Approximate Checking of Polynomials and Functional Equations *

(Extended Abstract)

Funda Ergün S Ravi Kumar Ronitt Rubinfeld
Department of Computer Science
Cornell University
Ithaca, NY 14853.

Abstract

In this paper, we show how to check programs that compute polynomials and functions defined by addition theorems — in the realistic setting where the output of the program is approximate instead of exact. We present results showing how to perform approximate checking, self-testing, and self-correcting of polynomials, settling in the affirmative a question raised by [GLR+91, RS92, RS96]. We then show how to perform approximate checking, self-testing, and self-correcting for those functions that satisfy addition theorems, settling a question raised by [Rub94]. In both cases, we show that the properties used to test programs for these functions are both robust (in the approximate sense) and stable. Finally, we explore the use of reductions between functional equations in the context of approximate self-testing. Our results have implications to the stability theory of functional equations.

1. Introduction

Program checking, which uses the program to verify its own correctness, was first formalized and developed in [BK89]. Related notions of self-testing and self-correcting were further explored in [BLR93, Lip91]. These notions have proved to be very powerful from a practical point of view (e.g., [BW94]) and from a theoretical angle (e.g., [AS92, ALM+92]) as well. The techniques used usually consist of tests which compare the output of the program either to a predetermined value or to a function of outputs of the same program at different inputs. In order to apply these powerful techniques to actual programs, however, several issues dealing with *precision* need to be dealt with. The standard model, which considers an output to be wrong even if it is off by a very small margin, is too strong to make practical sense, due to reasons such as: (1) in many cases, the algorithm is only intended to compute an approximation, e.g., Newton's method; (2) due to representational limitations, roundoff/truncation errors are inevitable in real-valued computations; (3) the representation of some fundamental constants (e.g., $\pi = 3.14159\ldots$) is inherently imprecise.

The framework presented by [GLR+91, ABC+93] accommodates these inherently inevitable or acceptably small losses of information, by overlooking small precision errors while detecting actual "bugs", which manifest themselves with greater magnitude. Given a function f, a program P that purports to compute f, and an error bound Δ, if $|P(x) - f(x)| \leq \Delta$ (denoted $P(x) \approx_\Delta f(x)$) under some appropriate notion of norm, we say $P(x)$ is *approximately correct*. *Approximate result checkers* test if P is approximately correct for a given input x. *Approximate self-testers* are programs that test if P is approximately correct for most inputs. *Approximate self-correctors* take programs that are approximately correct on most inputs and turn them into programs that are approximately correct on every input.

We define the *approximation* of a function by another:

Definition 1 *P (Δ, ϵ)-approximates f on domain $\mathcal{D}$ if $|P - f| \leq \Delta$ on $\geq 1 - \epsilon$ fraction of $\mathcal{D}$.*

DOMAINS. We operate on finite subsets of fixed point arithmetic that we refer to as *finite rational domains*. For $n, s \in \mathbb{Z}^+$, $\mathcal{D}_{n,s} \stackrel{\rm def}{=} \{\frac{i}{s} : |i| \leq n, i \in \mathbb{Z}\}$. Usually, $s = 2^l$ where l is the precision. We allow s and n to vary for generality. For a domain $\mathcal{D}$, let $\mathcal{D}^+$ and $\mathcal{D}^-$ denote the positive and negative elements in $\mathcal{D}$.

TESTING USING PROPERTIES. A property $\mathcal{I}$ is an equation $\mathcal{I}^f(x_1, \ldots, x_k) = 0$ that relates the values of function f at various tuples of locations $\langle x_1, \ldots, x_k\rangle$ coming from a

*This work is partially supported by ONR Young Investigator Award N00014-93-1-0590 and the Alfred P. Sloan Research Award. The first and second authors are also supported by NSF grant DMI-91157199. The third author is also supported by an NSF VPW award and grant No. 92-00226 from the United States–Israel Binational Science Foundation (BSF), Jerusalem, Israel. Part of this research was conducted while visiting the M.I.T. Lab. for Computer Science.

certain distribution $\mathcal{E}$ over $\mathcal{D}^k$. $\langle \mathcal{I}, \mathcal{E} \rangle$ characterizes a function family $\mathcal{F}$ where $f \in \mathcal{F}$ if and only if $\forall \langle x_1, \ldots, x_k \rangle$ with non-zero support coming from $\mathcal{E}$, $\mathcal{I}^f(x_1, \ldots, x_k) = 0$. An important technique for self-testing programs has been to identify "robust" properties for $\mathcal{F}$ [RS92]. Informally, the $(\mathcal{D}, \mathcal{D}')$ robustness of $\langle \mathcal{I}, \mathcal{E} \rangle$ in a function family $\mathcal{G}$ implies that if, for a program $P \in \mathcal{G}$, $\mathcal{I}^P(x_1, \ldots, x_k) = 0$ is satisfied with high probability (a constant independent of $|\mathcal{D}|$) when $\langle x_1, \ldots, x_k \rangle$ is chosen from the distribution $\mathcal{E}$ over $\mathcal{D}^k$, then there is a function $g \in \mathcal{F} \cap \mathcal{G}$ that agrees with P on a large fraction of the inputs in $\mathcal{D}'$. (Membership in $\mathcal{G}$ should be easy to test and $\mathcal{D}'$ should be a large subset of $\mathcal{D}$. All of our checkers satisfy these requirements). Once it is determined via random sampling that P satisfies the robust property, then it is very easy to determine via further random sampling whether P agrees on most inputs with a *specific* member of $\mathcal{F} \cap \mathcal{G}$ (*equality testing*). For example, consider the linearity property $\langle \mathcal{I}, \mathcal{E} \rangle$ where $\mathcal{I}^f(x_1, x_2, x_3) \equiv$ "$f(x_1) + f(x_2) = f(x_3)$" and $\mathcal{E}$ is $\langle x_1 \in_R \mathbb{Z}_p, x_2 \in_R \mathbb{Z}_p, x_1 + x_2 \rangle$. $\langle \mathcal{I}, \mathcal{E} \rangle$ characterizes $\mathcal{F} = \{f(x) = cx \mid c \in \mathbb{Z}_p\}$, the set of all linear functions over $\mathbb{Z}_p$. In this simple example, $\mathcal{G}$ is trivially the set of all functions (in some cases discussed in this paper, it is useful for $\mathcal{G}$ to be a smaller set of functions). $\langle \mathcal{I}, \mathcal{E} \rangle$ is known to be robust in $\mathcal{G}$. Therefore, once it is tested that P satisfies $P(x_1) + P(x_2) = P(x_1) + P(x_2)$ with large enough probability where $x_1, x_2 \in_R \mathbb{Z}_p$, it is possible to conclude that P agrees with some linear function on most inputs. Then, for instance, to ensure that P computes the specific linear function $f(x) = x$ on most inputs, we can perform the equality test which ensures that $P(x+1) = P(x) + P(1)$ for most x. (Note that the equality test on its own is not sufficient for testing the program). This combined approach yields extremely efficient testers for programs computing homomorphisms (e.g., multiplication of integers and matrices, exponentiation, logarithm). This idea is further generalized in [Rub94], where the class of functional equations called *addition theorems* are shown to be useful for self-testing. An addition theorem is a mathematical identity of the form $\forall x, y, f(x+y) = G[f(x), f(y)]$. Addition theorems characterize many useful and interesting mathematical functions [Acz66, CR92]. When G is algebraic, they can be used to characterize families of functions that are rational functions of x, e^{cx}, and doubly periodic functions (see Table 1 for examples). Polynomials of degree d can be characterized via several different robust functional equations (e.g., [BFL91, Lun91, ALM$^+$92, RS96]).

