Efficient Self-Testing of Linear Recurrences

S. Ravikumar[1] D. Sivakumar[2]

[1]Department of Computer Science, Cornell University, Ithaca, NY 14853. Email: ravi@cs.cornell.edu. This work was done when the author was visiting M.I.T. Research supported in part by ONR Young Investigator Award N00014-93-1-0590.

[2]Department of Computer Science, State Univ. of New York at Buffalo, Buffalo, NY 14260. Email: sivak-d@cs.buffalo.edu. Research supported in part by NSF grant CCR-9409104.

Abstract

We consider the problem of designing efficient self-testers for linear recurrences, and present a complete package of self-testers for this class of functions. The results are proved by demonstrating an efficient reduction from this problem to the problem of testing linear functions over certain matrix groups. Our tools include spectral analysis of matrices over finite fields, and various counting arguments that extend known techniques. The matrix twist yields completely new self-testers for polynomials over the finite field $\mathbb{Z}/(p)$. The efficiency of our polynomial self-testers is better than all previously known testers, and in the univariate case, we are able to match the efficiency of the Blum–Luby–Rubinfeld linearity tester. We also present self-testers for convolution identities over groups, and improved self-testers for polynomials over rational domains.

1 Introduction

The concept of program self-testing, introduced by Blum, Luby, and Rubinfeld [BLR93], is a radically different approach to testing the correctness of programs. In conventional program testing, the program is tested by comparing its output, say on many randomly chosen inputs, to the correct answer it is supposed to compute. The intellectual leap in program self-testing is to exploit properties of the function being computed, and to test the program's correctness by using the (possibly faulty) program itself to crosscheck its work.

Informally, a *self-tester* T for a function f is a probabilistic program that works as follows. Given a program P that purportedly computes f, T uses P as a black-box, and either rejects P as "too erroneous," or certifies P as "pretty trustworthy." The minimal requirements for T are that it should not require the knowledge of the correct value of f on too many inputs (as a conventional program tester does), and that it should be simpler than and different from a program to compute f (that is, it should not perform its tests by computing the correct values for itself).

The broad agenda for the research program on self-testing is to investigate and identify the mathematical functions that allow efficient self-testing. Research in this area has progressed in two directions. First, influenced by developments in interactive proof systems, efficient testers were built for linear functions and for polynomials. These ideas were used extensively in obtaining the breakthrough results on probabilistically checkable proofs [BFLS91, FGL$^+$91, AS92, ALM$^+$92]. Another main thrust of this research program has been to build realistic program self-testers that can be used, for example, in testing the correctness of VLSI chips designed to implement special functions, numerical algorithms, etc. Some of the main developments along this line of research includes the work in [Rub90, CL90, RS92, ABCG93, Rub94, Erg95, RS95]. These two lines of research have vastly benefited from each other, particularly exemplified by the combination of ideas from [AS92] and [RS92], culminating in the results of [ALM$^+$92].

In this paper, we continue the quest for identifying a larger class of self-testable functions. It has long been realized that the basis of many known self-testers is the self-reducibility of various functions—the most prominent examples include linear functions, polynomials, and functions described by "functional equations." Our work focuses on studying the self-testability of functions described by *linear recurrence relations*. Linear recurrences generalize polynomials, and are also the basis for many combinatorially interesting sequences (such as the Fibonacci sequence, Lucas sequence, etc.). Moreover, since linear recurrences have a very rich self-reducible structure, they form an ideal candidate to study in order to understand better the connection between self-reducibility and self-testability. Yet another reason to study linear recurrences is that it is a natural first step towards studying differential equations, which we believe to be of practical importance, and which would link the area of self-testing to many rich concepts in mathematical analysis.

Results. Our main result in this paper is a complete set of self-testers for linear recurrences over various rings and fields. We identify, analyze, and solve the various cases of linear recurrences, each of which requires new and different ideas. First we show that if the recurrence yields a "convolution identity" over groups, then we can extend the technique of [BLR93] to design a self-tester. To handle the other cases, we present a very efficient reduction from the problem of self-testing recurrences to the problem of self-testing linearity over a multiplicative group of matrices. Within the matrix framework, we present a systematic classification of the various cases, and proceed to solve them. A particularly interesting application is obtained by viewing polynomials as linear recurrences: we obtain totally new self-testers for univariate and multivariate polynomials, of much better efficiency than known before. Finally, we extend our work to the case of rational domains—sets of fixed point rational numbers of pre-defined precision.

To aid the exposition in the remainder of this Introduction, we present the formal definition of self-testers at this point.

Definition 1 (Blum, Luby, and Rubinfeld [BLR93]) *Let f be a function on a domain D. An* ***ϵ-self-tester*** *for f is a probabilistic program T that, given a program P for f, and a confidence parameter β, satisfies the following conditions:*

(1) $\Pr_{X \in D}[P(X) = f(X)] = 1 \Rightarrow \Pr[T^P$ *outputs "PASS"*$] \geq 1 - \beta$.

(2) $\Pr_{X \in D}[P(X) \neq f(X)] > \epsilon \Rightarrow \Pr[T^P$ *outputs "FAIL"*$] \geq 1 - \beta$.

As we observe in Section 2, the higher the value of $\delta = \epsilon/2$, fewer the queries the self-tester makes to the program P, so one of the goals in building self-testers is to make δ as large as possible.

Highlights of our approach. The main technical contribution of this work is the novel viewpoint using matrices. As we show, this approach has many pay-offs, some of which we summarize briefly.

(1) The matrix framework gives the first reduction from testing univariate polynomials to testing linear functions. From the point of view of self-testing, linear functions are fairly well-understood, and have extremely strong self-testers [BLR93] that can tolerate δ up to $2/9$. Since our reduction preserves efficiency, we obtain the first self-testers for univariate polynomials whose quality matches that of the linearity tester. Our proof is also much simpler than any currently known proof. Combining this with the work of Friedl and Sudan [FS95], we obtain self-testers for multivariate polynomials over the finite field $\mathbb{Z}/(p)$, with better parameters than known before: $\delta < 1/8$, and $O(d)$ probes for a polynomial of total degree d.

(2) The matrix approach lets us perform "bulk" testing of convolution identities (i.e., testing several instances of the identity together), and provides stronger self-testers for convolutions over groups than can be obtained through traditional ideas.

(3) We characterize those recurrences that can be self-tested with parameters matching that of the linearity tester. We prove a theorem that shows that among all linear recurrences, polynomials over $\mathbb{Z}/(p)$ enjoy a very special and unique mathematical structure, explaining somewhat the extremely efficient self-testers that have been built for polynomials. Our matrix viewpoint allows us to apply some of the fundamental results from linear algebra to the problem of self-testing. In particular, the spectral analysis of certain matrices that describe recurrence relations proves to be very useful. This shows a link, for example, between testing Fibonacci sequences and the notion of Jordan normal forms—two seemingly unrelated concepts.

(4) The convolution identities are a very general method to obtain robust characterizations from non-robust descriptions of recurrence relations. We adapt the BLR proof to prove the robustness of general convolution equations over groups, and extend it for certain special cases. The extension is non-trivial, and is combinatorially involved.

(5) The application to rational domains has an interesting consequence. Previously, it was known how to self-test univariate polynomials over rational domains [RS92], but this required that the program being tested work correctly on a rational domain of finer precision. Our work removes that requirement, transferring the overhead to the self-tester. Also, while the best known value for δ was $O(1/d)$, we achieve $\delta = \Omega(1)$, in fact, $1/24$.

(6) Finally, the matrix approach and ideas from spectral analysis promise hope for developments on the long-standing open problem of self-testing real-valued functions that are computed

approximately.

Notation. In this paper, $\mathbb{Z}$ denotes the set of integers, and for $N > 0$, D_N denotes the set $\{n \mid -N < n < N\}$, and D_N^+ denotes the set $\{n \mid 0 \leq n < N\}$.

Outline of the paper. The next section describes the idea of "robustness" of properties, which plays a key role in building self-testers. Section 3 links the testing of linear recurrences to the robustness of certain convolution identities. In Section 4, we first motivate the need for the matrix approach, and introduce the basic framework. Section 5 lists the various cases that need to be handled, and shows how to build self-testers via the matrix approach, modulo technicalities that differ from case to case. Section 6 sketches efficient schemes for each of the individual cases.

2 The Key Idea: Robust Properties

The conventional approach to building a self-tester would be the following: T picks many random inputs X, invokes P to compute $P(X)$, computes the correct values of $f(X)$ using some other trustworthy mechanism, and estimates the fraction of inputs on which P is good. As mentioned before, this is undesirable since the self-tester is just another program that computes the same function, which implies that we can't expect it to be faster than the program P itself. The key twist introduced by Blum, Luby, and Rubinfeld is to focus on some other property of the function f and exploit the following notion of robustness:

Definition 2 (Robust Property) *A property $\mathcal{I}$ is (ϵ, δ)-**robust** for a class of functions $\mathcal{F}$ over a domain $\mathcal{D}$, if*

(1) *Every $f \in \mathcal{F}$ satisfies $\mathcal{I}$ over the entire domain $\mathcal{D}$.*

(2) *If a function (program) P satisfies $\mathcal{I}$ on all but a δ fraction of the domain $\mathcal{D}$, then there exists a function $g \in \mathcal{F}$ that agrees with P on all but a ϵ fraction of the inputs in $\mathcal{D}$.*

For example, the property "$f(x + y) = f(x) + f(y)$" is robust (with appropriate parameters) for the class of all linear functions over groups. We call a robust property *useful* if it can be easily checked, given the required values (such as $f(x), f(y), f(x + y)$).

Let us see how a useful (ϵ, δ)-robust property $\mathcal{I}$ helps in building an ϵ-self-tester. Suppose a program P claims to compute a function f in the class $\mathcal{F}$, and say $\mathcal{I}$ is (ϵ, δ)-robust for $\mathcal{F}$. The self-tester may now estimate the fraction of the domain over which the property $\mathcal{I}$ is satisfied by the program P. Since $\mathcal{I}$ is useful, this is an easily accomplished task. If P satisfies $\mathcal{I}$ for all but a δ fraction of the domain, then robustness guarantees that there is a function $g \in \mathcal{F}$ which agrees with P on all but an ϵ fraction of the inputs. Moreover, if every function in $\mathcal{F}$ possesses some "random-self-reducibility," then using the program P that computes g correctly on most inputs, one can use P and compute g correctly on every input, with high probability. This process is called "self-correction" [BLR93, Lip91]. Linear functions and polynomials are random-self-reducible; in the next section, we point out the random-self-reducibility of convolution identities derived from recurrences. For the subsequent sections, this is irrelevant, since we reduce the testing of recurrences to testing linearity, which is clearly random-self-reducible.

