [BFLS91] demonstrated that any computation can be reduced to a coloring problem on a related graph, and our reduction can be thought of as a special case of their theorem, but with a simpler proof. We provide an algebraic description of a de Bruijn graph that enables us to identify the nodes of the graph with points in a vector space in such a way that the names of the neighbors of a node can be expressed as a low-degree polynomial in the name of that node. It seems that our technique of using a low-degree polynomial to address the neighbors of a node is actually necessary for the $n^{1+\epsilon}$ proof size claimed in [BFLS91].

Our holographic proofs have a recursive structure. We first construct holographic proofs that can be checked by making a constant number of queries of size roughly $\sqrt{n}$. We then observe that the checks can be performed by circuits of size $\sqrt{n}\log^{O(1)} n$. For each level of recursion, we can reduce the size of the queries by a square-root, while only increasing the proof size by a polylogarithmic factor. We terminate the recursion by applying the circuit-verifiers of [ALM+92].

2. Polynomial Codes

This and the next three Sections will be devoted to proving the effectiveness of the bivariate maximum-degree test. We view the presentations of multivariate polynomials that appear in the holographic proof literature as error-correcting codes. We begin by explaining this perspective.

The simplest polynomial code is the Reed-Solomon code [MS77], which can be described in the following fashion: Let $\mathcal{F}$ be a field and let $\{x_1, \ldots, x_n\}$ be a subset of $\mathcal{F}$. The messages to be transmitted will be the polynomials of degree d with coefficients in $\mathcal{F}$. The encoding of a polynomial $p(x)$ will be the list of values that $p(x)$ has at $x_1, \ldots, x_n$. That is,

$$E(p(x)) = (p(x_1), p(x_2), \ldots, p(x_n)).$$

Because any two distinct degree d polynomials can agree in value in at most d places, any two distinct codewords must differ in at least $n-d$ places.

The generalization of Reed-Solomon codes to multivariate polynomials has been an important building block in the construction of holographic proofs. In a *maximum degree* code of dimension m and degree d, the messages correspond to polynomials $p(x_1, \ldots, x_m)$ such that the degree of p in each variable is at most d. We encode p by writing down the list of values of p at each point in some set $\mathcal{H}^m$ where $\mathcal{H} \subseteq \mathcal{F}$ and $|\mathcal{H}| > d$. The advantage of these codes over Reed-Solomon codes is that one can verify that a list of values represents a valid encoding by performing many tests where each test only looks at a small portion of the list. We will show in Proposition 2 that in order to check that a list of values corresponds to the values obtained from some maximum degree d bivariate polynomial, it suffices to check that the list looks like a degree d polynomial in each row and column.

In this paper, we will consider bivariate polynomials over a domain $X \times Y$, where $X = \{x_1, \ldots, x_n\} \subseteq \mathcal{F}$ and $Y = \{y_1, \ldots, y_n\} \subseteq \mathcal{F}$. The following definition will be helpful:

Definition 1. A polynomial $p(x,y)$ has *degree* (d,e) if it has degree at most d in x and degree at most e in y.

Proposition 2 (Well-known). *Let $f(x,y)$ be a function on $X \times Y$ such that for $1 \leq j \leq n$, $f(x,y_j)$ agrees on X with some degree d polynomial in x, and for $1 \leq i \leq n$, $f(x_i,y)$ agrees on Y with some degree e polynomial in y. Then, there exists a polynomial $P(x,y)$ of degree (d,e) such that $f(x,y)$ agrees with $P(x,y)$ everywhere on $X \times Y$.*

Proof: Recall that a degree d univariate polynomial is uniquely determined by its values at $d+1$ points. For $1 \leq j \leq e+1$, let $p_j(x)$ be the degree d polynomial that agrees with $f(x,y_j)$. For $1 \leq j \leq e+1$, let $\delta_j(y)$ be the degree e polynomial in y such that

$$\delta_j(y_k) = \begin{cases} 1, & \text{if } j = k, \text{ and} \\ 0, & \text{if } 1 \leq k \leq e+1, \text{ but } j \neq k. \end{cases}$$

We let $P(x,y) = \sum_{j=1}^{e+1} \delta_j(y) p_j(x)$. It is clear that P has degree (d,e). Moreover, $P(x,y_j) = f(x,y_j)$ for all $x \in X$ and $1 \leq j \leq d+1$. To see that in fact $P(x,y) = f(x,y)$ for all $(x,y) \in X \times Y$, observe that P and f agree at $e+1$ points in column y. Since f agrees with some degree e polynomial in column y, that polynomial must be the restriction of P to column y. □

The maximum-degree testing lemma of [AS92] and its successors can be interpreted as saying that if a list of values resembles a degree d polynomial on most points of most rows and columns, then the list must be close to the list of values obtained from some degree (d,d) polynomial.

The efficient *total-degree* codes used in [ALM+92] are obtained by combining the maximum-degree testing lemma of [AS92] with total-degree testing techniques from [RS92]. The improvements in maximum-degree testing made in Section 5 can similarly be combined with the techniques of [RS92] to obtain even better total-degree codes.

3. The First Step

Let $C(x,y)$ be a polynomial of degree (n,d) and let $R(x,y)$ be a polynomial of degree (d,n) [1] such that

$$\operatorname*{Prob}_{(x,y)\in X\times Y}[R(x,y)\neq C(x,y)]\leq \delta^2.$$

In Theorem 9, we will show that there exists a polynomial $Q(x,y)$ of degree (d,d) such that

$$\operatorname*{Prob}_{(x,y)\in X\times Y}[R(x,y)\neq Q(x,y) \text{ or } C(x,y)\neq Q(x,y)]$$
$$< 2\operatorname*{Prob}_{(x,y)\in X\times Y}[R(x,y)\neq C(x,y)].$$

As in [Sud92], we begin by finding a low-degree "error correcting" polynomial that is zero whenever R and C disagree.

Lemma 3. *Let $S\subset X\times Y$ be a set of size at most $(\delta n)^2$, where δn is an integer. Then there exists a non-zero polynomial $E(x,y)$ of degree $(\delta n,\delta n)$ such that $E(x,y)=0$ for all $(x,y)\in S$.*

Proof: The set of polynomials of degree $(\delta n,\delta n)$ is a vector space of dimension $(\delta n+1)^2$. Consider the map that sends a polynomial to the vector of values that it takes for each point in S. That is, let $S=\{s_1,\ldots,s_m\}$ and consider the map

$$\phi: E(x,y)\mapsto (E(s_1),E(s_2),\ldots,E(s_m)).$$

This map is a homomorphism of a vector space of dimension $(\delta n+1)^2$ into a vector space of lesser dimension; so, there must be a non-zero polynomial in the vector space of polynomials of degree $(\delta n,\delta n)$ that evaluates to zero at every point in S. □

For the remainder of the paper, we will always assume that δn is an integer.