APPROXIMATE ROBUSTNESS AND STABILITY. When the program works with finite precision, the properties upon which the testers are built will rarely be satisfied, even by a good program, since they involve equalities. Thus, when testing, one might be willing to pass programs for which the properties are only approximately satisfied. This relaxation in the tests, however, causes problems, for in the approximate setting (1) it is much harder to analyze which function families are solutions to the robust properties, and (2) equality testing is more difficult.

To construct approximate self-testers, our approach is to first investigate a notion of *approximate robustness* of the property to be used. Formally, a property $\langle \mathcal{I}, \mathcal{E} \rangle$ for a function family $\mathcal{F}'$ is $(\delta, \epsilon, \Delta, \Delta', \Delta'', \mathcal{D}, \mathcal{D}')$-*approximately robust* in $\mathcal{G}$ if $\forall P \in \mathcal{G}$, $\Pr_{x_1, \ldots, x_k \in \mathcal{E}}[\mathcal{I}^P(x_1, \ldots, x_k) \approx_\Delta 0] \geq 1 - \epsilon$ implies there is a $g \in \mathcal{F}' \cap \mathcal{G}$ that (Δ'', δ)-approximates P and $\mathcal{I}^g(x_1, \ldots, x_k) \approx_{\Delta'} 0$ for all tuples $\langle x_1, \ldots, x_k \rangle$ with with nonzero support in $\mathcal{E}$ such that $x_i \in \mathcal{D}'$, $1 \leq i \leq k$. Once we know that the property is approximately robust in $\mathcal{G}$, the second step is to analyze the *stability* of the property, i.e., to characterize the set of functions $\mathcal{F}'$ that satisfy the property approximately (the *Hyers-Ulam stability*) and compare it to $\mathcal{F}$, the set of functions that satisfy the property exactly. In our linearity example, the problem is the following: given g satisfying $|g(x+y) - g(x) - g(y)| \leq \Delta$ for all x, y in the domain, is there a homomorphism h that $(\Delta', 0)$-approximates g with Δ' depending only on Δ and *not* on the size of the domain? If the answer is affirmative, we say that the property is *stable*. If a property is both approximately robust and stable, then it can be used to determine whether P approximates some function in the desired family. If furthermore we have a method of doing approximate equality testing, then we can construct an approximate self-tester (for details, see Appendix B).

There has been significant work on the stability of specific functional equations. The stability of linearity and other homomorphisms is addressed in [Hye41, For80, FS89, Cho84]. The techniques used to prove the above results, however, cease to apply in the practical setting when the domain is not a closed mathematical space (e.g., a finite precision rational domain). The stronger property of stability in a non-closed space, called *local stability*, is addressed by Skof [Sko83] who proves that Cauchy functional equations are locally stable on a finite interval in $\mathbb{R}$. The problem of stability of univariate polynomials over continuous domains is first addressed in [AB83] and the problem of local stability on $\mathbb{R}$ is solved in [Gaj90]. See [For95] for a survey. The above results are not always extendible in an obvious manner to discrete domains. For those that can be extended, the error bounds obtained by naive extensions are not optimal. Our different approach allows us to operate on $\mathcal{D}_{n,s}$ *and* obtain tight bounds.

PREVIOUS WORK AND RESULTS. Previously, not many of the known checkers have been extended to the approximate case. The issue is first mentioned in [GLR$^+$91], where approximate checkers for mod, exponentiation, and logarithm are constructed. The domain is assumed to be closed in all of these results. In [ABC$^+$93] approximate checkers for $\sin$, $\cos$, matrix multiplication, matrix inversion, lin-

$G[f(x), f(y)]$	$f(x)$
$f(x) + f(y)$	Ax
$\frac{f(x)+f(y)}{1-f(x)f(y)}$	$\tan Ax$
$\frac{f(x)f(y)-1}{f(x)+f(y)}$	$\cot Ax$
$\frac{f(x)+f(y)-1}{2f(x)+2f(y)-2f(x)f(y)-1}$	$\frac{1}{1+\tan Ax}$
$\frac{f(x)+f(y)-2f(x)f(y)}{1-f(x)f(y)}$	$\frac{-Ax}{1-Ax}$
$\frac{f(x)+f(y)}{1+[f(x)f(y)]/A^2}$	$A \tanh Bx$
$\frac{f(x)f(y)}{f(x)+f(y)}$	$\frac{A}{x}$