Having ensured that P computes *some* function $g \in \mathcal{F}$, the self-tester then performs a few additional tests to ensure that $g \equiv f$, where f is the function that P was designed to compute. Typically, this requires testing if f and g agree on a certain minimal number of inputs that are enough to distinguish any two functions in the class $\mathcal{F}$. For example, if two linear functions of the form $L(n) = \omega n$ agree on one point, they must agree everywhere; if two degree d polynomials agree on $d+1$ points, they must agree everywhere. To compute g, the tester uses the faulty program P and the self-correcting mechanism.

The main parameter of interest in designing a self-tester is the number of queries made to the program P being tested. If the self-tester is based on an (ϵ, δ)-robust property, what the self-tester really does is estimate the probability δ that P fails to satisfy the robust property. To obtain a reasonably accurate estimate of this, the tester must make $\Omega(1/\delta^2)$ calls to the program P being tested. Therefore, it is desirable to make δ as large as possible. In all our self-testers, δ will be $\epsilon/2$.

3 The Robustness of Convolutions

The main purpose of this section is to develop the basic ideas that will be improved upon later, and to prove the robustness of certain convolution identities that one may derive from linear recurrences. To facilitate exposition, let us begin by considering two integer sequences that are interesting and simple. Our examples are the second order linear recurrence relations that describe the familiar Fibonacci sequence and arbitrary (integral) linear functions. The Fibonacci sequence is given by

$$F(n) = F(n-1) + F(n-2) \text{ for } n > 1, \qquad F(0) = 0, \qquad F(1) = 1,$$

and the linear function $L(n) = \omega n$ is described by the recurrence

$$L(n) = 2L(n-1) - L(n-2) \text{ for } n > 1, \qquad L(0) = 0, \qquad L(1) = \omega.$$

Of course, by the fundamental result of Blum, Luby, and Rubinfeld, it is well-known that linear functions can be efficiently self-tested. Nevertheless, we consider it for the sake of exposition, and to point out some differences and complications that arise for arbitrary recurrences. While robust equations naturally lead to self-testers, it is not hard to see that the recurrence relations, as written above, are not robust characterizations of the sequences described. For an illustration, see the Appendix.

Past experience with linear functions has indicated that while the identity $\forall n, f(n) = f(n-1) + f(1)$ is not a robust characterization of linear functions, the derived identity $\forall n, k, f(n) = f(n-k)+f(k)$ is indeed robust. Our first idea is to take the similar step. The following *convolution identities* may be easily derived by induction:

(1) For all $m, n > 0, F(n) = F(m+1)F(n-m) + F(m)F(n-m-1)$.

(2) For all $m, n > 0, \omega L(n) = L(m+1)L(n-m) - L(m)L(n-m-1)$.

Of course, technically, we have to extend the definitions of F and L to negative integers, but that does not pose a problem. For $n < 0$, we let $F(n) = F(n+2) - F(n+1)$ and $L(n) = 2L(n+1) - L(n+2)$. For the Fibonacci sequence, this implies that $F(-n) = (-1)^{n+1}F(n)$, and for the linear function, we have $L(-n) = -L(n)$. It is clear that these definitions preserve the recurrence relations, and satisfy the above convolution identities.

At this point, we will formally state a very simple version of our first goal. To make the manuscript readable, we use the notation F_n to denote $F(n)$. *Let F be a function from a group H into a ring K that satisfies the convolution identity*

$$\mathcal{I} \equiv \text{“}(\forall n, m \in H)\,[\omega F_n = \alpha F_{m+1}F_{n-m} + \beta F_m F_{n-m-1}]\text{”}$$

where $\alpha, \beta, \omega \in K$. Show that $\mathcal{I}$ is a robust property. This goal is simpler than our original goal, because we assume the input domain to be a group. Later we will point out the complications that arise if this is relaxed, and show how to solve them.

In fact, solving the above problem would only establish the robustness of the convolution identity. It does yield self-testers over the group in which the convolution holds. For example, it gives, in principle, a self-tester for Fibonacci sequences over the integers, since the convolution identity for the Fibonacci sequence is valid over the group of integers. However, any realistic computer program computing the Fibonacci sequence has limitations on the size of the domain it can handle. Suppose P is a program that can compute the n-th Fibonacci number F_n for all $n \in D_N$. This domain is not a group under integer addition but only under addition modulo $\pm N$. However, it is not hard to see that when restricted to inputs in this domain and addition modulo $\pm N$, the convolution identity is no longer valid (eg., when $n < 0, m > 0, n - m < -N$).

On the other hand, the convolution identity for $L(n)$ remains valid even if the additions are performed mod $\pm N$ (see Appendix for a proof). This is an important difference between the Fibonacci sequence and the linear function $L(n)$. Later, in Section 6.1, we will precisely characterize all recurrences whose convolution identities remain valid if the computations are performed in $\mathbb{Z}/(p)$ for prime p. It turns out that the only second order recurrences whose convolution identities remain valid in $\mathbb{Z}/(p)$ are the linear functions. We now proceed to prove the robustness of convolution identities satisfied by functions from a group H into a ring K. In the following theorem, the domain of the arguments is understood to be the group H.

Theorem 1 *If a program P satisfies "$\Pr_{n,m}[\omega P_n \neq \alpha P_{m+1} P_{n-m} + \beta P_m P_{n-m-1}] \leq \delta < 1/26$," then there is a function G such that G satisfies $\omega G_n = \alpha G_{m+1} G_{n-m} + \beta G_m G_{n-m-1}$ for all n, m, and such that $\Pr_n[P_n \neq G_n] \leq 2\delta$.*

The proof is similar in flavor to the proof of the Blum–Luby–Rubinfeld theorem attributed to Coppersmith (see [BLR93, RS93]), but requires an additional trick. The proof is presented in the Appendix, where we also show how to build self-testers for convolutions over groups, and obtain the following result.

Theorem 2 *Functions $F : H \to K$, where H is a group and K is a ring, that satisfy second order convolutions of the form $\omega F(n) = \alpha F(m+1)F(n-m) + \beta F(m)F(n-m-1)$ have ϵ-self-testers for $\epsilon/2 = \delta < 1/26$.*

4 The Matrix Framework

In this section, we set up some basic machinery that will let us get around the problems mentioned in the previous section. Let us first summarize the status of our goal. We set out with the task of building self-testers for linear recurrences of order $d > 1$. For $d = 2$, we showed that if the recurrence is a function from a group into a ring, and if the group can be efficiently sampled from, the convolution identity that one may obtain from the recurrence helps us build a self-tester. However, we are still faced with many problems:

(1) For $d > 2$, while the above approach works, the proof becomes very messy and the bound on δ is only $O(1/d^2)$.

(2) If a program purportedly computes a recurrence such as Fibonacci for all $n \in D_N$ (some finite segment of the integers), the convolution identity is of no help in testing such a program.

This is because the proof of the robustness of convolution relies on the fact that if n and k are uniformly distributed elements of a group H, then so is $n+k$. This is not true of the domain D_N. If we view the domain D_N as a group under addition mod N, then the convolution identity is no longer true.

(3) Not all integer recurrences have natural, meaningful, or even reasonable extensions to negative numbers. For example, the sequence defined by $F(n) = F(n-1) + 2F(n-2)$, $F(0) = 0$, $F(1) = 1$, has no natural (integral) extension to negative integers such that the derived convolution identity is preserved.

To handle the above problems, we introduce a matrix framework under which we study linear recurrences. As we will see, the matrix framework not only offers a clean and natural viewpoint for the above problems, but also offers elegant solutions to them.

Let f be a linear recurrence of order d from an abelian group H into a ring K. For the purposes of this paper, H will always be taken to be a ring of the form $\mathbb{Z}/(N)$ or the field $\mathbb{Z}/(p)$ for some prime p. Let $a_0, a_1, \ldots, a_{d-1}$ be d constants in K, not all zero, and assume that f is specified by the recurrence

$$f_n = \begin{cases} a_n & n < d, \\ c_1 f_{n-1} + c_2 f_{n-2} + \ldots + c_d f_{n-d} & n \geq d, \end{cases}$$

where $c_d \neq 0$. Wlog. we may also assume that f is a *genuine* recurrence of order d—that is, it cannot be written as a recurrence of smaller order. Simple calculations show that the following matrix recurrence holds:

$$\begin{pmatrix} f_n \\ f_{n-1} \\ f_{n-2} \\ \vdots \\ f_{n-d+1} \end{pmatrix} = \begin{pmatrix} c_1 & c_2 & \ldots & c_{d-1} & c_d \\ 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \end{pmatrix} \begin{pmatrix} f_{n-1} \\ f_{n-2} \\ f_{n-3} \\ \vdots \\ f_{n-d} \end{pmatrix}. \tag{1}$$

Denote by M the $d \times d$ matrix in the above expression; M has a special form—the bottom-left $(d-1) \times (d-1)$ submatrix is just the identity matrix I_{d-1}. Since $c_d \neq 0$, it is clear that the rows (or the columns) of M are linearly independent over K. From Equation (1), it is also immediate that

$$(f_n, f_{n-1}, \ldots, f_{n-d+1})^T = M^{n-d+1}(a_{d-1}, a_{d-2}, \ldots, a_0)^T.$$

For any k, we use the notation $\vec{f_k}$ as a shorthand for the vector $(f_k, f_{k-1}, \ldots, f_{k-d+1})^T$. Note that $\vec{f}_{d-1} = (a_{d-1}, a_{d-2}, \ldots, a_0)^T$. The first lemma we prove shows that the vectors obtained from the first $2d-1$ values of f are linearly independent. The proof of the lemma is given in the Appendix.