Let S be the subset of $X\times Y$ on which R and C disagree. By Lemma 3, we can choose $E(x,y)$ so that

$$R(x,y)E(x,y)=C(x,y)E(x,y) \text{ for all } (x,y)\in X\times Y.$$

Moreover, $C(x,y)E(x,y)$ is a polynomial of degree $(n+\delta n, d+\delta n)$ and $R(x,y)E(x,y)$ is a polynomial of degree $(d+\delta n, n+\delta n)$. By Proposition 2, there exists a polynomial $P(x,y)$ of degree $(d+\delta n, d+\delta n)$ such that

$$R(x,y)E(x,y)=C(x,y)E(x,y)=P(x,y) \qquad (1)$$

for all $(x,y)\in X\times Y$.

We would like to divide P by E as formal polynomials and conclude the proof. However, the most we can say is that

$$\frac{P(x,y)}{E(x,y)}=R(x,y)=C(x,y),$$

for all $(x,y)\in X\times Y$ such that $E(x,y)\neq 0$.

The next two sections will be devoted to showing that if n is sufficiently large, then E does in fact divide P. We will begin with one small step:

Lemma 4. *Let $P(x,y)$, $E(x,y)$, $R(x,y)$ and $C(x,y)$ be polynomials of degrees $(\delta n+d,\delta n+d)$, $(\delta n,\delta n)$, (d,n) and (n,d) respectively such that (1) holds. If $|X|>\delta n+d$ and $|Y|>\delta n+d$, then for all $y_0\in Y$ and for all $x_0\in X$, $P(x,y_0)\equiv R(x,y_0)E(x,y_0)$ and $P(x_0,y)\equiv C(x_0,y)E(x_0,y)$.*

Proof: For fixed y_0, $P(x,y_0)$ and $R(x,y_0)E(x,y_0)$ are degree $\delta n+d$ polynomials that have the same value on at least $d+\delta n+1$ points, so $P(x,y_0)\equiv R(x,y_0)E(x,y_0)$ as formal polynomials. □

4. Resultants

In this section, we will review some standard facts about resultants. A more complete presentation can be found in [Lan93, vdW53]. We note that Sudan [Sud92] introduced the idea of using the resultant to prove that E divides P.

Let $\mathcal{F}$ be a field and let

$$\begin{aligned} P(x) &= P_0+P_1x+\cdots+P_dx^d, \qquad \text{and}\\ E(x) &= E_0+E_1x+\cdots+E_ex^e \end{aligned}$$

be polynomials in x with coefficients in the field $\mathcal{F}$.

Proposition 5. *$P(x)$ and $E(x)$ have a non-trivial common factor if and only if there exist polynomials $A(x)$ of degree $e-1$ and $B(x)$ of degree $d-1$ such that $P(x)A(x)-E(x)B(x)=0$.*

Proof: If there exist $F(x)$, $\hat{P}(x)$ and $\hat{E}(x)$ such that $\deg F(x)\geq 1$, $P(x)=F(x)\hat{P}(x)$, and $E(x)=F(x)\hat{E}(x)$, then we can choose $A(x)=\hat{E}(x)$ and $B(x)=\hat{P}(x)$.

To go the other direction, assume that such $A(x)$ and $B(x)$ exist. Since the degree of $P(x)$ is greater than the degree of $B(x)$, $P(x)$ and $E(x)$ must share a common factor. □

We can reformulate this as a system of linear equations in the coefficients of A and B:

$$\begin{aligned} P_dA_{e-1} &= E_eB_{d-1}\\ P_{d-1}A_{e-1}+P_dA_{e-2} &= E_{e-1}B_{d-1}+E_eB_{d-2}\\ \vdots\quad &\vdots\quad \vdots\\ P_0A_0 &= E_0B_0 \end{aligned}$$

If we treat the coefficients of A and B as the variables of a system of linear equations, then we find that the

[1] Any function on a domain of size n can be represented by a polynomial of degree $n-1$.

above equations have a solution if and only if the matrix $M(P,E)$ has determinant zero, where $M(P,E) =$

$$\left(\begin{array}{cccccccc} P_d & P_{d-1} & \cdots & \cdots & P_0 & 0 & \cdots & 0 \\ 0 & P_d & & \cdots & P_1 & P_0 & \cdots & 0 \\ \vdots & \ddots & \ddots & & & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & P_d & \cdots & \cdots & P_1 & P_0 \\ E_e & E_{e-1} & \cdots & E_0 & 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & & & & \ddots & & \vdots \\ \vdots & & \ddots & & & & \ddots & 0 \\ 0 & \cdots & & 0 & E_e & E_{e-1} & \cdots & E_0 \end{array}\right) \begin{array}{l} \left.\vphantom{\begin{array}{c}0\\0\\0\\0\end{array}}\right\} e \text{ rows} \\ \left.\vphantom{\begin{array}{c}0\\0\\0\\0\end{array}}\right\} d \text{ rows} \end{array}$$

We now define $R(P,E)$, the resultant of P and E, to be the polynomial in the coefficients of P and E obtained by taking the determinant of $M(P,E)$. We obtain:

Proposition 6. *The polynomials $P(x)$ and $E(x)$ share a common factor if and only if $R(P,E) = 0$.*

The following fact about the derivative of the determinant of a matrix of polynomials will play a crucial role in our proof: Let $M(x) = (p_{i,j}(x))_{i,j}$ be an n-by-n matrix of polynomials in x over $\mathcal{F}$ and let $R(x)$ be the determinant of $M(x)$.

Proposition 7. *$R'(x)$, the derivative of $R(x)$, can be expressed as*

$$\begin{aligned} R'(x) \quad = \quad & \begin{vmatrix} p'_{1,1}(x) & p'_{1,2}(x) & \cdots & p'_{1,n}(x) \\ p_{2,1}(x) & p_{2,2}(x) & \cdots & p_{2,n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,1}(x) & p_{n,2}(x) & \cdots & p_{n,n}(x) \end{vmatrix} \\ & + \cdots + \begin{vmatrix} p_{1,1}(x) & p_{1,2}(x) & \cdots & p_{1,n}(x) \\ p_{2,1}(x) & p_{2,2}(x) & \cdots & p_{2,n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ p'_{n,1}(x) & p'_{n,2}(x) & \cdots & p'_{n,n}(x) \end{vmatrix} \end{aligned}$$

5. Piecing it Together

Since the propositions of the previous section concerned univariate polynomials, you may be wondering how we are going to apply them to bivariate polynomials. The idea is to treat the polynomials $P(x,y)$ and $E(x,y)$ as polynomials in y over $\mathcal{F}(x)$, the field of rational functions in x. $\mathcal{F}(x)$ is the field comprising terms of the form $p(x)/q(x)$, where $p(x)$ and $q(x)$ are polynomials in $\mathcal{F}$. It is easy to verify that this is in fact a field.

We can now consider P and E as polynomials in y with coefficients in $\mathcal{F}(x)$ by writing

$$\begin{aligned} P(x,y) &= P_0(x) + P_1(x)y + \cdots + P_{\delta n+d}(x)y^{\delta n+d} \\ E(x,y) &= E_0(x) + E_1(x)y + \cdots + E_{\delta n}(x)y^{\delta n}. \end{aligned}$$

We will show that E divides P as a polynomial in y over the field $\mathcal{F}(x)$. By Gauss' Lemma[2], this implies that E divides P over $\mathcal{F}[x]$, the ring of *polynomials* in x, which means that $E(x,y)$ divides $P(x,y)$.