$G[f(x), f(y)]$	$f(x)$
$f(x)f(y) - \sqrt{1-f(x)^2}\sqrt{1-f(y)^2}$	$\cos Ax$
$\frac{f(x)+f(y)-2f(x)f(y)}{1-2f(x)f(y)}$	$\frac{1}{1+\cot Ax}$
$\frac{f(x)+f(y)-2f(x)f(y)\cos a}{1-f(x)f(y)}$	$\frac{\sin Ax}{\sin Ax+a}$
$\frac{f(x)+f(y)-2f(x)f(y)\cosh a}{1-f(x)f(y)}$	$\frac{\sinh Ax}{\sinh Ax+a}$
$\frac{f(x)+f(y)+2f(x)f(y)\cosh a}{1-f(x)f(y)}$	$\frac{-\sinh Ax}{\sinh Ax+a}$
$\frac{f(x)+f(y)+2f(x)f(y)}{1-f(x)f(y)}$	$\frac{Ax}{1-Ax}$
$f(x)f(y) + \sqrt{f(x)^2-1}\sqrt{f(y)^2-1}$	$\cosh Ax$

Table 1. Some Addition Theorems of the form $f(x+y) = G[f(x), f(y)]$.

ear system solving, and determinant are given. The domain is assumed to be closed in the results on $\sin$ and $\cos$. In [BW94] an approximate checker for floating-point division is given. In [Sud91], a clever technique using approximation theory is used to test univariate polynomials of degree ≤ 9. It is left open in [GLR$^+$91, ABC$^+$93, RS96, Rub94] whether the properties used to test polynomial, hyperbolic, and other trigonometric functions are stable. Showing the stability of such functional equations is not obvious; for instance, if the functional equation involves division with a large numerator and a small denominator, a small additive error in the denominator leads to a large additive error in the output.

In this paper, we answer the questions of [GLR$^+$91, ABC$^+$93, RS96, Rub94] in the affirmative, by giving the first approximate versions of most of their testers. All of our results apply to non-closed discrete domains. We first present an approximate tester for linear and multilinear functions with tight bounds. These results apply to several functions, including multiplication, exponentiation, and logarithm, over non-closed domains. We next present the first approximate testers for multivariate polynomials. Finally, we show how to approximately test functions satisfying addition theorems. Our results apply to many algebraic functions of trigonometric and hyperbolic functions (e.g., $\sinh$, $\cosh$). See Appendix B for details of the self-testers.

To do this, we prove new local stability results for discrete domains. Our techniques for showing the stability of multilinearity differ from those used previously in that (1) they do not require the domain to be discrete and (2) they do not require the range to be a complete metric space. This allows us to apply our results to multivariate polynomial characterizations. In addition to new combinatorial arguments, we employ tools from approximation theory and stability theory. Our techniques appear to be more generally applicable and cleaner to work with than those previously used.

Self-correctors are built by taking advantage of the random self-reducibility of polynomials and functional equations [BLR93, Lip91] in the exact case. Details are given in Appendix A. To build an approximate checker for all of these functions, we combine the approximate self-tester and approximate self-corrector as in [BLR93].

DEFINITIONS. The following definitions from [GLR$^+$91] capture the idea of approximate checking, self-testing, and self-correcting in a formal manner. Let P denote a program for f, x an input to P, and β the confidence parameter.

Definition 2 *A* Δ-approximate result checker *for f is a probabilistic oracle program T that, given P, x, and β, satisfies the following:*

(1) $P(x) = f(x) \Rightarrow \Pr[T^P$ *outputs "PASS"*$] \geq 1 - \beta$.

(2) $P(x) \not\approx_\Delta f(x) \Rightarrow \Pr[T^P$ *outputs "FAIL"*$] \geq 1 - \beta$.

Definition 3 *A* $(\Delta_1, \epsilon_1, \Delta_2, \epsilon_2)$-approximate self-tester *for f is a probabilistic oracle program T that, given P and β, satisfies the following:*

(1) P (Δ_1, ϵ_1)*-approximates* f $\Rightarrow$ $\Pr[T^P$ *outputs "PASS"*$] \geq 1 - \beta$.

(2) P *does not* (Δ_2, ϵ_2)*-approximate* f $\Rightarrow$ $\Pr[T^P$ *outputs "FAIL"*$] \geq 1 - \beta$.

Definition 4 *A* $(\Delta, \epsilon, \Delta')$-approximate self-corrector *for f is a probabilistic oracle program $P_{\rm sc}$ that, given P that (Δ, ϵ)-approximates f, x, and β, outputs $P_{\rm sc}(x)$ such that* $\Pr[P_{\rm sc}(x) \approx_{\Delta'} f(x)] \geq 1 - \beta$.

We require a notion of distance between two functions.

Definition 5 (Chebyshev Norm) *For a function f on a domain $\mathcal{D}$,* $|f|_{\mathcal{D}} = |f| = \sup_{x \in \mathcal{D}} \{|f(x)|\}$.

When the domain is obvious from the context, we drop it. Given functions f, g, the *distance* between them is $|f - g|$.

2. Linearity and Multilinearity

In this section, we consider the problem of approximate linearity and multilinearity over the finite rational domain $\mathcal{D}_{n,s}$, which is not a group. The results in this section, in addition to being interesting in themselves, are crucial to our results in Section 3.

Approximate robustness is easy to show by appropriately modifying the proof of robustness [Rub94]. To show stability, we use two types of bootstrapping arguments: the first shows that an error bound on a small subset of the domain implies the same error bound on a larger subset of the domain; the second shows by contradiction that an error bound on the whole domain implies a tighter error bound over the same domain. These results can be applied to give the first approximate self-testers for several functions over $\mathcal{D}_{n,s}$ including multiplication, exponentiation, and logarithm.

2.1. Approximate Linearity

A function g is *Δ-approximately linear* if $\forall x, y \in \mathcal{D}_{n,s}, g(x+y) \approx_\Delta g(x) + g(y)$. Hyers [Hye41] and Skof [Sko83] obtain a linear approximation to an approximately linear function when the domain is $\mathbb{R}$. Their methods are not extendible to discrete domains. In [GLR$^+$91], a solution is given when the domain is a finite group. Their bootstrapping technique requires that the domain be closed under addition, and therefore does not work for $\mathcal{D}_{n,s}$. Instead, we employ a different constructive technique to obtain a linear h on $\mathcal{D}_{n,s}$ given a Δ-approximately linear g. Our technique yields a tight bound of 2Δ on the error $e \equiv h - g$ (instead of 4Δ in [Sko83]) and does not require that the domain be closed under addition. Recall that $\frac{n}{s}$ is the greatest positive element of the domain, and note that e is a Δ-approximately linear function. The following lemma shows how to construct a linear function h that is within $2\Delta + \rho$ of a Δ-approximately linear function g in $\mathcal{D}^+_{n,s}$.