Lemma 3 *If f is a genuine linear recurrence of order d, the vectors $\vec{f}_{d-1}, \vec{f}_d, \vec{f}_{d+1}, \ldots, \vec{f}_{2d-2}$ are all linearly independent over K (and hence over the quotient field of K).*

Before we proceed to describe the solutions built using this approach, let us review, within the matrix framework, the problems mentioned at the beginning of this section. Let F denote the linear recurrence of order d, let $M = M_F$ denote the matrix obtained by writing the recurrence in matrix form, and let A denote the matrix $\left(M^{d-1}\vec{f}_{d-1}, M^{d-2}\vec{f}_{d-1}, \ldots, M\vec{f}_{d-1}, \vec{f}_{d-1}\right)$. By Lemma 3, the columns of A are linearly independent over K. We also know that the columns of M are linearly independent over K. For the Fibonacci sequence F and the linear function $L(n) = \omega n$,

$$M_F = A_F = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, \qquad M_L = \begin{pmatrix} 2 & -1 \\ 1 & 0 \end{pmatrix}, \qquad \text{and} \qquad A_L = \begin{pmatrix} 2\omega & \omega \\ \omega & 0 \end{pmatrix}.$$

The simplest case is if both M and A are, in fact, invertible over K. Since $\det M = c_d$, M is invertible over K if and only if c_d is invertible over K. If K is the ring of integers, M is invertible if and only if $c_d = \pm 1$. Clearly, M_F, A_F, and M_L are invertible over the integers, but A_L is not. A moment's reflection points out that the invertibility of M is the crucial reason why these functions could be defined naturally for negative integers. The fact that the recurrence $f_n = f_{n-1} + 2f_{n-2}$ does not extend as an integer sequence over negative integers is precisely because the matrix M is not invertible over the integers. As we will show soon, the invertibility of A is not so crucial, but let us pretend for the moment that A is invertible over K. Since for all $n \geq 2d-2$, $\vec{f}_n = M^{n-d+1}\vec{f}_{d-1} = M^{n-2d+2}M^{d-1}\vec{f}_{d-1} = M^{n-2d+2}\vec{f}_{2d-2}$, we have:

$$\begin{aligned} \widehat{f}_n &=_{\text{def}} \left(\vec{f}_n \ \vec{f}_{n-1} \ \cdots \ \vec{f}_{n-d+2} \ \vec{f}_{n-d+1}\right) \\ &= M^{n-2d+2} \times \left(\vec{f}_{2d-2} \ \vec{f}_{2d-3} \ \cdots \ \vec{f}_d \ \vec{f}_{d-1}\right) = M^{n-2d+2} \times A. \end{aligned}$$

If A is invertible over K, for every $n \geq 0$, we may recover $M^n = \widehat{f}_{n+2d-2} \times A^{-1}$. This is precisely the idea behind the self-tester that we build.

Given a program P that purports to compute f, the strategy for testing P is to build a program Q that takes n as an argument, and computes M^n in this fashion by using P as an oracle to compute $\widehat{f}_{n+2d-2}$. The program Q will have A^{-1} "hardwired" into it—this is reasonable, since to self-test f, we anyway need to know the values of f on at least d points. The knowledge of A^{-1} is tantamount to knowing only $2d-1$ values of f; moreover A^{-1} can be pre-computed by a reliable program for matrix inversion. The function $\varphi(n) = M^n$ is now a homomorphism from $\mathbb{Z}$ into the ring of matrices over K, and our plan is to self-test Q, which computes φ.

What our reduction does is effectively make the linearity that is latent in the recurrence take an explicit form as the homomorphism φ, which is more amenable to efficient testing, using the linearity tester of Blum, Luby, and Rubinfeld [BLR93]. The conceptual and technical advantage achieved stems from the fact that the Blum–Luby–Rubinfeld self-tester for linear functions (homomorphisms) is very efficient. Our matrix approach can be viewed another way. Testing the validity of $M^n = M^k \times M^{n-k}$ is the same as testing the validity of $\widehat{f}_{n'} = \widehat{f}_{k'}A^{-1}\widehat{f}_{(n-k)'}$, and each entry of $\widehat{f}_{n'}$ is a convolution of d entries each from $\widehat{f}_{k'}$ and $\widehat{f}_{(n-k)'}$. Thus, testing the linearity of the matrix powers is really a simultaneous "bulk" testing of the convolution identity at $2d-1$ adjacent points. This turns out to have an added benefit of being able to improve the parameters of the self-tester. For example, while straightforward testing of recurrences using the convolution identity requires $\delta = O(1/d^2)$, performing the linearity testing tolerates $\delta = \Omega(1)$.

The application of the BLR-style self-tester is not straightforward, and a number of additional technical issues still remain. In the next sections, we show how to handle the various cases that arise.

5 Building Self-Testers: Overview

As observed in the previous section, the function $\varphi(n) = M^n$ is a homomorphism from $\mathbb{Z}$ into the ring K. Realistically, however, programs are built to compute functions or recurrences not for all integers, but for a finite subset D_N^+ or D_N. It is this situation that we wish to address, and build self-testers for. We begin by outlining the different cases that we solve.

(1) (Domain $= D_N$ or D_N^+, Range $= \mathbb{Z}/(p)$) When the range is the finite field $\mathbb{Z}/(p)$, we consider the order of the matrix M in the abelian group $\mathcal{M}$ of matrices over $\mathbb{Z}/(p)$. If the order of the matrix is N, then φ is a homomorphism from the group $\mathbb{Z}/(N)$ into the matrix group $\mathcal{M}$.

Our central idea in this case is to focus on the spectrum of the matrix M to determine its order, and identify those recurrences whose matrices have a small order. The main payoff is that we can use the BLR linearity tester "as is," without having to resort to more complicated versions of it. Consequently, we achieve the same strength in the parameter δ as the BLR test. For example, while we only achieved a bound of $\delta < 1/26$ for the case of second order convolutions over groups (this can be improved slightly), appealing to the BLR tester gives a bound of $\delta < 2/9$ for the same problem.

(2) (Domain $= D_N^+$ or D_N, Range $=$ a ring K) If the matrix A is invertible over K, one can compute M^n, given $\widehat{f}_{n+2d-2}$. Even if A is not invertible over K, the identity $\widehat{f}_{n+2d-2} = M^n \times A$ is valid. For any matrix A over a ring K, the identity "$AA^\perp = \det A$" is valid over the ring, where $A^\perp$ denotes the adjoint of A (the matrix of cofactors of A). Therefore, the identity $\widehat{f}_{n+2d-2}A^\perp = (\det A)M^n$ is valid. Since M^n is a matrix over the ring K, even if $\det A$ is not invertible in K, the product $\widehat{f}_{n+2d-2}A^\perp$ *must* be divisible by $\det A$. To obtain M^n, therefore, the program Q invokes the program P being tested to compute $\widehat{f}_{n+2d-2}$ and multiplies it by $A^\perp$. If this product is not divisible by $\det A$, the test $M^n = M^k M^{n-k}$ is counted as a failure. Clearly, if P is a perfect program, Q will have no trouble doing the division. Conversely, if Q is certified on $1-\epsilon$ fraction of the inputs, then it has passed a more stringent test than before, and the conclusion about P remains valid.

(3) (Domain $= D_N$, Range $= K$ (ring or field)) Since D_N has negative values, it is implicit that M is invertible over K, so that the recurrence has a meaningful extension to negative integers. The main advantage of M being invertible, or equivalently, $\varphi(n)$ being defined on positive and negative values of n, is that we may appeal to a known adaptation of the BLR test, due to Rubinfeld [Rub94], building on ideas from [GLR+91] and [RS92]. In this case, by careful choice of parameters, Rubinfeld's self-tester works for δ up to $1/24$, and guarantees that the program works correctly for a large fraction of D_N, roughly the set $\{n \mid -N/10 < n < N/10\}$. Equivalently, to ensure that the program works correctly on D_N, the self-tester requires that it work correctly for a domain roughly ten times the size of D_N. We omit further details.

(4) (Domain $= D_N^+$, Range $= K$ (ring or field)) In many cases, as we pointed out, it is neither natural nor reasonable to extend the recurrence to negative values. Suppose a program claims to compute such a recurrence (into a ring K) for all values in D_N^+. We show how to self-test it and certify that it works correctly for the much larger subset (than in the previous case) $\{n \mid 0 \le n < N/4\}$. The proof of this self-tester is more complicated, and achieves $\delta < 1/3456 = \Omega(1)$.

(5) (Domain, Range are rational domains) Finally, we make some observations that show how the ideas of the previous cases can be extended to the case where the recurrence is really a polynomial over the reals. Under the assumption that the program claims to compute the recurrence correctly over a set of "fixed point" (rational) numbers, we show how to self-test the program efficiently. This improves, in a technical sense, the results of Rubinfeld and Sudan [RS92]—while their work requires the program to work correctly over a rational domain of finer precision, ours does not.

The above cases differ in how the program Q that computes the function φ is tested, and that is what we will discuss in the subsequent sections. Once Q has been tested, we still have to "transfer" the conclusions about Q to conclusions about P. Below we show how this is done. (The domain D is any of D_N, D_N^+, $\mathbb{Z}/(N)$, or $\mathbb{Z}/(p)$.) The proof is deferred to the Appendix.

Theorem 4 *If P computes f perfectly, then Q computes φ perfectly, and the tester always outputs "PASS." Conversely, suppose the self-tester for Q outputs "PASS" and certifies that Q computes $\varphi(n) = M^n$ correctly for $1-\epsilon$ fraction of the inputs in the domain D. Then P computes the recurrence f correctly on (roughly) $1-\epsilon$ fraction of the inputs in the domain D.*

6 Details of the Self-Tester

6.1 Case (1): Recurrences over Finite Fields & Application to Polynomials

Let $f : \mathbb{Z} \to K$ be a recurrence of order d over the field K, and let M and A be the associated matrices in the multiplicative group $\mathcal{M}$ of $d \times d$ matrices over K. Recall that the *order* of M (denoted $ord(M)$) is the least integer $n \geq 0$ such that $M^n = I$, the identity matrix. As noted in the previous section, if N is the order of the matrix M, the function $\varphi : \mathbb{Z}/(N) \to \mathcal{M}$, defined by $\varphi(n) = M^n$, is in fact a homomorphism that can be self-tested using the BLR self-tester. In this section, we focus on the spectrum of the matrix M and investigate under what conditions M has small order. Due to space limitations, the discussion is given in the Appendix. The upshot of the spectral analysis is the following:

Theorem 5 *If M has all eigenvalues in K, then $ord(M) \mid p(p-1)$. If M is diagonalizable over K, then $ord(M) \mid p-1$.*

We give a special application for polynomials over the finite field $K = \mathbb{Z}/(p)$. Using the Lagrange interpolation formula, we can express degree d polynomials as linear recurrences of order $d+1$ as follows: $f_n = \sum_{i=1}^{d+1} \alpha_i f_{n-i}$, where the α_i's are fixed constants in K, given by $\alpha_i = (-1)^{i+1}\binom{d+1}{i}$. The matrix M that corresponds to a degree d polynomial, therefore, consists of the α_i's in the first row, and a copy of $[I_d \; 0]$ below it. Based on an analysis of properties of this matrix, we obtain:

Theorem 6 *The matrices M that correspond to polynomials over $\mathbb{Z}/(p)$ have order exactly p. Moreover, these are the only recurrences whose matrices have order exactly p.*

Therefore, the function $\varphi : \mathbb{Z}/(p) \to \mathcal{M}$, where $\mathcal{M}$ is the multiplicative group of matrices over $\mathbb{Z}/(p)$, defined by $\varphi(n) = M^n$ is a group homomorphism, since $M^p = I$. This gives us a new self-tester for univariate polynomials over the finite field $\mathbb{Z}/(p)$ that achieves the bound $\delta < 2/9$, improving all previously known results for this case [RS92, RS93, FS95]. We note that effectively the test conducted by the self-tester is extremely simple: all it does is to check the convolution identity at multiple points simultaneously. The convolution identity, of course, was derived from the Lagrange interpolation identity. It is remarkable that the new viewpoint, namely bulk-testing of the convolution identity using matrices, is rather powerful and achieves a much stronger bound than what was obtained by directly using the interpolation identity as a robust property.