We will begin our proof by dividing E and P by their greatest common divisor. If that greatest common divisor is not E, then we obtain two polynomials with no common factor. To obtain a contradiction, we will show that these two polynomials have a common factor when considered as polynomials in y over $\mathcal{F}(x)$. By Gauss' Lemma, this will imply that they share a common factor when considered as polynomials in x and y.

Lemma 8. *Let $E(x,y)$ be a polynomial of degree (b,a) and let $P(x,y)$ be a polynomial of degree $(b+d, a+d)$. If there exist distinct $x_1, \ldots, x_n$ such that $E(x_i,y)$ divides $P(x_i,y)$ for $1 \leq i \leq n$, distinct $y_1, \ldots, y_n$ such that $E(x,y_i)$ divides $P(x,y_i)$ for $1 \leq i \leq n$ and if*

$$n > \min\{2b+2d, 2a+2d\},$$

then $E(x,y)$ divides $P(x,y)$.

Proof: Assume, without loss of generality, that $a \geq b$. Let $F(x,y)$ be the largest common factor of $P(x,y)$ and $E(x,y)$. Assume by way of contradiction that $F \not\equiv E$ and that $F(x,y)$ has degree (f,e). Set

$$P(x,y) \equiv \hat{P}(x,y)F(x,y) \quad \text{and} \quad E(x,y) \equiv \hat{E}(x,y)F(x,y).$$

We will now divide P and E by F and apply the lemma to $\hat{P}$ and $\hat{E}$. The conditions of the lemma are satisfied by $\hat{P}$ and $\hat{E}$ because $n - f > 2b + 2d - f \geq 2(b-f) + 2d$. (We need to take $n - f$ because $F(x,y)$ could be the zero polynomial for as many as f values of x.) Thus, we can assume without loss of generality that $P(x,y)$ and $E(x,y)$ have no common factors. We will use this assumption to obtain a contradiction. Write

$$\begin{aligned} P(x,y) &\equiv P_0(x) + P_1(x)y + \cdots + P_{a+d}(x)y^{a+d} \\ E(x,y) &\equiv E_0(x) + E_1(x)y + \cdots + E_a(x)y^a, \end{aligned}$$

and form the matrix $M(P,E)(x) =$

$$\left(\begin{array}{cccccc} P_{a+d}(x) & \cdots & \cdots & P_0(x) & \cdots & 0 \\ \vdots & \ddots & & & \ddots & \vdots \\ 0 & \cdots & P_{a+d}(x) & \cdots & \cdots & P_0(x) \\ E_a(x) & \cdots & E_0(x) & 0 & \cdots & 0 \\ \vdots & \ddots & & & \ddots & 0 \\ 0 & \cdots & 0 & E_a(x) & \cdots & E_0(x) \end{array}\right) \begin{array}{l} \left.\vphantom{\begin{array}{c}0\\0\\0\end{array}}\right\} a \\ \left.\vphantom{\begin{array}{c}0\\0\\0\end{array}}\right\} a+d \end{array}$$

[2] The usual statement of Gauss' Lemma is that if a polynomial with integer coefficients can be factored over the rationals, then it can be factored over the integers. A proof of Gauss' Lemma can be found in most undergraduate algebra textbooks.

$R(P,E)(x)$, the resultant of P and E, is the determinant of $M(P,E)(x)$ and can therefore be viewed as a polynomial in x. $M(P,E)(x)$ has a rows of coefficients of P and $a+d$ rows of coefficients of E, so $R(P,E)(x)$ will be a polynomial of degree at most $a(b+d)+(a+d)b$. We will show that $R(P,E)(x)$ is in fact the zero polynomial by demonstrating that it has more than $a(b+d)+(a+d)b$ roots.

For $1 \leq i \leq n$, $E(x_i, y)$ divides $P(x_i, y)$, so we can see that the first a rows of $M(P,E)(x_i)$ are dependent on the last $a+d$ rows of $M(P,E)(x_i)$. This implies that $M(P,E)(x_i)$ is a matrix of rank at most $a+d$ (actually, the rank is exactly $a+d$). By Proposition 7, the k-th derivative of $R(P,E)(x)$ at x_i is the sum of determinants of matrices of rank at most $a+d+k$. Since $M(P,E)(x)$ is a matrix of side $2a+d$, $R^{(k)}(P,E)(x_i)$ is zero for $k < a$. That is, $R(P,E)(x)$ has a zero of multiplicity a at each of $x_1, \ldots, x_n$. Because we assumed that

$$na > 2ab + 2ad \geq 2ab + d(a+b) = a(b+d) + (a+d)b,$$

$R(P,E)(x)$ must be the zero polynomial. Applying Proposition 6, we see that E and P must have a non-trivial common factor when considered as polynomials in y over $\mathcal{F}(x)$, which is a contradiction. □

This lemma is a variation on Bezout's Theorem. In fact, the proof is similar to the proof of Bezout's Theorem in [vdW53]. We can now prove:

Theorem 9 (Bivariate Testing). *Let $\mathcal{F}$ be a field, let $X = \{x_1, \ldots, x_n\} \subseteq \mathcal{F}$, and let $Y = \{y_1, \ldots, y_n\} \subseteq \mathcal{F}$. Let $R(x,y)$ be a polynomial over $\mathcal{F}$ of degree (d,n) and let $C(x,y)$ be a polynomial over $\mathcal{F}$ of degree (n,d). If*

$$\operatorname*{Prob}_{(x,y)\in X\times Y}[R(x,y) \neq C(x,y)] < \delta^2,$$

and $n > 2\delta n + 2d$, then there exists a polynomial $Q(x,y)$ of degree (d,d) such that

$$\operatorname*{Prob}_{(x,y)\in X\times Y}[R(x,y) \neq Q(x,y) \text{ or } C(x,y) \neq Q(x,y)] < 2\delta^2.$$

Proof: Let S be the set of points such that $R(x,y) \neq C(x,y)$. By Lemma 3, there exists a polynomial $E(x,y)$ of degree $(\delta n, \delta n)$ such that S is contained in the zero set of E. By Lemmas 4 and 8, there exists a polynomial $Q(x,y)$ of degree (d,d) such that

$$R(x,y)E(x,y) = C(x,y)E(x,y) = Q(x,y)E(x,y),$$

for all $(x,y) \in X \times Y$.

This implies that in any row on which $E(x,y)$ is non-zero, Q agrees with R on that entire row. However, E can be identically zero on at most δn rows; so, E must be non-zero on at least $(1-\delta)n$ rows. Thus, Q must agree with R on at least $(1-\delta)n$ rows. We can similarly show that Q must agree with C on at least $(1-\delta)n$ columns. We could stop now, content in the knowledge that R and C agree on the intersection of $(1-\delta)n$ columns and rows; however, we will show that they agree on many more points.