Lemma 6 *Let g be a Δ-approximately linear function on $\mathcal{D}^+_{n,s}$, and let h be linear on $\mathcal{D}_{n,s}$. Define $e(x) = h(x) - g(x)$. If $|e(\frac{n}{s})| = \rho$, then $\forall x \in \mathcal{D}^+_{n,s}, |e(x)| \leq 2\Delta + \rho$.*

Proof. We prove by contradiction that $\forall x \in \mathcal{D}^+_{n,s}, e(x) \leq 2\Delta + \rho$. A symmetric argument can be made to show that $e(x) \geq -(2\Delta + \rho)$.

Assume that there exists a point in $\mathcal{D}^+_{n,s}$ with error $> 2\Delta + \rho$. Let p be the maximal such element. p has to lie between $\frac{n}{2s}$ and $\frac{n}{s}$, otherwise $2p \in \mathcal{D}^+_{n,s}$ would have error $> 2\Delta+\rho$, contradicting the maximality of p. Let $q = \frac{n}{s} - p$. Then, $e(q)+e(p) \approx_\Delta e(\frac{n}{s})$, therefore $e(q) < -\Delta$. Also, for any $x \in (p, \frac{n}{s}] \subseteq \mathcal{D}^+_{n,s}$, by definition of p, $e(x) \leq 2\Delta + \rho$. Note that any such x can be written as $x = x' + p$, where $x' \in (0, q]$. To satisfy the approximate linearity property that $e(x') + e(q) \approx_\Delta e(x)$, x' can have error at most Δ.

We now know that the points in the interval $(0, q]$ have error at most 2Δ (in fact, at most Δ), and that the point q itself has error strictly less than $-\Delta$. Putting these two facts and approximate linearity together, and since any $x \in (q, 2q]$ can be written as $q + y$ where $y \in (0, q]$, we can conclude that at any point in $(q, 2q]$, the error is at most $2\Delta + \rho$. Now we can repeat the same argument by taking y from $(0, 2q]$ rather than $(0, q]$ to bound the error in the interval $(0, 3q]$ by $2\Delta+\rho$. By a continuing argument, eventually the interval contains the point p, which means that p has error $\leq 2\Delta + \rho$. This contradicts our initial assumption that $e(p) > 2\Delta + \rho$.

In addition, since $e(0) \approx_\Delta e(0) + e(0)$, $|e(0)| \leq \Delta$. $\square$

We now generalize the error bound on $\mathcal{D}^+_{n,s}$ to $\mathcal{D}_{n,s}$.

Lemma 7 *If g is Δ-approximately linear on $\mathcal{D}_{n,s}$, and h and e are defined as before, and if $|e(\frac{n}{s})| = \rho$, then $\forall x \in \mathcal{D}_{n,s}, |e(x)| \leq 2\Delta + \rho$.*

Proof. Observe that if the error is upper bounded by σ in $[0, \frac{n}{s}]$ then $|e(x)| \leq (\sigma + \Delta)/2$ whenever $0 \leq x \leq \frac{n}{2s}$, so that $e(2x) \leq \sigma$. Also, if $|e(x)| \leq \mu$ then $|e(-x)| \leq \mu + 2\Delta$ since $e(0) \leq \Delta$. By Lemma 6, $e(x) \leq 2\Delta + \rho$ for all $x \in \mathcal{D}^+_{n,s}$. We will bound the error in $\mathcal{D}^-_{n,s}$ first by $3\Delta + \rho$ and then by $2\Delta + \rho$. From the above observations, we have $e(x) \leq 4\Delta + \rho$ for $x \in \mathcal{D}^-_{n,s}$, $e(x) \leq (3\Delta + \rho)/2$ for $x \in [0, \frac{n}{2s}]$ and $e(x) \leq (5\Delta + \rho)/2$ for $x \in [-\frac{n}{2s}, 0]$.

Assume that $\exists x \in \mathcal{D}^-_{n,s}$ such that $e(x) = 3\Delta + \rho + \epsilon > 3\Delta + \rho$. Let p be the minimal such point. Then $p > -\frac{n}{2s}$, otherwise the error at $2p$ would exceed $3\Delta + \rho$. Let t be the point with the highest error in $\mathcal{D}^+_{n,s}$ (maximal such one if there is a tie). We consider the possible locations for t to bound $e(t)$: (1) if $t \leq \frac{n}{2s}$, then to ensure that $e(2t) \leq e(t)$, $e(t) \leq \Delta$; (2) if $\frac{n}{2s} < t \leq |p|$, then $t + p \in [-\frac{n}{2s}, 0]$, therefore, to satisfy the bound above on $e(t + p)$, $e(t) \leq \Delta/2 - \epsilon \leq \Delta$; (3) if $t > |p|$, then $t + p \in (0, \frac{n}{2s}]$, therefore to satisfy the bound above, $e(t) \leq -\Delta/2 - \epsilon \leq \Delta$.

Regardless of where t lies, $e(t) \leq \Delta \leq \Delta + \rho$, hence the error in $\mathcal{D}^+_{n,s}$ is bounded by $\Delta + \rho$. However, $e(\frac{n}{s} + p) \geq 3\Delta + \rho + \epsilon - \Delta > 2\Delta + \rho$. Since $\frac{n}{s} + p \in \mathcal{D}^+_{n,s}$, this is a contradiction with the bound that we established before. Therefore, there cannot be a point in $\mathcal{D}^-_{n,s}$ with error $> 3\Delta$. A symmetric argument can be used to bound negative error.