One minor technical detail is that our tester works if the program computes a polynomial whose degree is precisely d, whereas the standard consideration in the literature is testing if the program computes a polynomial of degree at most d. To handle this, our self-tester will test for all $d' \leq d+1$, whether the program P computes a polynomial of degree precisely d'. If any of them certifies P, the self-tester outputs "PASS." While this increases the complexity of the tester, it does not increase the number of queries made to P, since the same queries can be reused by all the self-testers.

Theorem 7 *Univariate polynomials over the field $\mathbb{Z}/(p)$ are ϵ-self-testable for $\epsilon/2 = \delta < 2/9$. The self-tester makes $O(d)$ queries to the program, and computes $O(d)$ matrix products over $\mathbb{Z}/(p)$.*

Combining this with the work of Friedl and Sudan [FS95], we obtain[1] (see Appendix):

Theorem 8 *Multivariate polynomials over the field $\mathbb{Z}/(p)$ are ϵ-self-testable for $\epsilon/2 = \delta < 1/8$, provided p is large enough (as a function of $1/8 - \delta$). The self-tester makes $O(d)$ queries to the program, and computes $O(d)$ matrix products over $\mathbb{Z}/(p)$.*

6.2 Case (4): Testing recurrences on D_N^+

The result of this subsection is more general, and applies to arbitrary linear functions defined on the integers. The function $\varphi(n) = M^n$ that is of interest to us is just a special case.

Suppose φ is a linear function from $\mathbb{Z}$ to some group H, and suppose that a program P is purported to compute $\varphi(x)$ correctly for all $x \in D_N^+$. We show that P can be self-tested, provided it computes $\varphi(x)$ correctly not only over D_N^+, but does so also over D_{4N}^+, the neighborhood domain roughly four times larger. In the sequel, let $D_1 = D_N^+ = \{y \mid 0 \leq y < N\}$, $D_2 = \{y \mid 0 \leq y < 2N - 1\}$.

Theorem 9 *Suppose a program P satisfies the following condition for $\delta < 1/3456$:*

$$\Pr_{x \in D_2, y, z \in D_2}[P(x) \neq P(x + (y + z)) - P((y + z))] \leq \delta.$$

Then there is a function ψ such that

(1) $\Pr_{x \in D_1}[P(x) \neq \psi(x)] \leq 2\delta$, *that is, ψ agrees with P on most inputs, and*

(2) *For all $x, y \in D_1$, $\psi(x + y) = \psi(x) + \psi(y)$, that is, ψ is a linear function when restricted to inputs in D_1.*

The proof builds on the Blum–Luby–Rubinfeld proof as in Theorem 1, but is far more complicated since D_N^+ is not a group under integer addition. This entails us to dig deeper into the BLR proof, and use new counting arguments. We present the proof in the Appendix.

6.3 Testing over Rational Domains

Rational domains are of the form $D_{r,s} = \{\frac{t}{s} \mid |t| \leq r\}$ for $r, s \in \mathbb{Z}^+$. In the Appendix, we illustrate how to extend our scheme for testing univariate polynomials over rational domains. Our approach strengthens the result of Rubinfeld and Sudan [RS92] in two respects: the domain size on which the program is expected to be correct is much smaller, and the value of δ we achieve is $\Omega(1)$, whereas their result achieved only $O(1/d)$. More details of the following theorem are given in the Appendix.

Theorem 10 *Univariate polynomials of degree d over the rational domain $D_{r,s}$ (where inputs and the coefficients are in $D_{r,s}$) into the rational domain $D_{r^{d+1},s^{d+1}}$ (which suffices, and might be necessary) are ϵ-self-testable for $\epsilon/2 = \delta < 1/24$. The self-tester makes $O(d)$ queries to the program, and computes $O(d)$ matrix products over the domain $D_{dr^{2d},s^{2d}}$ (only slightly larger than the required domain).*

[1] Madhu Sudan [personal communication] has pointed out to us that an $\Omega(1)$-self-tester with $O(d)$ probes may be inferred from the work of Polishchuk and Spielman [PS94].

Acknowledgments

We thank Funda Ergün, Ronitt Rubinfeld, Madhu Sudan, and Ravi Sundaram for helpful conversations, and Jing Huang for discussions that led us to thinking about recurrences.

References

[ABCG93] S. Ar, M. Blum, B. Codenotti, and P. Gemmell. Checking approximate computations over the reals. In *Proc. 25th Annual ACM Symposium on the Theory of Computing*, pages 786–795, 1993.

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

[AS92] S. Arora and S. Safra. Probabilistic checking of proofs. In *Proc. 33rd Annual IEEE Symposium on Foundations of Computer Science*, pages 2–13, 1992.

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

[BLR93] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. Comp. Sys. Sci.*, 47(3):549–595, 1993. An earlier version appeared in STOC 1990.

[CL90] R. Cleve and M. Luby. A note on self-testing/correcting methods for trigonometric functions. TR 90-032, ICSI, Berkeley, July 1990.

[Erg95] F. Ergün. Testing multivariate linear functions: Overcoming the generator bottleneck. In *Proc. 27th Annual ACM Symposium on the Theory of Computing*, pages 407–416, 1995.

[FGL+91] U. Feige, S. Goldwasser, L. Lovasz, S. Safra, and M. Szegedy. Approximating clique is almost NP-complete. In *Proc. 32nd Annual IEEE Symposium on Foundations of Computer Science*, pages 2–12, 1991.

[FS95] K. Friedl and M. Sudan. Some improvements to total degree tests. In *Proc. 3rd Israel Symposium on Theory of Computing and Systems*, pages 190–198, 1995.

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

[Lip91] R. Lipton. New directions in testing. In *Proc. of DIMACS Workshop on Distributed Computing and Cryptography*, pages 191–202, 1991.

[PS94] A. Polishchuk and D. Spielman. Nearly-linear size holographic proofs. In *Proc. 26th Annual ACM Symposium on the Theory of Computing*, pages 194–203, 1994.

[RS92] R. Rubinfeld and M. Sudan. Testing polynomial functions efficiently and over rational domains. In *Proc. 3rd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 23–43, 1992.

[RS93] R. Rubinfeld and M. Sudan. Robust characterizations of polynomials and their applications to program testing. TR 93-1387, Dept. of Computer Science, Cornell University, 1993. To appear in *SIAM Journal of Computing.*

[RS95] S. Ravikumar and D. Sivakumar. On self-testing without the generator bottleneck. UBCS–TR 95-43, Computer Science Dept., University at Buffalo, 1995. To appear in *Proc. 15th FST & TCS*, Bangalore, India, 1995.

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

[Rub94] R. Rubinfeld. Robust functional equations with applications to self-testing/correcting. In *Proc. 35th Annual IEEE Symposium on Foundations of Computer Science*, pages 288–299, 1994.

Appendix

Example of the non-robustness of recurrences.

For any reasonably large value $\delta > \sqrt{2}/|D_N|$, suppose a program P purportedly computes the linear function $L(n) = n$ for all integers $n \in D_N$. In reality, however, P computes $\delta|D_N|$ different linear functions $L_1(n) = n$ for the first $\delta|D_N|$ inputs, $L_2(n) = 2n$ for the next $\delta|D_N|$ inputs, etc. Clearly P passes the test "$P(n) = 2P(n-1) - P(n-2)$" on all but $1/\delta$ of the inputs n, which is a $1/(\delta|D_N|)$ fraction of the domain. However, it is not possible to make a guarantee that P computes a linear function on all but $2/(\delta|D_N|)$ fraction of the inputs, since no linear function agrees with P on more than $\delta|D_N|$ inputs, and $2/(\delta|D_N|) < \delta|D_N|$.

Validity of the convolution identity for $L(n)$ over D_N.

For $k \geq 0$, define $L(N+k) = L(-(N-(k+1)))$, identifying N with $-(N-1)$, $N+1$ with $-(N-2)$, etc. Similarly, define $L(-N-k) = L(N-(k+1))$. Now consider the convolution identity $L(n) = L(m+1)L(n-m) - L(m)L(n-m-1)$. Clearly, if n and m are of the same sign, or if $|n-m| < N$, the identity is valid, therefore, wlog. let $n > 0$ and $m < 0$ such that $n - m > N$. We will require that $n - m \neq N$. Now,

$$\begin{aligned}
& L(m+1)L(n-m) - L(m)L(n-m-1) \\
= \;& L(m+1)L(N+(n-m-N)) - L(m)L(N+(n-m-1-N)) \\
= \;& L(m+1)L(-(N-(n-m-N+1))) - L(m)L(-(N-(n-m-1-N+1))) \\
= \;& -L(m+1)L(N-(n-m-N+1)) + L(m)L(N-(n-m-1-N+1)) \\
= \;& -L(N+N+m)L(-n+(N+N+m-1)) \\
& +L(N+N+m-1)L(-n+(N+N+m-1+1)) \\
= \;& L(N+N+m)L(n-(N+N+m-1)) \\
& -L(N+N+m-1)L(n-(N+N+m-1+1)) \\
= \;& L(n). \qquad \square
\end{aligned}$$

Proof of Theorem 1

Theorem 1 *If a program P satisfies the property*

$$\Pr_{n,m}[\omega P_n \neq \alpha P_{m+1} P_{n-m} + \beta P_m P_{n-m-1}] \leq \delta < \frac{1}{26},$$

then there is a function G such that G satisfies $\omega G_n = \alpha G_{m+1} G_{n-m} + \beta G_m G_{n-m-1}$ for all n, m, and such that $\Pr_n[P_n \neq G_n] \leq 2\delta$.

We will show the existence of the required function G by appropriately defining it. Define $G_n = \text{majority}_m\{(\alpha P_{m+1} P_{n-m} + \beta P_m P_{n-m-1})/\omega\}$. Claim 2 proves that G agrees with P on $1 - 2\delta$ fraction of the inputs, and Claim 5 proves that G satisfies the convolution identity everywhere. This, in particular, implies that the division is always valid in the ring K.

Claim 2 $\Pr_n[G_n = P_n] > 1 - 2\delta$.