As before, let S be the set of points at which $R(x,y) \neq C(x,y)$. Let T be the set of points at which $R(x,y) = C(x,y)$, but $Q(x,y) \neq R(x,y)$. We will show that $|T| \leq |S|$, which will complete the proof of the theorem. Call the rows on which R disagrees with Q *bad* and define *bad* columns similarly. Let b_r be the number of bad rows and let b_c be the number of bad columns. Call *good* any row or column that is not bad. We will say that a row and column disagree if R and C take different values at their intersection. We first observe that there can be at most $d + b_r$ points of T in any bad column: if a column has more than $d + b_r$ points of T, then it must have at least $d+1$ points in good rows at which C agrees with R and therefore Q, implying that that column is in fact good. Thus, every bad column must have at least $n/2$ points of S in the intersection of that column with the good rows. We can analyze the rows similarly to see that $|T| \leq |S|$. (the basic idea is that the points of T must lie in the lower left-hand corner of Figure 1). □

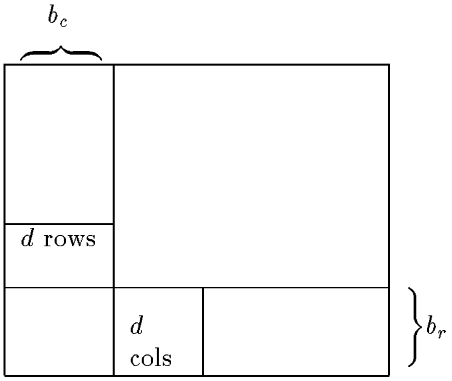

Figure 1: The arrangement of bad rows and columns.

We will make use of the following variation of Theorem 9:

Theorem 10. *There exists a constant k such that the following holds: Let $\mathcal{F}$ be a field and let $X = \{x_1, \ldots, x_n\} \subseteq \mathcal{F}$, $Y = \{y_1, \ldots, y_n\} \subseteq \mathcal{F}$, and $Z = \{z_1, \ldots, z_m\} \subseteq \mathcal{F}$. Let $P_1(x,y,z)$, $P_2(x,y,z)$ and $P_3(x,y,z)$ be polynomials over $\mathcal{F}$ of degrees (d,n,m), (n,d,m) and (n,n,d_z) respectively. If*

$$\text{Prob}[P_1(x,y,z) = P_2(x,y,z) = P_3(x,y,z)] > 1 - \delta,$$

and $n > k\delta n + kd$ and $m > k\delta m + kd_z$, then there exists a polynomial $Q(x, y, z)$ of degree (d, d, d_z) such that

$$\text{Prob}[Q(x,y,z) = P_1(x,y,z) = P_2(x,y,z) = P_3(x,y,z)]$$

$$> 1 - 4\delta,$$

where the probabilities are taken over $(x, y, z) \in X \times Y \times Z$.

Proof: [Sketch] It is not difficult to prove a version of Theorem 9 for the case that X and Y are of different sizes, as long as the ratio of the size of X to the size of Y is the same as the ratio of the degree of R to the degree of C: you just need to construct an error-correcting polynomial that is unbalanced in the same ratio. To go from a bivariate theorem to a trivariate theorem, first apply the unbalanced bivariate theorem in every plane perpendicular to the x or y-axis. We thereby obtain a polynomial $R(x, y)$ of degree (d, n) and a polynomial $C(x, y)$ of degree (n, d) over $\mathcal{F}[z]$ that satisfy the conditions of Theorem 9. □

6. A Graph Coloring Problem

Consider a boolean circuit C with m binary gates and n inputs. One way to represent such a circuit is to draw a picture containing the inputs, the gates, and lines representing the edges that connect them (See Figure 2). We will show how such a picture can be "drawn" on any network on which one can perform routing with merging. We will draw a circuit with m gates and n inputs on a leveled routing network of $(5r + 1)2^r$ nodes, where $2^r \geq n + m$. The routing network will consist of $(5r + 1)$ levels of 2^r nodes, where node i in one level will be connected to node j in the next if (i, j) is an edge of a de Bruijn graph [Lei92] (See Figure 3).

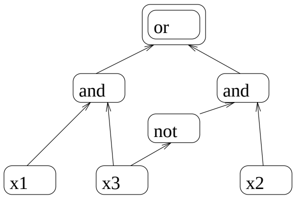

Figure 2: A drawing of a circuit

We will associate each input and gate in the circuit with two nodes in the same row of the routing network, one in the first level and one in the last. View each wire of the circuit as a packet to be routed from the first-level node representing its origin to the last-level node representing its destination. Allow every internal node to act as a switch, or, if it has only one incoming message, to send it to both outputs. Using standard packet-routing techniques [Lei92], we can find switching actions for each node that establish collision-free paths in the graph for each packet (Figure 3 presents such a description of the circuit that appears in Figure 2).

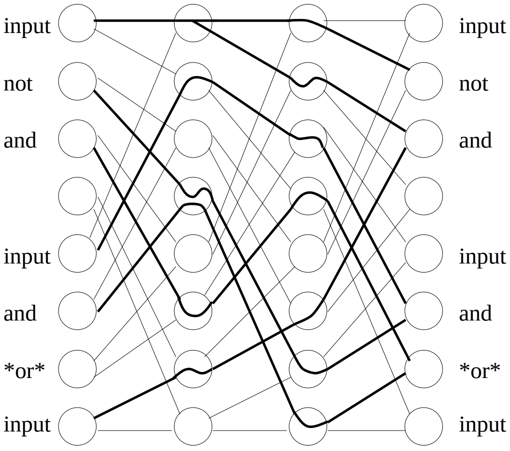

Figure 3: A (toy) de Bruijn router

We will call this description of the circuit a *coloring* of the nodes of the routing network. Because each node is identified as being a gate, the output gate, an input, or one of 4 switches (depending on how you count them), we will need only a small, fixed number of colors [3]. We will call a coloring of the network that represents a circuit a *theorem candidate.* A proof of the theorem candidate will be an assignment of 0's, 1's and blanks to the inputs, gates, and wires of the circuit that is both a valid computation and that causes the circuit to output 1 (See Figure 4). This can be described as a coloring of the nodes by writing, for each node, the values on the node's incoming and outgoing wires. Thus, a proof can also be described using a finite set of colors.

In order to check that a proof candidate is actually a proof of a theorem candidate, it suffices to check that

1. for each node, the 0's, 1's and blanks assigned by the proof candidate to the node and its neighbors have been switched in accordance with the action assigned to the node by the theorem candidate, and

2. for each gate, the output of the gate assigned by the proof candidate on the first level is what the gate would compute on the inputs provided by the proof candidate on the last level.

[3] It is interesting to observe that this description of a circuit differs in size from a more conventional description by only a constant factor. In a conventional description, one must assign a name to each input and gate, which uses $\log m$ bits per name. In this description, it is unnecessary to assign names to gates and inputs, because their connections are described by the routing network.