Now we reduce the error bound to $2\Delta + \rho$. Assume that p is the minimal point in $\mathcal{D}^-_{n,s}$ with error $\geq 2\Delta + \rho$. The proof is similar to the previous stage, using the tighter bound $e(x) \leq 2\Delta + \rho/2$ for $x \in [-\frac{n}{2s}, 0]$. Cases (1) and (3) stay the same; for case (2) we have: $e(t+p) \leq -\epsilon \leq \Delta$. Therefore, the error cannot exceed $\Delta + \rho$ in $\mathcal{D}^+_{n,s}$. But $e(\frac{n}{s} + p) \geq 2\Delta + \epsilon + \rho - \Delta$, which is a contradiction. $\square$

Theorem 8 *If g is Δ-approximately linear on $\mathcal{D}_{n,s}$, then there is a linear h such that $|h - g| \leq 2\Delta$.*

Proof. Set $h(\frac{n}{s}) = g(\frac{n}{s})$ in Lemma 7. This uniquely defines h with $\rho = 0$. □

The intuition that drives us to set $h(\frac{n}{s}) = g(\frac{n}{s})$ in the proof of Theorem 8 is as follows. Consider the following function: $g(\frac{n}{s}) = (\frac{n}{s} + \frac{[(n-1)/3]}{s})\Delta$ ($[x]$ denotes integer part of number). It is easy to see that $g(x+y) \approx_\Delta g(x) + g(y)$. If we set $h(\frac{1}{s}) = g(\frac{1}{s})$, then we obtain $h(\frac{n}{s}) = \frac{n}{s}\Delta$. But $|g - h|$ is a growing function of n and so there is no way to bound the error at all points. An example showing that our proof is tight is given in the full version.

2.2. Approximate Multilinearity

In this section we focus our attention on *multilinear functions*. A multivariate function is multilinear if it is linear in any one input when all the other inputs are fixed. A multilinear function of k variables is called a k-linear function. An example of a *bilinear* function is multiplication, and bilinearity property can be stated concisely as $f(x_1 + x_1', x_2 + x_2') = f(x_1, x_2) + f(x_1', x_2) + f(x_1, x_2') + f(x_1', x_2')$. Note that distributive property is a special case of multilinearity.

A natural extension of this class of functions is the class of *Δ-approximately k-linear* functions, which are functions that are Δ-approximately linear in each input. For instance, for $k = 2$, a function g is Δ-approximately bilinear if $\forall x_1, x_1', x_2, x_2' \in \mathcal{D}, g(x_1 + x_1', x_2) \approx_\Delta g(x_1, x_2) + g(x_1', x_2)$ and $g(x_1, x_2 + x_2') \approx_\Delta g(x_1, x_2) + g(x_1, x_2')$.

Now we generalize Lemma 7 to Δ-approximately k-linear functions. Let g be a Δ-approximately k-linear function and h be the multilinear function uniquely defined by the condition $h(\frac{n}{s}, \ldots, \frac{n}{s}) = g(\frac{n}{s}, \ldots, \frac{n}{s})$. Let $e \equiv h - g$. e is a Δ-approximately k-linear function.

Since g takes k inputs from $\mathcal{D}_{n,s}$, if we consider each input to g as a coordinate, the set of all possible k-tuples of inputs of g form a $(2n+1) \times \cdots \times (2n+1)$ cube of dimension k. We show that for any point $(x_1, \ldots, x_k)$ in this cube, $|e(x_1, \ldots, x_k)|$ is bounded.

Theorem 9 *If function g is Δ-approximately k-linear, then there exists a k-linear h such that $|h - g| \le 2k\Delta$.*

Proof. With h defined as above, $e(\frac{n}{s}, \ldots, \frac{n}{s}) = 0$. First, we argue about points that have one coordinate that is different from $\frac{n}{s}$. Fix $k-1$ of the inputs to be $\frac{n}{s}$ (hard-wire into g) and vary one (say x_i). This operation transforms g from a Δ-approximately k-linear function of $x_1, \ldots, x_k$ to a Δ-approximately linear function of x_i. By Lemma 7, this function cannot have an error of more than 2Δ in $\mathcal{D}_{n,s}$. Therefore, $|e(\frac{n}{s}, \ldots, \frac{n}{s}, x_i, \frac{n}{s}, \ldots, \frac{n}{s})| \le 2\Delta$, if $|x_i| < \frac{n}{s}$. Next we consider points which have two coordinates that are different from $\frac{n}{s}$. Consider without loss of generality an input $a, b, \frac{n}{s}, \ldots, \frac{n}{s}$. By the result we just argued, we know that $e(\frac{n}{s}, b, \frac{n}{s}, \ldots, \frac{n}{s}) \le 2\Delta$. By fixing inputs 2 through k to be $b, \frac{n}{s}, \ldots, \frac{n}{s}$ and varying the first input, by Lemma 7, we have $|e(a, b, \frac{n}{s}, \ldots, \frac{n}{s})| \le 4\Delta$ for any $a \in \mathcal{D}_{n,s}$. Thus, via symmetric arguments, we can bound the error by 4Δ if any two inputs are different from $\frac{n}{s}$. Continuing this way, it can be shown that for all inputs, the error is $\le 2k\Delta$. □

The following theorem shows that the error can be reduced to $(1+\mu)\Delta$ for any constant $\mu > 0$ by imposing the multilinearity condition on a larger domain $\mathcal{D}'$ and fitting the multilinear function h on $\mathcal{D}$, where $|\mathcal{D}'|/|\mathcal{D}| = \lceil 2k/\mu \rceil$, i.e., the precision is multiplied by $2k/\mu$. Note that doubling the domain size only involves adding one more bit to the representation of a domain element.

Theorem 10 *For any $\mu > 0$, if g is Δ-approximately k-linear in $\mathcal{D}^k_{2kn/\mu,s}$, then there exists a k-linear function h such that $|h - g|_{\mathcal{D}^k_{n,s}} \le (1+\mu)\Delta$.*

Proof. By Theorem 9, g is $2k\Delta$-close to a k-linear h on $\mathcal{D}_{2kn/\mu,s}$. For any $x = x_1, \ldots, x_k$, we fix all coordinates except x_i and argue in the i-th coordinate as below.