Proof. Consider the set of numbers n such that

$$\Pr_m[\omega P_n = \alpha P_{m+1} P_{n-m} + \beta P_m P_{n-m-1}] < 1/2.$$

If the fraction of such elements is more than 2δ, then it contradicts the hypothesis that $\Pr_{n,m}[\omega P_n \neq \alpha P_{m+1} P_{n-m} + \beta P_m P_{n-m-1}] < \delta$. By defining G to be the majority, we have for the other elements $G_n = P_n$. Thus, (1) of Theorem 1 is proved. $\square$

Fact 3 *The probability that the same object is drawn twice from a set in two independent trials lower bounds the probability of drawing the most likely object in one trial.*

Proof. Let p_i be the probability of drawing object i. Let the objects be ordered such that $p_1 \geq p_2 \geq \cdots$. Then the probability of drawing same object twice is $\sum_i p_i^2 \leq \sum_i p_1 p_i = p_1 \sum_i p_i = p_1$. $\square$

Claim 4 *For all n,* $\Pr_m[\omega G_n = \alpha P_{m+1} P_{n-m} + \beta P_m P_{n-m-1}] > 1 - 4\delta$.

Proof. Since G is defined to be the majority, for any n,

$$\Pr_m[\omega G_n = \alpha P_{m+1} P_{n-m} + \beta P_m P_{n-m-1}]$$

is the probability of drawing the most likely object. Since this probability is at least as much as the probability of drawing the same object twice (Fact 3), which is

$$\Pr_{m,k}[\alpha P_{m+1} P_{n-m} + \beta P_m P_{n-m-1} = \alpha P_{m+k+1} P_{n-(m+k)} + \beta P_{m+k} P_{n-(m+k} - 1)],$$

it suffices to provide a lower bound for the latter quantity. We have

$$
\begin{aligned}
&= \Pr_{m,k}[\alpha P_{m+1}P_{n-m} + \beta P_m P_{n-m-1} \\
&= (1/\omega) * \Big[\alpha P_{m+1}(\alpha P_{k+1}P_{(n-m)-k} + \beta P_k P_{(n-m)-k-1}) \\
&\quad + \beta P_m(\alpha P_k P_{(n-m-1)-(k-1)} + \beta P_{k-1}P_{(n-m-1)-(k-1)-1})\Big] \\
&= (1/\omega) * \Big[\alpha^2 P_{m+1}P_{k+1}P_{n-m-k} + \alpha\beta P_{m+1}P_k P_{n-m-k-1} \\
&\quad + \alpha^2 P_m P_k P_{n-m-k} + \alpha\beta P_m P_{k-1}P_{n-m-k-1}\Big] \\
&= (1/\omega) * [(\alpha P_{m+1}P_{k+1} + \beta P_m P_k)\alpha P_{n-m-k} \\
&\quad + (\alpha P_{m+1}P_k + \beta P_m P_{k-1})\beta P_{n-m-k-1}] \\
&= \alpha P_{m+k+1}P_{n-(m+k)} + \beta P_{m+k}P_{n-(m+k)-1}] \geq 1 - 4\delta
\end{aligned}
$$

If m and k are distributed uniformly in H, the r.v.'s $n-m$ and $n-m-1$ are distributed uniformly and identically. Therefore, the first equality fails with probability $\leq 2\delta$ by hypothesis about P. The second and third equalities are merely regrouping of terms. Again, since the r.v.'s $m+k$ and $m+k-1$ are distributed uniformly, the last equality fails with probability $\leq 2\delta$. $\square$

Claim 5 *For all* $n, m, \omega G_n = \alpha G_{m+1}G_{n-m} + \beta G_m G_{n-m-1}$.

Proof.

$$
\begin{aligned}
&\mathrm{Pr}_{k,\ell}\,[\omega G_n \\
&= \alpha P_{k+1}P_{n-k} + \beta P_k P_{n-k-1} \\
&= (1/\omega)* \quad [\alpha P_{k+1} \quad (\alpha P_{m+\ell+1}P_{n-k-m-\ell} + \beta P_{m+\ell}P_{n-k-m-\ell-1}) \\
&\qquad + \beta P_k \quad (\alpha P_{m+\ell}P_{n-k-m-\ell} + \beta P_{m+\ell-1}P_{n-k-m-\ell-1})] \\
&= (1/\omega^2)* \quad [\alpha P_{k+1} \quad (\alpha P_{n-k-m-\ell} \quad \langle \alpha P_{k+\ell+1}P_{m-k+1} + \beta P_{k+\ell}P_{m-k}\rangle \\
&\qquad\qquad + \beta P_{n-k-m-\ell-1} \quad \langle \alpha P_{k+\ell}P_{m-k+1} + \beta P_{k+\ell-1}P_{m-k}\rangle) \\
&\qquad + \beta P_k \quad (\alpha P_{n-k-m-\ell} \quad \langle \alpha P_{k+\ell+1}P_{m-k} + \beta P_{k+\ell}P_{m-k-1}\rangle \\
&\qquad\qquad + \beta P_{n-k-m-\ell-1} \quad \langle \alpha P_{k+\ell}P_{m-k} + \beta P_{k+\ell-1}P_{m-k-1}\rangle)] \\
&= (1/\omega^2)* \quad [\alpha P_{k+1} \quad (\alpha P_{m-k+1} \quad \langle \alpha P_{k+\ell+1}P_{n-k-m-\ell} + \beta P_{k+\ell}P_{n-k-m-\ell-1}\rangle \\
&\qquad\qquad + \beta P_{m-k} \quad \langle \alpha P_{k+\ell}P_{n-k-m-\ell} + \beta P_{k+\ell-1}P_{n-k-m-\ell-1}\rangle) \\
&\qquad + \beta P_k \quad (\alpha P_{m-k} \quad \langle \alpha P_{k+\ell+1}P_{n-k-m-\ell} + \beta P_{k+\ell}P_{n-k-m-\ell-1}\rangle \\
&\qquad\qquad + \beta P_{m-k-1} \quad \langle \alpha P_{k+\ell}P_{n-k-m-\ell} + \beta P_{k+\ell-1}P_{n-k-m-\ell-1}\rangle)] \\
&= (1/\omega^2)* \quad [\alpha(\alpha P_{k+\ell+1}P_{n-k-m-\ell} + \beta P_{k+\ell}P_{n-k-m-\ell-1})(\alpha P_{k+1}P_{m-k+1} + \beta P_k P_{m-k}) \\
&\qquad + \beta(\alpha P_{k+\ell}P_{n-k-m-\ell} + \beta P_{k+\ell-1}P_{n-k-m-\ell-1})(\alpha P_{k+1}P_{m-k} + \beta P_k P_{m-k-1})] \\
&= \alpha G_{m+1}G_{n-m} + \beta G_m G_{n-m-1}] \geq 1 - 26\delta > 0
\end{aligned}
$$

The first equality fails with probability $\leq 4\delta$ by Claim 4. The second equality fails with probability $\leq 2\delta$ by two applications of the hypothesis in Theorem 1 about P. Note all the points are uniformly distributed in H since ℓ is chosen uniformly from H. Therefore, the third equality fails with probability $\leq 4\delta$ by four applications of the hypothesis about P (for $P_{m+\ell+1}, P_{m+\ell}$ and $P_{m+\ell-1}$). The fourth and fifth equalities are just regrouping of terms. The last equality fails with probability $\leq 16\delta$ by four applications of Claim 4. Since the equation is independent of k, ℓ and $\delta < 1/26$, we have $\omega G_n = \alpha G_{m+1} G_{n-m} + \beta G_m G_{n-m-1}$ hold always. □

Self-testers for convolutions

Using the robust characterization provided by Theorem 1, we now show how to build self-testers for convolutions over groups. The basic idea is the standard two-step process outlined in Section 3.

(1) Verify that the program satisfies the hypothesis of Theorem 1. If it does, then we are guaranteed that there is a function G that always satisfies the convolution.

(2) Verify that $F(0) = G(0)$ and $F(1) = G(1)$.

Let δ denote the probability that P fails to satisfy the convolution identity. By testing the convolution identity at $O((1/\delta^2)\ln(6/\beta))$ pairs of points, standard Chernoff bound arguments imply that with probability at least $1-\beta/2$, we may obtain an estimate of δ to within an error of at most $\delta/2$. The strategy to compute $G(0)$ and $G(1)$ relies on the following observation. The explicit form of the robust property yields a *self-corrector* for the function. The self-corrector lets us compute G correctly on every input (with high probability). In our case, to compute $G(n)$, the self-corrector randomly picks $m \in H$, computes $P(m+1), P(n-m), P(m)$, and $P(n-m-1)$, and uses the convolution identity to compute $G(n)$. Since $P(n) = G(n)$ for $1-2\delta$ fraction of the inputs, each of these calls fails with probability at most 2δ, and therefore $G(n)$ is correctly computed with probability at least $1-8\delta$. Since $\delta < 1/26$, this probability is at least 0.69. By repeating the experiment $O(\ln(1/\beta))$ many times and taking the majority, one may achieve success probability $1-\beta/2$ in computing $G(n)$ correctly, for every $n \in H$. In particular, one may correctly compute $G(1)$ and $G(2)$ with probability $1-\beta/2$, and check that these agree with the values of the true function, namely $F(1)$ and $F(2)$. Since G satisfies the convolution identity everywhere, by induction it is easy to show that $G \equiv F$. Also, the probability that the self-tester fails to do its job is at most β. We have the following theorem:

Theorem 2 *Functions $F : H \to K$, where H is a group and K is a ring, that satisfy second order convolutions of the form $\omega F(n) = \alpha F(m+1)F(n-m) + \beta F(m)F(n-m-1)$ have ϵ-self-testers for $\epsilon/2 = \delta < 1/26$.*

Proof of Lemma 3

Lemma 3 *If f is a genuine linear recurrence of order d, the vectors $\vec{f}_{d-1}, \vec{f}_d, \vec{f}_{d+1}, \ldots, \vec{f}_{2d-2}$ are all linearly independent over K (and hence over the quotient field of K).*

Proof. Suppose that for some k, $d \leq k \leq 2d-2$, $\vec{f}_k$ can be expressed as a linear combination of $\vec{f}_j$'s, for $j < k$. Since $\vec{f}_j = M^{j-d+1}\vec{f}_{d-1}$, this implies the equation

$$M^{k-d+1}\vec{f}_{d-1} = \left(\sum_{j=d-1}^{k-1} \alpha_j M^{j-d+1}\right)\vec{f}_{d-1}$$

for some constants $\alpha_j \in K$, not all zero. That is,

$$\begin{aligned}
\vec{f_k} &= (f_k f_{k-1} \dots f_{k-d+1})^T \\
&= M^{k-d+1} \vec{f}_{d-1} \\
&= \left(\alpha_{k-1} M^{k-d} + \alpha_{k-2} M^{k-d-1} + \alpha_{k-3} M^{k-d-2} + \dots \alpha_d M + \alpha_{d-1}\right) \vec{f}_{d-1}. \qquad (2)
\end{aligned}$$