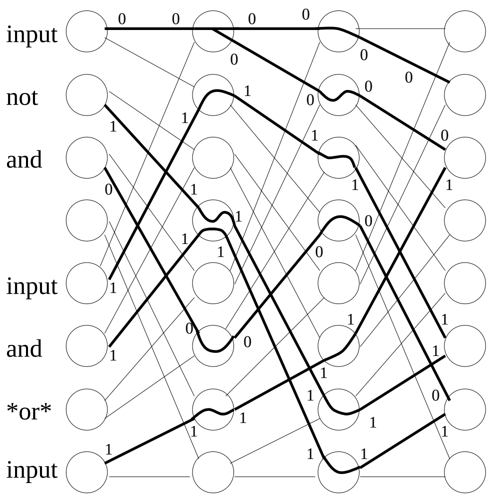

Figure 4: A proof

These conditions will comprise our *coloring rules.* Our graph coloring problem is: Given a first coloring of the graph (a theorem candidate), find a second coloring that satisfies all the coloring rules (a proof candidate).

7. The Arithmetization

In this section, we provide an algebraic description of the routing network presented in Section 6. We will first define a directed graph, G_n, on the elements of the field $GF(2^n)$. Let α be a primitive element of $GF(2^n)$ (*i.e.* an element such that $\alpha^{2^n-1} = 1$ and $\alpha^p \neq 1$ for any $0 < p < 2^n - 1$). The set of edges of G_n will be

$$\{(\gamma, \alpha\gamma) : \gamma \in GF(2^n)\} \cup \{(\gamma, \alpha\gamma + 1) : \gamma \in GF(2^n)\}\,.$$

Lemma 11. *The graph G_n is isomorphic to the de Bruijn graph on 2^n vertices.*

Proof: [Sketch] Let $p(\alpha) = \alpha^n + c_1\alpha^{n-1} + \cdots + c_n$ be an irreducible polynomial in α over $GF(2)$. Represent the elements of $GF(2^n) \simeq GF(2)[\alpha]/p(\alpha)$ by polynomials of degree at most $n-1$ in α over $GF(2)$. We can construct an isomorphism of the de Bruijn graph with G_n by mapping the vertex $(b_1, \ldots, b_n)$ to

$$\alpha^{n-1}b_1 + \alpha^{n-2}(b_2 + c_1 b_1) + \cdots + \left(b_n + \sum_{i=1}^{n-1} c_i b_{n-i}\right).$$

We now see that the shift-left operation on $(b_1, \ldots, b_n)$ is equivalent to multiplication by α, up to the possible addition of 1. □

Corollary 12. *Let r be even, let $\mathcal{F} = GF(2^{r/2})$, let α be a primitive element of $\mathcal{F}$, and let $\mathcal{E} = \{\alpha^i : 0 \leq i \leq 5r\}$. The graph on $\mathcal{F} \times \mathcal{F} \times \mathcal{E}$ with edge set*

$$\{((x,y,z),(y,\alpha x,\alpha z)) : x,y \in \mathcal{F} \textit{ and } z \in \mathcal{E} - \alpha^{5r}\} \cup$$

$$\{((x,y,z),(y,\alpha x+1,\alpha z)) : x,y \in \mathcal{F} \textit{ and } z \in \mathcal{E} - \alpha^{5r}\}$$

is isomorphic to the routing network described in Section 6. The x and y coordinates label the row of a node, and the z coordinate indicates its level.

For the remainder of the paper, we will define $\mathcal{F}$ and $\mathcal{E}$ as above. We will also define $\mathcal{G} = GF(2^r)$. Note that $\mathcal{F}$ is a subfield of $\mathcal{G}$. It will also be useful to define $\mathcal{E}' = \mathcal{E} - \alpha^{5r}$.

This algebraic description of the routing network enables us to construct an algebraic formulation of the coloring rules. We will identify the colors with a subset of $\mathcal{F}$, say $\{c_1, \ldots, c_k\}$. We will then view a theorem candidate, T, and a proof candidate, P, as functions from $\mathcal{F} \times \mathcal{F} \times \mathcal{E}$ into $\mathcal{F}$. Thus, we can identify a gate as $(x,y) \in \mathcal{F} \times \mathcal{F}$, and determine what type of gate it is supposed to be by examining $T(x,y,1)$. Similarly, $P(x,y,1)$ contains the value that P claims gate (x,y) outputs. To check that P is a proof of T, it suffices to verify that

0. $\theta(P(x,y,z)) = 0$, for all $(x,y,z) \in \mathcal{F} \times \mathcal{F} \times \mathcal{E}$, where $\theta(w) = \prod_{i=1}^{k}(w - c_i)$ is a polynomial that is zero if and only if w is in the set of colors.

1. $\phi(T(x,y,z), P(x,y,z), P(y,\alpha x,\alpha z), P(y,\alpha x+1,\alpha z)) = 0$, for all $(x,y,z) \in \mathcal{F} \times \mathcal{F} \times \mathcal{E}'$, where ϕ is a constant-degree polynomial that, given that its arguments are colors[4], is zero if and only if the patterns of 0's and 1's assigned by P to (x,y,z) and its neighbors have been switched in accordance with the color assigned to (x,y,z) by T (This arithmetizes condition 1 from the previous section).

2. $\psi(T(x,y,1), P(x,y,1), P(x,y,\alpha^{5r})) = 0$, for all $(x,y) \in \mathcal{F} \times \mathcal{F}$, where ψ is a constant-degree polynomial that, given that its arguments are colors, is zero if and only if the output value assigned to (x,y) by $P(x,y,1)$ corresponds to what the gate assigned to (x,y) by $T(x,y,1)$ would compute on the inputs provided by $P(x,y,\alpha^{5r})$ (This arithmetizes condition 2 from the previous section).

We know that the constant-degree polynomials ϕ and ψ exist because they are constrained at only a finite number of values of their arguments. The advantage of being able to traverse edges by applying linear polynomials is that the degree of ϕ is a constant times the degree of P. If we applied polynomials of high-degree, then the degree of ϕ would become too large for us to obtain the compact representation of ϕ that we will use to construct nearly-linear size holographic proofs.

[4] There may be values of P that are elements of $\mathcal{F}$, but which are not valid colors, that cause ϕ to become zero. This is why we include condition 0.

8. Constructing Proofs

In this section, we will show how to construct a size $O(m \log^2 m)$ probabilistically checkable proof of circuit satisfiability that can be checked by the examination of a constant number of segments of the proof, each of size $O(\sqrt{m} \log^2 m)$. In the next section, we will show how to apply this proof system recursively to decrease the query sizes.

In the previous section, we suggested that P and T be viewed as functions from $\mathcal{F} \times \mathcal{F} \times \mathcal{E}$ into $\mathcal{F}$. We may as well view them as polynomials of degree $(2^{r/2} - 1, 2^{r/2} - 1, 5r)$. Because $\mathcal{F} \subset \mathcal{G}$, we can evaluate the polynomials defined by P and T at points of $\mathcal{G}$.