For any $\mathcal{D}_{m,s}$, first we show that if $|e(x)|_{\mathcal{D}_{m,s}} \le \rho$ then $|e(x)|_{\mathcal{D}_{m/2,s}} \le (\rho + \Delta)/2$. To observe this, note that if $x \in \mathcal{D}_{m/2,s}$, then $2x \in \mathcal{D}_{m,s}$. Therefore the function should satisfy $e(x) + e(x) \approx_\Delta e(2x)$, which implies that $|e(x)| \le (\rho + \Delta)/2$. Thus in general, the maximum error in $\mathcal{D}_{m/2^i,s}$ is $\le \rho/2^i + \Delta(1 - 1/2^i)$. Since the error in $\mathcal{D}_{2kn/\mu,s}$ is $\le 2k\Delta$, the error in $\mathcal{D}_{n,s}$ is $\le (1+\mu)\Delta$ by our choice of parameters. □

If g is symmetric, one can always assume that h is symmetric. This observation will be used for polynomials. The following illustrates the scheme.

Corollary 11 *In Theorem 9, if g is symmetric then one can always replace h with a symmetric h'.*

Proof. If construction in the proof of the theorem yields a symmetric h, then $h' = h$. Assume that it does not. Obtain a symmetric h' from h as in [Gaj90]. For a permutation σ of $\{1, 2, \ldots, k\}$, $\vec{x} = x_1, \ldots, x_k$, let $\sigma(\vec{x})$ denote $x_{\sigma(1)}, \ldots, x_{\sigma(k)}$. Then, set $h'(\vec{x}) = 1/k! \sum_\sigma h(\sigma(\vec{x}))$. From symmetry of g, we have $|h' - g| \le |1/k! \sum_\sigma h(\sigma(\vec{x})) - g(\sigma(\vec{x}))| \le 1/k! \sum_\sigma |h(\sigma(\vec{x})) - g(\sigma(\vec{x}))| \le 2k\Delta$ as before. □

3. Polynomials

To test programs purportedly computing polynomials, it is tempting to (1) interpolate the polynomial from randomly chosen points, and then (2) verify that the program is approximately equal to the interpolated polynomial for a large fraction of the inputs. Since a degree d k-variate polynomial can have $(d+1)^k$ terms, this leads to exponential running times. Furthermore, it is not obvious how error bounds that are *independent* of the domain size can be obtained.

Our test uses the same "evenly spaced" interpolation identity as that in [RS92]: for all $x, t \in \mathcal{D}$, $\sum_{i=0}^{d+1}(-1)^{d+1-i}\binom{d+1}{i}f(x+it) = 0$. This identity is computed by the method of successive differences which never explicitly interpolates the polynomial computed by the program, thus giving a particularly simple and efficient ($O(d^2)$ operations) test.

We can show that the interpolation identity is approximately robust by modifying the robustness theorem in [RS92]. Our proof of stability of the interpolation identity (Section 3.2), however, uses a characterization of polynomials in terms of multilinear functions that previously has not been applied to program checking. This in turn allows us to use our results on the stability of multilinearity (Section 2.2) and other ideas from stability theory. See full version for extension of these techniques to multivariate polynomials.

3.1. Preliminaries

In this section, we present the basic definitions and theorems that we will use. Define $\nabla_t f(x) \stackrel{\text{def}}{=} f(x+t) - f(x)$ be the standard *forward difference operator*. Let $\nabla_t^d f(x) \stackrel{\text{def}}{=} \overbrace{\nabla_t \cdots \nabla_t}^{d} f(x) = \sum_{k=0}^{d}(-1)^{d-k}\binom{d}{k}f(x+kt)$ and $\nabla_{t_1,t_2}f(x) \stackrel{\text{def}}{=} \nabla_{t_1}\nabla_{t_2}f(x)$.

Facts 12 *(1) ∇ is linear, (2) ∇ is commutative, (3)* $\nabla_{t_1+t_2} - \nabla_{t_1} - \nabla_{t_2} = \nabla_{t_1,t_2} = \nabla_{t_2,t_1}$.

Let $x^{[k]}$ denote $\overbrace{x,\ldots,x}^{k}$. For any k-ary symmetric f, let $f^*(x) = f(x^{[k]})$ denote its diagonal restriction. We use the following characterization of polynomials [MO34, Dji69].

Fact 13 *The following are equivalent: (1)* $\forall x \in \mathcal{D}, f(x) = \sum_{k=0}^{d} a_k x^k$, *(2)* $\forall x,t \in \mathcal{D}, \nabla_t^{d+1}f(x) = 0$, *(3) there exist symmetric k-linear functions F_k, $0 \le k \le d$ such that* $\forall x \in \mathcal{D}, f(x) = \sum_{k=0}^{d} F_k^*(x)$.

The following definitions are motivated by the notions of using evenly and unevenly spaced points in interpolation.

Definitions 14 *A function g is called* strongly Δ-approximately degree d polynomial *if $\forall x, t_1, \ldots, t_{d+1}$ such that $-n < x + t_1 + \cdots + t_{d+1} < n$, $|\nabla_{t_1,\ldots,t_{d+1}}g(x)| \le \Delta$. A function g is called* weakly Δ-approximately degree d polynomial *if $\forall x, t$ such that $-n < x + t(d+1) < n$, $|\nabla_t^{d+1}g(x)| \le \Delta$.*

3.2. Stability for Polynomials

First, we prove that if a function is strongly Δ-approximately polynomial then there is a polynomial that $(2^{d \lg d}\Delta, 0)$-approximates it. Next, we show that if a function is weakly approximately polynomial on a domain, then there is a coarser subdomain on which the function is strongly approximately polynomial. By using Theorem 10, we can bring the above error arbitrarily close to Δ by assuming the hypothesis on a large enough domain. In order to pass programs that err by at most Δ', we need to set $\Delta \ge (d+1) \cdot 2^d \Delta'$.