Since $k \leq 2d-2$, $k-d \leq d-2$ and $k-d+1 \leq d-1$. The following fact may be easily verified by inspection: For $1 \leq j \leq d-1$, the last row of the matrix M^j has precisely one 1 in the $(d-j)$-th column and zeroes everywhere else. Let $m = k-d+1$. Let $\widehat{M}$ denote the matrix sum $\left(\sum_{j=d-1}^{k-1} \alpha_j M^{j-d+1}\right)$. By Equation (2), we know that f_m is the product of the last row of the matrix sum $\widehat{M}$ with the vector $\vec{f}_{d-1}$. We have

$$f_m = \left(\widehat{M}(d,1), \widehat{M}(d,2), \dots, \widehat{M}(d,d)\right)(f_{d-1}, f_{d-2}, \dots, f_0)^T \qquad (3)$$

By the fact noted above, for all $j \leq k-d$, M^j has only zeroes in the positions $(d,1)$, $(d,2)$, ..., $(d, d-(k-d)-1)$, that is, in the positions $(d,1)$, $(d,2)$, ..., $(d, d-m)$, and hence $\widehat{M}$ has zeroes in these positions. On the other hand, the values in these positions are precisely the coefficients of $f_{d-1}, f_{d-2}, \dots, f_m$ in Equation (3). In other words, f_m can be expressed in terms of $f_{m-1}, f_{m-2}, \dots, f_0$, and $m < d$. Let us rewrite Equation (2) in terms of $m = k-d+1$ and the matrix sum $\widehat{M}$ as

$$\vec{f}_{m+d-1} = (f_{m+d-1} f_{m+d-2} \dots f_m)^T = M^m \vec{f}_{d-1} = \widehat{M} \vec{f}_{d-1}. \qquad (4)$$

Now, for any $\ell \geq m$, by multiplying Equation (4) by $M^{\ell-m}$ gives

$$\begin{aligned}
\vec{f}_{\ell+d-1} &= (f_{\ell+d-1} f_{\ell+d-2} \dots f_\ell)^T \\
&= M^\ell \vec{f}_{d-1} \\
&= M^{\ell-m} M^m \vec{f}_{d-1} \\
&= \widehat{M}(M^{\ell-m} \vec{f}_{d-1}) \\
&= \widehat{M} \vec{f}_{\ell-m+d-1} \\
&= (f_{\ell-m+d-1} f_{\ell-m+d-2} \dots f_{\ell-m})^T \qquad (5)
\end{aligned}$$

Since the coefficients of $f_{\ell-m+d-1}, f_{\ell-m+d-2}, \dots, f_\ell$, namely the entries in the last row of $\widehat{M}$—$\widehat{M}(d,1), \widehat{M}(d,2), \dots, \widehat{M}(d, d-m)$—remain zero, it follows that f_ℓ may be expressed as the same linear combination (given by the last row of $\widehat{M}$) of the elements $f_{\ell-1}, \dots, f_{\ell-m}$. Since this is a linear recurrence of order strictly less than d and since $\ell \geq m$ was chosen arbitrarily, it follows that f can be written as a linear recurrence of order strictly less than d, a contradiction. □

Proof of Theorem 4

Theorem 4 *If P computes f perfectly, then Q computes φ perfectly, and the tester always outputs "PASS." Conversely, suppose the self-tester for Q outputs "PASS" and certifies that Q computes $\varphi(n) = M^n$ correctly for $1-\epsilon$ fraction of the inputs in the domain D. Then P computes the recurrence f correctly on (roughly) $1-\epsilon$ fraction of the inputs in the domain D.*

Proof. (Sketch.) The first part is obvious. For the converse, we first note that if M^n is correct, then *all* the entries of $\widehat{f}_{n+2d-2} = M^n \times A$ are correct. For if not, let $\widehat{f'}_{n+2d-2}$ denote the vector of correct values. Then, by definition, $M^n = \widehat{f'}_{n+2d-2} \times A^{-1}$. On the other hand, by assumption Q computes M^n correctly using the formula $M^n = \widehat{f}_{n+2d-2} \times A^{-1}$. Subtracting, $(\widehat{f'}_{n+2d-2} - \widehat{f}_{n+2d-2}) \times A^{-1} = 0$, contradicting the fact that the rows and columns of A^{-1} are linearly independent.

Next, we observe that for any n, $\widehat{f}_n$ has at least one new value of $P(\cdot)$ that is not present in $\widehat{f}_k$ for any $k < n$. Therefore, if $1 - \epsilon$ fraction of M^n's (and hence $\widehat{f}_n$'s) are correct, going from the smallest to the largest index of the correct $\widehat{f}_n$'s, we collect at least one new value for each $\widehat{f}_n$. This implies that at least $1 - \epsilon$ fraction of the values of $P(\cdot)$ are correct. □

A simple trick reduces the number of points where Q has to invoke P to compute M^n. To compute M^n, Q asks P for $f_n, \ldots, f_{n+d}$ only and computes $f_{n+d+1}, \ldots, f_{n+2d-2}$ for itself, using the recurrence.

Spectral Analysis of the matrix M

Let $f : \mathbb{Z} \to K$ be a recurrence of order d over the field K, and let M and A be the associated matrices over the multiplicative group $\mathcal{M}$ of $d \times d$ matrices over K. Recall that the *order* of M (denoted $ord(A)$) is the least integer $n \geq 0$ such that $M^n = I$, the identity matrix. As noted in Section 5, if N is the order of the matrix M, the function $\varphi : \mathbb{Z}/(N) \to \mathcal{M}$, defined by $\varphi(n) = M^n$, is in fact a homomorphism which can be self-tested using the BLR self-tester. Our goal is to investigate under what conditions M has small order.

The following fact may be easily verified by expanding along the last column of $M - xI$.

Fact 5 *The characteristic polynomial of M is given by $(-1)^{d-1}(-x^d + c_1 x^{d-1} + \cdots + c_d)$.*

Corollary 6 *The eigenvalues of M are given by the roots of the polynomial $\sum_{i=0}^{d} c_i x^{d-i}$, taking $c_0 = 1$.*

Two matrices A, B over a field K are called *similar* if there is a matrix C over K such that $A = CBC^{-1}$.

Fact 7 *If A and B are similar matrices, they have the same order.*

Proof. Let $A = CBC^{-1}$ and $ord(B) = n$. Then, $A^n = (CBC^{-1})^n = CB^nC^{-1} = CIC^{-1} = I$. Thus, $ord(B) \mid ord(A)$. The converse follows by the symmetry of similarity. □

A $m \times m$ matrix over a field K is in *elementary Jordan normal form* if it has some non-zero value $\lambda \in K$ along the main diagonal, 1 along the first sub-diagonal, and 0's elsewhere. Such a matrix is denoted by $J_{\lambda,m}$. A matrix J is in *Jordan normal form* if it is the direct sum of elementary Jordan normal matrices. The following is a standard fact from linear algebra.

Fact 8 *If A is an $n \times n$ matrix over field K and if the characteristic polynomial for A factors completely over K, then there is a unique (up to rearrangement of blocks) matrix B in Jordan normal form such that A and B are similar.*

The first interesting case we focus on is that of recurrences whose matrices have a Jordan normal form. Suppose the characteristic polynomial of M factors completely over K with eigenvalues $\lambda_1, \ldots, \lambda_k (k \leq d)$, where λ_i has multiplicity $m_i \geq 1$. By Fact 8, M is similar to a unique Jordan normal form matrix, which we will denote by J. In fact, J is a block diagonal matrix with blocks $J^{(1)}, \ldots, J^{(k)}$. Each $J^{(i)}$ is also a block diagonal matrix with blocks $J_1^{(i)}, \ldots, J_{n_i}^{(i)}$ where n_i is the dimension of the space of eigenvectors associated with λ_i. Each $J_j^{(i)}$ is an elementary Jordan normal form matrix $J_{\lambda_i, m_{ij}}$ where $m_{ij} (\leq n_i)$ results from the cyclic decomposition of $M - \lambda_i I$.

Now, computing the order of M is the same as computing the order of J. Since J is the direct sum of the $J_{\lambda_i, m_{ij}}$'s, it suffices to compute the order of the latter. It turns out that the exact value of the m_{ij}'s (the size of the elementary Jordan matrices) does not concern us—indeed, we will prove that the order of $J_{\lambda,m}$ is independent of m.

For $\lambda \in K$, define the order of λ, $ord(\lambda)$, in the usual way to be the least integer $n \geq 0$ such that $\lambda^n = 1$.

Lemma 9 *If $J_{\lambda,m} (m > 1)$ is an elementary Jordan matrix over K, $ord(J_{\lambda,m}) = ord(\lambda)p$.*

Proof. The lemma follows from analyzing the structure of $J_{\lambda,m}^k$ for any k. By induction, we can see that $J_{\lambda,m}^k$ is the following non-trivial lower triangular matrix: the i-th diagonal of $J_{\lambda,m}^k$ has $m - i$ elements, each of which is $\binom{k}{i}\lambda^{k-i}$. Thus, $J_{\lambda,m}^k$ has λ^k along the main diagonal, $k\lambda^{k-1}$ along the first sub-diagonal, and so on. To find $ord(J_{\lambda,m})$, we need the least n such that $J_{\lambda,m}^n = I$. That is, n has to satisfy the following two conditions: $\lambda^n = 1$ (to make main diagonal all 1's) and $\binom{n}{i}\lambda^{n-i} \equiv 0 (\bmod p)$ for all $i > 0$ (to make the rest of the matrix 0). The first condition forces $ord(\lambda) \mid n$. Since $\lambda^{n-i} \not\equiv 0 (\bmod p)$ for $\lambda \neq 0$, the second condition implies that $p \mid \binom{n}{i}$ for every $i > 0$. Taking $i = 1$, we have $p \mid n$. Combining the two, and noting that $(ord(\lambda), p) = 1$, we have $ord(\lambda)p \mid n$. Thus it clearly suffices if $ord(\lambda)p|n$. On the other hand, since $(ord(\lambda), p) = 1$, $ord(\lambda)p$ must be the smallest value that suffices. Thus, $n = ord(\lambda)p$ □

The following corollary follows by the observation that for every non-zero $\lambda \in K$, $ord(\lambda) \mid (p-1)$ by Fermat's theorem.