For a polynomial, $P(x, y, z)$, and a domain $\mathcal{D} = \mathcal{D}_1 \times \mathcal{D}_2 \times \mathcal{D}_3 \subseteq \mathcal{G} \times \mathcal{G} \times \mathcal{G}$, we will define a *presentation of P on $\mathcal{D}$* to be

i. a table of the values of P at every point of $\mathcal{D}$, and

ii. a table of the univariate polynomials obtained by restricting P to each axis-parallel line passing through $\mathcal{D}$.

We will eventually select sets $\mathcal{F} \subset \mathcal{H} \subset \mathcal{G}$ of size $O(2^{r/2})$ and $\mathcal{E} \subset \mathcal{K} \subset \mathcal{G}$ of size $O(5r)$, and ask that the proof contain a presentation of P on $\mathcal{H} \times \mathcal{H} \times \mathcal{K}$. The entries in the first table have r bits each and the entries in the second table, which are univariate polynomials of degree at most $(2^{r/2} - 1)$, have at most $r2^{r/2}$ bits each (the polynomials in the z direction are smaller). The presentation of P will have a total of $O(m \log^2 m)$ bits.

In general, we will define a (d_x, d_y, d_z)-presentation on $\mathcal{D}$ to be

i. a table containing one element of $\mathcal{G}$ for each element of $\mathcal{D}$, and

ii. a table of univariate polynomials of degree d_x (or d_y or d_z) for every line parallel to the x-axis (or y or z) through $\mathcal{D}$.

We will say that $\hat{P}$, a (d_x, d_y, d_z)-presentation on $\mathcal{D}$, is ϵ-good if there exists a degree (d_x, d_y, d_z) polynomial P such that the presentation of P on $\mathcal{D}$ differs from $\hat{P}$ in at most ϵ entries in tables i and ii, in which case we say that $\hat{P}$ *represents* P. To test that a presentation is ϵ-good, the verifier will choose a point of $\mathcal{D}$ uniformly at random and verify that the three polynomials provided in the second table of the presentation passing through that point have the value assigned to that point by the first table. For sufficiently large domains, we can use Theorem 10 to prove that if the probability that the presentation passes this test is greater than $(1 - \frac{\epsilon}{k})$, for some constant k, then the presentation is ϵ-good. We can repeat this test a few[5] times to boost our level of confidence that the table is ϵ-good.

The probabalistically checkable proof will contain a presentation of P, along with some additional information that will help the verifier check that P satisfies conditions 0, 1, and 2 of Section 7. The construction is based on the polynomial construction rules introduced in [Sud92]. We will restrict our discussion to the verification that condition 1 is satisfied because it is the most complicated of the three. The proof will consist of presentations of polynomials that we will call P_1, P_2, ϕ_1, ϕ', ϕ'' and ϕ''' on certain domains[6]. The verifier will begin by testing that each of these presentations is ϵ-good for some sufficiently small constant ϵ so that we can hereafter assume that when the verifier queries the presentations at randomly chosen places, it gets the values and restrictions of the polynomials represented.

Let $\phi_1(x, y, z) = \phi(T(x, y, z), P(x, y, z), P(y, \alpha x, \alpha z), P(y, \alpha x + 1, \alpha z))$. The proof will contain a presentation of ϕ_1, a proof that this presentation actually represents ϕ_1, and a proof that ϕ_1 is zero on the desired domain. We will assume that the verifier has access to a presentation of T, either because it has been provided by another source, or because the verifier has computed it.

Let $P_1(x, y, z) = P(y, \alpha x, \alpha z)$ and let $P_2(x, y, z) = P(y, \alpha x + 1, \alpha z)$. The proof will need to provide the verifier access to $P_1(x, y, z)$ and $P_2(x, y, z)$. It will do this by providing presentations of these polynomials on carefully chosen domains. Let $\mathcal{F} \subset \mathcal{I} \subset \mathcal{G}$ be a set of size $c2^{r/2}$, for a constant c which we will choose later, such that $\mathcal{I} + 1 = \mathcal{I}$. Similarly, let $\mathcal{E} \subset \mathcal{J} \subset \mathcal{G}$ be of size $5rc$. We let $\mathcal{H} = (\mathcal{I} \cup \alpha\mathcal{I})$ and $\mathcal{K} = (\mathcal{J} \cup \alpha\mathcal{J})$, so that the proof should present P on $\mathcal{H} \times \mathcal{H} \times \mathcal{K}$ and P_1 and P_2 on $\alpha^{-1}\mathcal{H} \times \mathcal{H} \times \alpha^{-1}\mathcal{K}$.

To check that $P_1(x, y, z) = P(y, \alpha x, \alpha z)$, the verifier should choose a few points $(x, y, z) \in \mathcal{H} \times \mathcal{H} \times \mathcal{K}$ uniformly at random and check that $P(x, y, z) = P_1(\alpha^{-1}y, x, \alpha^{-1}z)$. The verifier should then perform the analogous test for P_2. If the presentations pass these tests then the verifier will be confident that P, P_1 and P_2 satisfy the desired relations.

We now note that $\mathcal{I} \subseteq \mathcal{H} \cap \alpha^{-1}\mathcal{H}$ and that $\mathcal{J} \subseteq \mathcal{K} \cap \alpha^{-1}\mathcal{K}$, and that a presentation of P on $\mathcal{I} \times \mathcal{I} \times \mathcal{J}$ is contained in the presentation of P on $\mathcal{H} \times \mathcal{H} \times \mathcal{K}$. Similarly, the presentations of P_1 and P_2 contain presentations on $\mathcal{I} \times \mathcal{I} \times \mathcal{J}$. Since $|\mathcal{I}| \geq \frac{1}{2}|\mathcal{H}|$ and $|\mathcal{J}| \geq \frac{1}{2}|\mathcal{K}|$, if the original presentations are ϵ-good, then the presentations on $\mathcal{I} \times \mathcal{I} \times \mathcal{J}$ are 8ϵ-good.

The proof should contain a presentation of ϕ_1 on $\mathcal{I} \times$

[5] When we say "few", we mean that there is a constant that would enable us to prove the theorem, but that we will not use precious space to determine that constant.

[6] Marcos Kiwi has pointed out that we can construct the proofs without using P_1 and P_2.

$\mathcal{I}\times\mathcal{J}$. The constant c should be chosen so that $\mathcal{I}$ and $\mathcal{J}$ are large enough that we can apply Theorem 10 to show that we can test that the presentation of ϕ_1 is ϵ-good (we can choose c to be a constant because the degree of ϕ is a constant). To test that ϕ_1 has been properly constructed, the verifier will choose a few points (x,y,z) of $\mathcal{I}\times\mathcal{I}\times\mathcal{J}$ uniformly at random and verify that the values that the first tables of the presentations of T, P, P_1, P_2 and ϕ_1 satisfy the relation

$$\phi_1(x,y,z)=\phi(T(x,y,z),P(x,y,z),P_1(x,y,z),P_2(x,y,z)).$$

If the presentation of ϕ_1 passes this test then the verifier can be confident that the presentation of ϕ_1 represents the desired polynomial.