STRONGLY APPROXIMATE CASE. One must be careful in defining polynomial h that is close to g. Attempts to define h based on the values of g at some $d+1$ points do not seem to work. We proceed by modifying techniques in [AB83, Gaj90], using the following fact:

Fact 15 *If f is a symmetric k-linear function, then $\nabla_{t_1,\ldots,t_d}f^*(x) = k!f(t_1,\ldots,t_k)$ if $k = d$ and 0 if $k < d$.*

Theorem 16 *If g is strongly Δ-approximately degree d polynomial on $\mathcal{D}_{n(d+2),s}$, then there is a degree d polynomial h_d such that $|g - h_d|_{\mathcal{D}_{n,s}} \le O(2^{d \lg d})\Delta$.*

Proof. Restricted to $\mathcal{D}_{n,s}$, we have $\forall x, t_1, \ldots, t_{d+1} \in \mathcal{D}_{n,s}$, $|\nabla_{t_1,\ldots,t_{d+1}}g(x)| \le \Delta$, because $x + t_1 + \cdots + t_{d+1} \in \mathcal{D}_{(d+2)n,s}$ and the original hypothesis guarantees this.

We induct on the degree. When $d = 0$, we have $|\nabla_t g(x)| \le \Delta$ i.e., $|\nabla_t g(0)| = |g(t) - g(0)| \le \Delta$ for all $t \in \mathcal{D}_{n,s}$. Setting $h_0 = g(0)$, a constant, we are done.

Suppose the lemma holds when the degree is $< d+1$. Now, we have $|\nabla_{t_1,\ldots,t_{d+1}}g(x)| \le \Delta$. Using Fact 12, we have $|\nabla_{t_1+t_1',t_2,\ldots,t_d}g(x) - \nabla_{t_1,t_2,\ldots,t_d}g(x) - \nabla_{t_1',t_2,\ldots,t_d}g(x)| = |\nabla_{t_1,t_1',\ldots,t_d}g(x)| \le \Delta$ by our hypothesis. By symmetry of the difference operator, we have in effect an Δ-approximate symmetric d-linear function on $\mathcal{D}_{n,s}$, say $G(t_1,\ldots,t_d) = \nabla_{t_1,\ldots,t_d}g(0)$. Corollary 11 on multilinearity guarantees a symmetric d-linear H with $|G - H| \le 2d\Delta$. Let $H_d(x_1,\ldots,x_d) = H(x_1,\ldots,x_d)/d!$. Let $g'(x) = g(x) - H_d^*(x)$ for $x \in \mathcal{D}_{n,s}$. Now, using our modified hypothesis on $\mathcal{D}_{n,s}$, our construction of H_d, and Fact 15, we have $\forall x, t_1, \ldots, t_d$, $|\nabla_{t_1,\ldots,t_d}g'(x)| = |\nabla_{t_1,\ldots,t_d}(g(x) - H_d^*(x))| \le |\nabla_{t_1,\ldots,t_d}g(x) - \nabla_{t_1,\ldots,t_d}g(0)| + |\nabla_{t_1,\ldots,t_d}g(0) - \nabla_{t_1,\ldots,t_d}H_d^*(x))| = |\nabla_{h_1,\ldots,h_d,x}g(0)| + |G(t_1,\ldots,t_d) - d!H_d(t_1,\ldots,t_d)| \le \Delta + |G(t_1,\ldots,t_d) - H(t_1,\ldots,t_d)| \le (2d+1)\Delta$.

Now we apply the induction hypothesis. g' satisfies the hypothesis above for d and $\Delta' = (2d+1)\Delta$ and we are guaranteed the existence of a degree $d-1$ polynomial h_{d-1} such that $|g' - h_{d-1}| \le e_{d-1}\Delta'$. Set $h_d = h_{d-1} + H_d^*$. By Fact 13 about the characterization of polynomials, h_d is a degree d polynomial. Now, $e_d = |g - h_d| = |g - h_{d-1} - H_d^*| = |g' - h_{d-1}| \le e_{d-1}\Delta' = e_{d-1}(2d+1)\Delta$.

Unwinding the recurrence, the final error $|g - h_d| = \Delta \prod_{i=1}^{d}(2i+1)$. $\square$

WEAKLY APPROXIMATE CASE. We need the following useful fact [Dji69] which helps us to go from equally spaced points to unequally spaced points:

Fact 17 *For any* $\lambda_1, \ldots, \lambda_d \in \{0,1\}^d$, *if* $t'_{\lambda_1,\ldots,\lambda_d} = -\sum_{i=1}^{d} \lambda_i t_i / i$ *and* $t''_{\lambda_1,\ldots,\lambda_d} = \sum_{i=1}^{d} \lambda_i t_i$ *then* $\nabla_{t_1,\ldots,t_d} f(x) = \sum_{\lambda_1,\ldots,\lambda_d \in \{0,1\}^d} (-1)^{\lambda_1+\cdots+\lambda_d} \nabla^d_{t'_{\lambda_1,\ldots,\lambda_d}} f(x + t''_{\lambda_1,\ldots,\lambda_d})$.

Using this fact, we obtain the following theorem. Let $\mu(d) = \mathrm{lcm}\{1, 2, \ldots, d\}$.

Theorem 18 *If g is weakly $(\Delta/2^{d+1})$-approximately degree d polynomial on $\mathcal{D}_{n(d+1),s\mu(d+1)}$, then g is strongly Δ-approximately degree d polynomial on $\mathcal{D}_{n,s}$.*

Proof. For $t_1, \ldots, t_{d+1} \in \mathcal{D}_{n,s}$, and for any $\lambda_1, \ldots, \lambda_{d+1} \in \{0,1\}^{d+1}$, we have by our choice of parameters that $t'_{\lambda_1,\ldots,\lambda_{d+1}}, t''_{\lambda_1,\ldots,\lambda_{d+1}} \in \mathcal{D}_{n(d+1),s\mu(d+1)}$. Therefore, $|\nabla_{t_1,\ldots,t_{d+1}} g(x)| \leq \sum_{\lambda_1,\ldots,\lambda_{d+1} \in \{0,1\}^{d+1}} |\nabla^{d+1}_{t'_{\lambda_1,\ldots,\lambda_{d+1}}} f(x + t''_{\lambda_1,\ldots,\lambda_{d+1}})| \leq \Delta$. $\square$

4. Functional Equations

Extending the technique in Theorem 8 to addition theorems $f(x+y) = G[f(x), f(y)]$ does not appear to be straightforward, since G can be an arbitrary function. In order to prove approximate robustness (Section 4.3), several related properties of G are required. Proving that G satisfies each individual one is tedious; however, the notion of *modulus of continuity* from approximation theory gives a general approach to this problem. We show that bounds on the modulus of continuity imply bounds on all of the quantities of G that we require. The stability of G is shown by a careful inductive technique based on a canonical generation of the elements in $\mathcal{D}_{n,s}$ (Section 4.2). The scope of our techniques is not limited to addition theorems; e.g., in the full version they are used to show that Jensen's equation is approximately robust and stable.