Corollary 10 *If M has all eigenvalues in K, $ord(M) \mid p(p-1)$.*

Corollary 11 *If M is diagonalizable over K, then $ord(M) \mid p-1$.*

Proof. If M is diagonalizable, there is a diagonal matrix J such that M is similar to J. A diagonal matrix is just a special case of a JNF matrix, where each of the elementary Jordan forms are simple 1x1 matrices. Since for each λ, $ord(J_{\lambda,1}) = ord(\lambda) \mid (p-1)$, the corollary follows. □

Applications for Polynomials over finite fields

We now consider polynomials over the finite field $K = \mathbb{Z}/(p)$, and show how to apply the results of the spectral analysis to this spcial case. Using the Lagrange interpolation formula, we can express degree d polynomials as linear recurrences of order $d+1$ as follows: $f_n = \sum_{i=1}^{d+1} \alpha_i f_{n-i}$, where the α_i's are fixed constants in K, given by $\alpha_i = (-1)^{i+1}\binom{d+1}{i}$. The matrix M that corresponds to a degree d polynomial, therefore, consists of the α_i's in the first row, and a copy of $[I_d \; 0]$ below it. We proceed to investigate the properties of this matrix.

We first note that M has a single eigenvalue 1 of multiplicity $d+1$. This can be seen as follows: from Fact 5, the characteristic polynomial in this case is given by $\sum_{i=1}^{d+1}(-1)^{d+i+1}\binom{d+1}{i}x^{d+1-i}$, which is the same as $(1-x)^{d+1}$. Next, we observe that the dimension of the span of its eigenvectors is 1. Let $v \in K^{d+1}$ be a vector satisfying $Mv = v$. Since the entries $2, 3, \ldots, d$ of Mv (reading from top to bottom) are the same as the entries $1, 2, \ldots, d-1$ of v, we have $v_0 = v_1 = \cdots = v_d$. Hence, the space of the eigenvectors is just the span of the vector $(1, 1, \ldots, 1)^T$, which has dimension 1.

Finally, let us compute the order of M. Since the characteristic polynomial of M factors completely over K, M has a Jordan normal form, which is just the matrix $J_{1,d+1}$, which has 1's along the main and the first sub-diagonal, and 0's elsewhere. Therefore, $ord(M) = ord(J_{1,d+1}) = ord(1)p = p$. In other words, the mapping $\varphi(n) = M^n$ for $n \in K$ is really a group homomorphism, since $M^p = I$.

In fact, we will strengthen the above observation considerably, and show that among all linear recurrences over $\mathbb{Z}/(p)$, polynomials are the only ones whose matrices M have order exactly p. Suppose the $d \times d$ matrix M corresponding to some linear recurrence over $K = \mathbb{Z}/(p)$ has order p, that is, $M^p = I$. Let χ denote the characteristic polynomial of the matrix M. By Fact 8, the matrix M has a unique Jordan normal form matrix J over the splitting field E of χ. The field E is, in fact, an algebraic extension of K obtained by adjoining all roots of χ to K, and wlog. we may take this to be the field $GF(p^d)$. It is an elementary observation that for any power J^k of J, the diagonal entries of J^k are precisely the k-th powers of the diagonal entries of J. Sine $J^p = I$, we have that the order (in E) of every diagonal entry of J must divide p, that is, they must have order either 1 or p. Since these entries are elements of the field $E = GF(p^d)$, their orders must divide $p^d - 1$. Therefore, their orders can't be p, so they must be 1. The only element whose order is 1 in any field is, of course, 1 itself. It follows that the only eigenvalue of the matrix M (over E) is 1, with multiplicity d. In other words, the characteristic polynomial of M must be $\pm(1-x)^d$. These two polynomials have exactly opposite signs for the term x^d. However, any polynomial that can be the characteristic polynomial for a $d \times d$ matrix must have the term $(-1)^d x^d$, which we know is true of the linear recurrence that arises from polynomials of degree $d-1$ over K. Therefore, the other polynomial is not the characteristic polynomial for any $d \times d$ matrix; in particular, there is no other recurrence of order d over K that has this as its characteristic polynomial. We have completed the proof of Theorem 6, which we state again for completeness.

Theorem 6 *The matrix M that corresponds to a degree d polynomial over $\mathbb{Z}/(p)$ has order exactly p. Moreover, this is the only recurrence whose matrix has order exactly p.*

Proof of Theorem 9

Theorem 9 *Suppose a program P satisfies the following condition for $\delta < 1/3456$:*

$$\Pr_{x \in D_2, y, z \in D_2}[P(x) \neq P(x+(y+z)) - P((y+z))] \leq \delta.$$

Then there is a function ψ such that

(1) $\Pr_{x \in D_1}[P(x) \neq \psi(x)] \leq 2\delta$, *that is, ψ agrees with P on most inputs, and*

(2) *For all $x, y \in D_1$, $\psi(x+y) = \psi(x) + \psi(y)$, that is, ψ is a linear function when restricted to inputs in D_1.*

Proof. We will show the existence of the required function ψ by appropriately defining it. For $x \in D_2$, we define $\psi(x)$ to be the value that appears most often in the multiset

$$\{P(x + (y + z)) - P(y + z) \mid y, z \in D_2\}$$

Lemma 10 proves that ψ agrees with P on most inputs in D_1, and Lemma 14 proves that ψ is linear on D_1.

Lemma 10 $\Pr_{x \in D_1}[\psi(x) = P(x)] > 1 - 4\delta$.

Proof. The hypothesis about P states that the event "$P(x) = P(x + (y + z)) - P(y + z)$" fails with probability at most δ, when x is chosen uniformly from D_2 and y and z are chosen from D_2 uniformly. Since $|D_1|/|D_2| > 1/2$, this event can fail with probability at most 2δ, when x is chosen uniformly from D_1. Consider the set of elements $x \in D_1$ such that $\Pr_{y,z \in D_2}[P(x) = P(x + (y + z)) - P(y + z)] < 1/2$. If the fraction of such elements is more than 4δ, we obtain a contradiction. By defining ψ to be the most popular element, we have for the other elements $\psi(x) = P(x)$. Thus, (1) of the theorem is proved. □

We will introduce a matrix that simplifies the rest of the analysis. For $x \in D_2$, define the matrix M_x to be a $(2N - 1) \times (2N - 1)$ matrix, whose rows are indexed by $y \in D_2$ and whose columns are indexed by $z \in D_2$. The entry $M_x(y, z) = P(x + (y + z)) - P(y + z)$. We note the following facts about the matrix M_x (for any x):

(1) For any position y, z, $M_x(y, z)$ equals $M_x(y + 1, z - 1)$, if the latter exists. In fact, for any j, $M_x(y, z) = M_x(y + j, z - j)$, if the latter exists. Thus the matrix has a certain redundancy property along the anti-diagonal and every sub-anti-diagonal.

(2) The hypothesis about P says that if x, y and z are chosen uniformly from D_2, the probability that $M_x(0, 0) = M_x(y, z)$ is at least $1 - \delta$. In other words, at least $1 - \delta$ fraction of all entries, over all the matrices M_x, agree with their topmost-leftmost entry $M_x(0, 0)$.

(3) Within any matrix M_x, given any fixed row y, although there are only $2p - 1$ elements along row y, the total redundancy of the elements of this row is more than half of the matrix M_x. More precisely, for any fixed x and y,

$$\Pr_{y_1, z_1 \in D_2}[M_x(y_1, z_1) \in \{M_x(y, z) \mid z \in D_2\}] > 1/2.$$

This can be proved as follows:

$$\begin{aligned} & \Pr_{y_1, z_1 \in D_2}[M_x(y_1, z_1) \in \{M_x(y, z) \mid z \in D_2\}] \\ = \;& \Pr_{y_1, z_1 \in D_2}[M_x(y_1, z_1) \in \{M_x(y, z) \mid z \in D_2\} \mid y \geq y_1] \cdot \Pr_{y_1, z_1 \in D_2}[y \geq y_1] \\ + \;& \Pr_{y_1, z_1 \in D_2}[M_x(y_1, z_1) \in \{M_x(y, z) \mid z \in D_2\} \mid y < y_1] \cdot (1 - \Pr_{y_1, z_1 \in D_2}[y \geq y_1]) \end{aligned}$$

In the first case, the event must be true if $z_1 \geq y - y_1$, for if so, the entry $M_x(y_1, z_1)$ must exist, and by Fact (1) must equal $M_x(y_1 + (y - y_1), z_1 - (y - y_1)) = M_x(y, z_1 - (y - y_1))$, which also must exist. Clearly, $z_1 \geq y - y_1$ iff $y_1 + z_1 \geq y$, which is certainly true if $y_1 + z_1 \geq 2N - 2$.

The latter event happens with probability more than $1/2$ when y_1 and z_1 are picked uniformly from D_2.

Similarly, in the second case, the event must be true if $z_1 \leq 2N-2-(y_1-y)$, for if so, the entry $M_x(y_1, z_1)$ must exist, and by Fact (1) must equal $M_x(y_1-(y_1-y), z_1+(y_1-y)) = M_x(y, z_1+(y_1-y))$, which also must exist. Clearly, $z_1 \leq 2N-2-(y_1-y)$ iff $y_1+z_1 \leq 2N-2+y$, which is certainly true if $y_1+z_1 \leq 2N-2$. The latter event happens with probability more than $1/2$ when y_1 and z_1 are picked uniformly from D_2.

(4) By definition, $\psi(x)$ is the value that occurs most often in M_x.

(5) By the probability fact 3 used in the proof of Theorem 1,

$$\begin{aligned}
& \Pr_{y,z}[\psi(x) = P(x+(y+z)) - P(y+z)] \\
= \; & \Pr_{y,z}[\psi(x) = M_x(y,z)] \\
\geq \; & \Pr_{y_1,z_1,y_2,z_2}[M_x(y_1,z_1) = M_x(y_2,z_2)] \\
= \; & \Pr_{y_1,z_1,y_2,z_2}[P(x+(y_1+z_1)) - P(y_1+z_1) = P(x+(y_2+z_2)) - P(y_2+z_2)],
\end{aligned}$$

hence if we wish to lower bound the first probability in this list, it suffices to provide a lower bound for the last probability.