Let $\mathcal{F}=\{f_1,\ldots,f_{2^{r/2}}\}$. In order for the proof to convince the verifier that $\phi_1(x,y,z)$ is zero on $\mathcal{F}\times\mathcal{F}\times\mathcal{E}'$, it should contain presentations of the polynomials ϕ', ϕ'' and ϕ''' on the domain $\mathcal{I}\times\mathcal{I}\times\mathcal{J}$, where ϕ', ϕ'' and ϕ''' are defined by

$$\begin{aligned}\phi'(x,y,z) &= \textstyle\sum_{i=1}^{2^{r/2}}\phi_1(f_i,y,z)x^{i-1} \qquad (*)\\ \phi''(x,y,z) &= \textstyle\sum_{j=1}^{2^{r/2}}\phi'(x,f_j,z)y^{j-1}\\ \phi'''(x,y,z) &= \textstyle\sum_{l=0}^{5r-1}\phi''(x,y,\alpha^l)z^l\\ &= \textstyle\sum_{i=1}^{2^{r/2}}\sum_{j=1}^{2^{r/2}}\sum_{l=0}^{5r-1}\phi_1(f_i,f_j,\alpha^l)x^{i-1}y^{j-1}z^l.\end{aligned}$$

Thus, $\phi'''\equiv 0$ if and only if $\phi_1(x,y,z)=0$ for all $(x,y,z)\in\mathcal{F}\times\mathcal{F}\times\mathcal{E}'$. To test that ϕ' satisfies the required relation with ϕ_1, the verifier will choose a few lines in $\mathcal{I}\times\mathcal{I}\times\mathcal{J}$ parallel to the x-axis and verify that the restrictions of ϕ_1 and ϕ' to this line satisfy relation (*). The verifier will similarly test that ϕ'' and ϕ''' have been well constructed. If these tests are passed, then the verifier can be confident that the presentation of ϕ''' has been properly derived from ϕ_1. If $\phi'''\not\equiv 0$, then $\phi'''(x,y,z)\neq 0$ for a large constant fraction of the $(x,y,z)\in\mathcal{I}\times\mathcal{I}\times\mathcal{J}$, so we can check that $\phi'''\equiv 0$ by checking that it is zero at a few randomly chosen points of $\mathcal{I}\times\mathcal{I}\times\mathcal{J}$.

We formalize the above discussion in the following lemma.

Lemma 13. *There exists a polynomial time Turing machine V that, when given a random string R of length $O(\log m)$ and a circuit C of size m as input, will output the description of a circuit C_R of size $O(\sqrt{m}\log^2 m)$ such that*

i. C_R expects a sequence of strings S as input, each of which has length at most $O(\sqrt{m}\log m)$ and whose total length is at most $O(m\log^2 m)$, but C_R only reads a constant number of the strings of S,

ii. if there exists an input that causes C to output 1, then there exists a sequence of strings S so that $\mathrm{Prob}_R[C_R(S)=1]=1$, and

iii. if there exists a sequence of strings S such that $\mathrm{Prob}_R[C_R(S)=1]>\frac{2}{3}$, then there exists an input that causes C to output 1.

Proof: We need to examine the size of the circuit C_R needed to perform the tests described in the discussion above. Let C have n inputs and let r be the least integer such that $2^r\geq n+m$. The tests that a presentation is ϵ-good each involved reading a univariate polynomial of degree $O(2^{r/2})$ and evaluating it at a point of $\mathcal{G}$. This can be performed by a circuit of size $O(2^{r/2}r^2)$. The tests that random points of the presentations of T, P, P_1, P_2 and ϕ_1 satisfied the desired relations can be performed by circuits of size $O(r^2)$.

To verify that relations $(*)$ are satisfied, the verifier needs to be able to test that univariate polynomials $p(w)$ and $p'(w)$ satisfy the relation $p'(w)=\sum_{i=1}^{2^{r/2}}p(f_i)w^{i-1}$. To enable the verifier to do this, we will choose special representations of p and p': We will represent p' by listing its coefficients, but we will represent p by listing its values at a set of points containing $\mathcal{F}$ of size one greater than the degree of p. This makes it trivial to check that $p(w)$ and $p'(w)$ satisfy the relation. We note that the size of the circuit needed to evaluate a polynomial at a point does not substantially depend on which representation we use. The verifier will expect that ϕ_1 use the list-of-values description of univariate polynomials in the second table of its presentation, and that the presentation of ϕ' use the coefficient description of the polynomials in x, and the list-of-values description of the polynomials in y and z. We can make similar restrictions on the presentations of ϕ'' and ϕ''' to simplify the rest of the verification process. □

9. Recursion

To apply the proof systems recursively, we will make use of the input-encoding techniques developed in [BFLS91] and [AS92]. The statements of the lemma and theorem of this section have a similar form to statements in [ALM$^+$92], except that we analyze the sizes of the circuits needed to perform the tests that we describe.

For a string x, let $E(x)$ denote the Justesen encoding of x. We will make use of the following standard properties of Justesen codes [MS77]:

1. The length of $E(x)$ is linear in the length of x.

2. There is a size $O(|x|\log^2(|x|))$ circuit that on input x returns $E(x)$.

3. There is a constant ϵ_J such that for $x\neq y$, $d(E(x),E(y))>\epsilon_J$, where d is the hamming distance.

We define $E^{-1}(z)$ to be the lexicographically first string x that minimizes $d(E(x), z)$. We will now prove:

Lemma 14. *There exists a polynomial time Turing machine V_1 that, when given as input a random string R of length $O(\log m)$ and a circuit C of size m that takes $O(1)$ strings $X_1, \ldots, X_k$ as input, will output the description of a circuit C_R of size $O(\sqrt{m} \log^4 m)$ such that*

i. C_R expects two types of inputs: Strings $Y_1, \ldots, Y_k$, which are supposed to encode the input strings of C, and a sequence of strings S, each of which has length at most $O(\sqrt{m} \log^4 m)$ and whose total length is at most $O(m \log^4 m)$, but C_R only reads a constant number of the strings of S, and a constant number of bits from each of $Y_1, \ldots, Y_k$,

ii. if there exist strings $X_1, \ldots, X_k$ that cause C to output 1, then there exists a sequence of strings S so that $\mathrm{Prob}_R[C_R(S, E(X_1), \ldots, E(X_k)) = 1] = 1$, and

iii. if there exist $Y_1, \ldots, Y_k$ and a sequence of strings S such that $\mathrm{Prob}_R[C_R(S, Y_1, \ldots, Y_k) = 1] > \frac{2}{3}$, then $E^{-1}(Y_1), \ldots, E^{-1}(Y_k)$ is an input that causes C to output 1.

Proof: [Sketch] Let $X_1, \ldots, X_k$ be strings that cause C to output 1. Consider a circuit C' which contains both the circuit C as well as circuits that produce the Justesen encodings of $X_1, \ldots, X_k$. The strings S should contain a proof of the form constructed in Section 8 that C' accepts. After checking this proof, the verifier will choose a few bits at random of each input string Y_i and verify that these bits agree with the corresponding bits produced by the Justesen encoding circuits contained in C'. To show that this can be done, we need to prove that the verifier can check the outputs of individual gates of the circuit represented in S.