Lemma 11 *Suppose*

$$\Pr_{y,z}[P(x+y) - P(y) = P(x+(y+z)) - P(y+z)] = \Pr_{y,z}[M_x(y,0) = M_x(y,z)] \geq 1-\rho.$$

Then for any $c > 0$,

$$\begin{aligned}
& \Pr_{y_1,z_1,y_2,z_2}[P(x+(y_1+z_1)) - P(y_1+z_1) = P(x+(y_2+z_2)) - P(y_2+z_2)] \\
= \; & \Pr_{y_1,z_1,y_2,z_2}[M_x(y_1,z_1) = M_x(y_2,z_2)] \\
\geq \; & 1-(3c\rho + 2/c).
\end{aligned}$$

Proof. By the hypothesis of the lemma, the expectation, when a row y is chosen uniformly, of the the fraction of z's s.t. $M_x(y,0) \neq M_x(y,z)$ is at most ρ. For any $c > 0$, by Markov's inequality, $\Pr_y$[the fraction of z's s.t. $M_x(y,0) \neq M_x(y,z) \geq c\rho] \leq 1/c$. Call a row y of M_x *useful* if the fraction of z's for which $M_x(y,0) \neq M_x(y,z)$ is at most $c\rho$. It is clear that at least $1-1/c$ fraction of the y's are useful. Now, suppose we pick uniformly two points (y_1, z_1) and (y_2, z_2) in M_x, and assume, wlog. that $y_1 \leq y_2$. From the anti-diagonal redundancy of M_x, it follows that $M_x(y_1, y_2-y_1) = M_x(y_2, 0)$. We will lower bound the probability of $M_x(y_1,z_1) = M_x(y_2,z_2)$ by restricting attention to the case where both y_1 and y_2 are useful, and z_1 and z_2 are "nice" columns along these rows. We will upper bound the probability that at least one of these niceness conditions fails. The probability that one of y_1 or y_2 is not a useful row is at most $2/c$. Given that y_1 and y_2 are useful, the probability that at least one of the events "$M_x(y_1,0) = M_x(y_1,z_1)$," "$M_x(y_1,0) = M_x(y_1,y_2-y_1)$," and "$M_x(y_2,0) = M_x(y_2,z_2)$" fails is at most $3c\rho$. Therefore, the probability that none of these five events fails is at least $1-(3c\rho+2/c)$. When they all happen, it is clear that $M_x(y_1,z_1) = M_x(y_2,z_2)$.

Lemma 12 *For all $x \in D_2$,*

$$\Pr_{y,z}[M_x(y,0) = M_x(y,z)] = \Pr_{y,z}[P(x+y) - P(y) = P(x+(y+z)) - P(y+z)] > 1 - 4\delta.$$

Proof. Fix $x \in D_2$. Consider the probability that $P(x+y) = P(x+y+z) - P(z)$. By rearrangement of terms, this probability is the same as the probability that $P(z) = P(z+(x+y)) - P(x+y)$, where y and z are chosen uniformly from D_2. If x also is a uniformly distributed element of D_2, then by the hypothesis about P, this probability would have been at least $1-\delta$. However, x is fixed; that is, we pick a matrix M_z uniformly, and a column y within it uniformly, fixing the row to be x; but Fact (3) about the matrix M_z comes to our rescue: for any fixed z, the density of the elements in row x of M_z is at least $1/2$, and hence, the total probability density of the elements of row x, over all z's, is at least $1/2$. Therefore, if δ is the total probability of failure of the event "$P(z) = P(z+(x+y)) - P(x+y)$," over all x's and y's and z's, then for any fixed x, this event can fail with probability $\leq 2\delta$ (over y's and z's). Summarizing this discussion, we have shown that

$$\Pr_{y,z \in D_2}[P(x+y) = P(x+y+z) - P(z)] \geq 1 - 2\delta.$$

By a similar discussion, we obtain

$$\Pr_{z \in D_2}[P(y) = P(y+z) - P(z)] = \Pr_{z \in D_2}[P(z) = P(z+(0+y)) - P(0+y)] \geq 1 - 2\delta.$$

Combining the two, we conclude that

$$\begin{aligned}
& \Pr_{y,z \in D_2}[P(x+y) - P(y) = P(x+y+z) - P(z) - P(y+z) + P(z)] \\
= & \Pr_{y,z \in D_2}[P(x+y) - P(y) = P(x+y+z) - P(y+z)] \\
\geq & \ 1 - 4\delta,
\end{aligned}$$

and the lemma is proven. □

Lemmas 11 and 12, together with Fact (5) about the matrix M_x, imply:

Corollary 13 *For any $c > 0$, for all $x \in D_2$, $\Pr_{y,z \in D_2}[\psi(x) = P(x+(y+z)) - P(y+z)] > 1 - (12c\delta + 2/c)$.*

Lemma 14 *For all $x, y \in D_1$, $\psi(x) + \psi(y) = \psi(x+y)$.*

Proof. Fix $x, y \in D_1$. If there exists at least one $z \in D_1$ such that all of the following three claims hold, then clearly the lemma holds for x and y.

(1) $\psi(x) = P(x+y+z) - P(y+z)$.

(2) $\psi(y) = P(y+z) - P(z)$.

(3) $\psi(x+y) = P(x+y+z) - P(z)$.

We will show the existence of at least one such z probabilistically. Consider a z picked uniformly from D_1. Now, the following three claims can be proven to methods similar to the proof of lemma 12, for any $c > 0$. Here too, the crucial fact is the following: for any matrix M_x, the values along any fixed row y of M_x have probability measure at least $1/2$ within M_x. Therefore, if the event $\psi(x) = P(x+y+z) - P(y+z)$ fails with probability at most ρ when y and z are chosen uniformly from D_2, it fails with probability at most 2ρ when y is fixed to an arbitrary value in D_2, and z is chosen uniformly.

(1) $\Pr[\psi(x) = P(x+y+z) - P(y+z)] \geq 1 - 2(12c\delta + 2/c)$.

(2) $\Pr[\psi(y) = P(y+z) - P(z)] \geq 1 - 2(12c\delta + 2/c)$.

(3) $\Pr[\psi(x+y) = P(x+y+z) - P(z)] \geq 1 - 2(12c\delta + 2/c)$.

Therefore, the existence of a good z occurs with probability at least $1 - 6(12c\delta + 2/c)$, which is non-zero if $\delta < 1/(72c) - 1/(6c^2)$. The quantity $1/(72c) - 1/(6c^2)$ is maximized when $c = 24$, which gives us $\delta < 1/3456$. □

Proof of Theorem 8

Theorem 8 *Multivariate polynomials over the field $\mathbb{Z}/(p)$ are ϵ-self-testable for $\epsilon/2 = \delta < 1/8$. The self-tester makes $O(d)$ queries to the program, and computes $O(d)$ matrix products over $\mathbb{Z}/(p)$.*

Proof. (Sketch.) The proof combines Theorem 7 with the work of Friedl and Sudan [FS95], who present a reduction from the multivariate self-testing problem to the univariate self-testing problem. For $x, h \in K^m$, let $l_{x,h}$ denote the line $\{x + th \mid t \in K\}$, and let $f^{x,h}$ denote the polynomial $f^{x,h}(t) = f(x+th)$. This is sometimes called the "line polynomial" along the line $l_{x,h}$. A function f is an m-variate polynomial of total degree at most d over $\mathbb{Z}/(p)$ if and only if every line x, h, the line polynomial $f^{x,h}$ is a univariate polynomial of degree at most d in the variable t. Based on this characterization, Friedl and Sudan prove the following:

Theorem 9 ([FS95]) *For every $\rho > 0$, there is a constant $c = c(\rho)$ such that for all d, if $|K| \geq cd$, then given any $P : K^m \to K$ and degree d polynomials $\{f^{x,h}\}$ satisfying $\Pr_{x,h,t}[f^{x,h}(t) \neq P(x + th)] = \delta \leq 1/8 - \rho$, then there is a polynomial g of total degree d such that g agrees with P on $\geq 1 - 2\delta$ fraction of the inputs.*

While their result provides a very efficient "low-degree tester," it does not give a program self-tester because it assumes that the tester has oracle access to the "correct" line polynomials $f^{x,h}$. If we attempt to recursively test along many randomly chosen lines using the known univariate self-testers, they require $\delta = O(1/d)$. However, using the self-tester we built in Theorem 7, this task can be accomplished.

More precisely, the self-tester for multivariate polynomials works as follows: pick many random $x, h \in K^m$, and test if P agrees with a univariate polynomial of degree at most d on at least $1 - \delta_1$ fraction of the inputs along the line $l_{x,h}$. Suppose the fraction of lines $l_{x,h}$ along which P fails the δ_1-self-tester for univariate polynomials is at most δ_2. Combining the two, we have $\Pr_{x,h,t}[f^{x,h}(t) \neq P(x+th)] \leq \delta_1 + \delta_2$. If $\delta = \delta_1 + \delta_2 \leq 1/8 - \rho$, then we can conclude that P agrees with a polynomial g of total degree at most d on $1 - 2\delta$ fraction of the inputs in K^m. □

Testing over Rational Domains: Proof of Theorem 10

Theorem 10 *Univariate polynomials of degree d over the rational domain $D_{r,s}$ (where inputs and the coefficients are in $D_{r,s}$) into the rational domain $D_{r^{d+1},s^{d+1}}$ (which suffices, and might be necessary) are ϵ-self-testable for $\epsilon/2 = \delta < 1/24$. The self-tester makes $O(d)$ queries to the program, and computes $O(d)$ matrix products over the domain $D_{dr^{2d},s^{2d}}$ (only slightly larger and finer than the required domain).*

Proof. (Sketch.) We will assume that $r \geq d$. Let us denote by $D_{u,v}$ the domain $D_{r^{d+1},s^{d+1}}$, and by $D_{u',v'}$ the domain $D_{dr^{2d},s^{2d}}$. The domain $D_{r,s}$ is isomorphic to the domain D_{N+1}, where $N = rs$, if we view the domain $D_{r,s}$ as built up of "units" of size $1/s$. Now, we can use the ideas similar to the case of recurrences from D_N into a field K. The domain $D_{u,v}$ is not a field, of course, but as we shall see, this does not pose a problem.

The interpolation identity for polynomials is still valid, and the matrix M is exactly as before. The self-tester builds, as before, the program Q, which takes $n \in D_N$, and computes M^n by calling the program P being tested at the points $(n+d-1)/s, (n+d)/s, \ldots, (n+2d-3)/s, (n+2d-2)/s$, and multiplying the resulting matix of values of P by the matrix A^{-1}. It can be shown that A^{-1} must be a matrix with entries in $D_{dr^{2d},s^{2d}} = D_{u',v'}$. Notice that, in contrast to the approach of Rubinfeld and Sudan [RS92], we do not expect the program P to work on a finer domain than required (P must be expected to work in $D_{u,v}$ anyway, since it is the smallest domain that contains all values of degree d polynomials over $D_{r,s}$). The overhead is now shifted to the self-testing program, which also works in a domain only slightly larger and slightly (quadratically) finer. The rest is as before, similar to the case of linear recurrences over D_N, borrowing on the technique of [Rub94, GLR+91, RS92]. $\square$

In fact, the proof in [RS92], when specialized to linear functions, does not require a finer-precision domain (as noted in [Rub94]). Since our matrix approach effectively reduces the polynomial testing problem to the linearity testing problem, this is just another benefit that we inherit!