Let $\hat{P}$ be the presentation of P and let (x_0, y_0, z_0) be the node of the de Bruijn router whose output value we would like to examine. It does not suffice to examine the value at (x_0, y_0, z_0) in the first table of $\hat{P}$ because there is no guarantee that the value contained there is actually $P(x_0, y_0, z_0)$. To obtain a value that we can be confident is correct, we use an idea from [BF93]: The verifier should choose a few points of the form (x, y, z_0) uniformly at random and verify that the three polynomials provided in the second table of the presentation passing through that point have the value assigned to that point by the first table. Passage of this test certifies the sub-representation on $\mathcal{I} \times \mathcal{I} \times z_0$. The verifier should then perform the same test on a few points of the form (x, y_0, z_0). Passage of this test certifies that the polynomial representing the restriction of P to $\mathcal{I} \times y_0 \times z_0$ actually is the restriction of P to that line, so the verifier can confidently use the value of that polynomial at (x_0, y_0, z_0).

One can verify that these tests can be performed by circuits of the required sizes. □

Theorem 15. *For all $q(n) = O(\log \log n)$, there exists a polynomial time Turing machine V_2 that, when given a random string R and a circuit C of size m as input, will output the description of a circuit C_R of size $2^{O(q(n))}$ such that*

i. C_R expects an input Π of $n^{1+2^{-O(q(n))}}(\log n)^{O(q(n))}$ bits, but C_R reads only $2^{O(q(n))}$ bits of Π,

ii. if there exists an input that causes C to output 1, then there exists an input Π so that $\mathrm{Prob}_R[C_R(\Pi) = 1] = 1$, and

iii. if there exists an input Π such that $\mathrm{Prob}_R[C_R(\Pi) = 1] > \frac{2}{3}$, then there exists an input that causes C to output 1.

Proof: [Sketch] V_2 first runs V of Lemma 13. To the circuit that V produces, it applies V_1 of Lemma 14. It should repeat the tests indicated by V_1 a few times to decrease the error probability to below 1/3. V_2 should then apply V_1, and decrease error probability, $q(n) - 1$ more times.

At each level of the recursion, the proof size increases by a factor of at most $\log^{O(1)} n$, the number of queries increases by a constant factor, and the size of the strings queried undergoes the transformation $m \mapsto \sqrt{m} \log m$. After $q(n)$ iterations, we have proofs of size $n(\log n)^{O(q(n))}$ to which we make $2^{O(q(n))}$ queries, each of length at most $n^{-2^{O(q(n))}}$. We now apply the main theorem of [ALM+92] to cap the recursion. This will enable the verifier to query a constant number of bits for each string, at the expense of a polynomial expansion of the size of each string. □

Corollary 16. *$NP = PCP(\log n, 1)$, with proofs of size $n^{1+\epsilon}$, for any $\epsilon > 0$.*

Proof: Let $q(n)$ be a constant in the above theorem. It is not difficult to verify that only $O(\log n)$ random bits are used.

10. Discussion

In [BFLS91], the authors describe how *any* proof in any reasonable formal system can be described as a coloring problem on a graph. Their framework has the advantage that their theorem candidate is encoded so that the verifier need only read a few bits of the theorem candidate, and the theorem candidate can be much smaller than its proof. It is not difficult to incorporate our techniques into their framework. They need a graph that can sort. Such a graph can be obtained by taking $O(r^2)$ levels of de Bruijn graphs. In fact, one can represent any finite cartesian product of line-graphs and de Bruijn graphs in such a way that a node's neighbors can be obtained by applying a linear polynomial.

We also want to note that it is possible to bootstrap a bivariate theorem such as Theorem 9 to obtain an m-variate version in which we replace the condition $n > 2\delta n + 2d$ with the condition that $n > m^{\Omega(1)}(\delta n + d)$.

11. Acknowledgements

We would like to thank Mike Sipser and Alexander Shen for putting us in touch with each other. The first author would like to thank Alexander Shen for bringing this problem to his attention and encouraging his pursuit of it. The second author would like to thank Mike Sipser, Madhu Sudan, Nick Reingold and Gian-Carlo Rota for many enlightening conversations, as well as Marcos Kiwi for proofreading and proof-checking.

References

[ALM+92] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. In *Proc. of the 33rd IEEE FOCS*, pages 14–23, 1992.

[AS92] S. Arora and S. Safra. Probabilistic checking of proofs. In *Proc. of the 33rd IEEE FOCS*, pages 2–13, 1992.

[BF90] D. Beaver and J. Feigenbaum. Hiding instances in multioracle queries. In *STACS90*, volume 415, pages 171–176. Springer LNCS, 1990.

[BF93] L. Babai and K. Friedl. On slightly superlinear transparent proofs. CS 93-13, The University of Chicago, Department of Computer Science, Chicago, IL, 1993.

[BFLS91] L. Babai, L. Fortnow, L. Levin, and M. Szegedy. Checking computations in polylogarithmic time. In *Proc. of the 23rd ACM STOC*, pages 21–31, 1991.

[BLR90] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. In *Proc. of the 22nd ACM STOC*, pages 73–83, 1990.

[FGL+91] U. Feige, S. Goldwasser, L. Lovàsz, S. Safra, and M. Szegedy. Approximating clique is almost np-complete. In *Proc. of the 32nd IEEE FOCS*, pages 2–12, 1991.

[GLR+91] P. Gemmell, R. Lipton, R. Rubinfeld, M. Sudan, and A. Wigderson. Self-testing/correcting for polynomials and for approximate functions. In *Proc. of the 23rd ACM STOC*, pages 32–42, 1991.

[GS92] P. Gemmell and M. Sudan. Highly resilient correctors for polynomials. *IPL*, 43:169–174, 1992.

[Lan93] S. Lang. *Algebra.* Addison-Wesley, 3rd edition, 1993. (material only in 3rd edition).

[Lei92] F. T. Leighton. *Introduction to Parallel Algorithms and Architectures.* Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1992.

[Lip91] R. J. Lipton. New directions in testing. In J. Feigenbaum and M. Merritt, editors, *Distributed Computing and Cryptography*, volume 2 of *Dimacs Series in Disc. Math. and Theor. Comp. Sci.*, pages 191–202. A.M.S., 1991.

[MS77] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes.* North Holland, Amsterdam, 1977.

[RS92] R. Rubinfeld and M. Sudan. Testing polynomial functions efficiently and over rational domains. In *Proc. of the 3rd ACM-SIAM Symp. on Discrete Algorithms*, pages 23–32, 1992.

[Rub90] R. Rubinfeld. *A Mathematical Theory of Self-Checking, Self-Testing and Self-Correcting Programs.* PhD thesis, University of California, Berkeley, CA, 1990.

[Sud92] M. Sudan. *Efficient checking of polynomials and proofs and the hardness of approximation problems.* PhD thesis, U.C. Berkeley, Oct. 1992.

[vdW53] B. L. van der Waerden. *Modern Algebra*, volume 1. Frederick Ungar Publishing Co., New York, 1953.