4.1. Preliminaries

For addition theorems, we can assume that G is algebraic and a symmetric function (the latter is true in general under some technical assumptions as in [Rub94]). We need a notion of "smoothness" of G. The following notions are well-known in approximation theory [Lor66, Tim63].

Definitions 19 (Moduli of Continuity) *The* modulus of continuity *of the function $f(\cdot)$ is the following function of $\delta \in [0,\infty)$* : $\omega(f;\delta) = \sup_{\substack{|x_1-x_2|\leq\delta \\ x_1,x_2\in\mathcal{D}}} \{|f(x_1) - f(x_2)|\}$.
The modulus of continuity *of the function $f(\cdot,\cdot)$ is the following function of $\delta_x, \delta_y \in [0,\infty)^2$* : $\omega(f;\delta_x,\delta_y) = \sup_{\substack{|x_1-x_2|\leq\delta_x, |y_1-y_2|\leq\delta_y \\ x_1,y_1,x_2,y_2\in\mathcal{D}}} \{|f(x_1,y_1) - f(x_2,y_2)|\}$.
The partial moduli of continuity *of the function $f(\cdot,\cdot)$ are the following functions of $\delta \in [0,\infty)$* :
$\omega(f;\delta,0) = \sup_{y\in[a,b]} \sup_{\substack{|x_1-x_2|\leq\delta \\ x_1,x_2\in[a,b]}} \{|f(x_1,y) - f(x_2,y)|\}$ *and*
$\omega(f;0,\delta) = \sup_{x\in[a,b]} \sup_{\substack{|y_1-y_2|\leq\delta \\ y_1,y_2\in[a,b]}} \{|f(x,y_1) - f(x,y_2)|\}$.

The following facts are easily proved:

Facts 20 *(1) $\omega(f;\delta) \geq 0$,*
(2) $\omega(f;\delta) \leq \omega(f;\delta')$ if $\delta \leq \delta'$,
(3) $\omega(f+g;\delta) \leq \omega(f;\delta) + \omega(g;\delta)$,
(4) If f' exists and is bounded in $\mathcal{D}$, then $\omega(f;\delta) \leq \delta|f'|_{\mathcal{D}}$,
(5) $\omega(f;\delta_x+\rho_x,\delta_y+\rho_y) \leq \omega(f;\delta_x,\delta_y) + \omega(f;\rho_x,\rho_y)$,
(6) $\omega(f;\delta,\delta) \leq \omega(f;0,\delta) + \omega(f;\delta,0)$,
(7) If $f(\cdot,\cdot)$ is symmetric, then $\omega(f;\delta,\delta) \leq 2\omega(f;\delta,0)$,
(8) If $\widehat{f} = |f'_x|_{\mathcal{D}}$ where f'_x is the partial derivative of f with respect to x, then $\omega(f;\delta,0) \leq \delta\widehat{f}$.

We need a notion of an "inverse" of G. If $G[x,y] = z$, denote $G_1^{-1}[z,y] = x, G_2^{-1}[x,z] = y$. Since G is symmetric, they are identical and we denote $G^{-1}[z,y] = x$. When G^{-1} is well-defined, it can be used to handle elements in $\mathcal{D}^-$.

AN EXAMPLE. Wherever necessary, we will illustrate our scheme using the following functional equation $f(x+y) = \dfrac{f(x)f(y)}{f(x)+f(y)}$, i.e., $G[x,y] = xy/(x+y)$. The solution to this functional equation is $f(x) = C/x$ for some constant C. The following fact [Tit47] is useful in locating the maxima of analytic functions.

Fact 21 (Maximum Modulus Principle) *If f is analytic in a compact region $\mathcal{D}$, then f attains extremum only on the boundary of $\mathcal{D}$.*

Over a bounded rectangle $\mathcal{D} = [L,U]^2$, where $0 < L \leq U$, G is analytic and hence by Facts 21 and 20, $\omega(G;\delta,0) \leq \sup_{y\in[L,U]} \dfrac{\delta y^2}{(L+y)^2} = \dfrac{\delta U^2}{(L+U)^2}$.

4.2. Stability for Functional Equations

Let the domain be $\mathcal{D}^+_{n,s}$ with $n = 2^k$. We construct the function h that is close to g. Let $e(x) = |h(x) - g(x)|$, i.e., $h(x) \approx_{e(x)} g(x)$). For simplicity, let $H_1(x) \stackrel{\mathrm{def}}{=} G[x,x]$ and let $\omega(H_1;\delta) = c\delta$. We consider the case when $c < 1$. The other two cases ($c > 1$ and $c = 1$) are presented in the full version. Call $\frac{x}{s}$ even (resp. odd) if x is even (resp. odd).

Lemma 22 $e(2x) \leq ce(x) + \Delta$.

Proof. $e(2x) = |g(2x) - h(2x)| \leq \Delta + |G[g(x), g(x)] - G[h(x), h(x)]|$. But, by Fact 20, we get $|H_1(g(x)) - H_1(h(x))| = |H_1(h(x) + e(x)) - H_1(h(x))| \leq \omega(H_1; e(x)) = ce(x)$ □

Now, we set $h(\frac{1}{s}) = g(\frac{1}{s})$. This determines h for all values in $\mathcal{D}$ by the fact that h satisfies the functional equation. We have to argue about relationship between $e(x + \frac{1}{s})$ and $e(x)$. For simplicity, let $H_2(x) \stackrel{\text{def}}{=} G[x, g(\frac{1}{s})]$ and $\omega(H_2; \delta) = |H_2'(\cdot, g(\frac{1}{s}))| \leq d\delta$ for some constant d. Now,

Lemma 23 $e(x + \frac{1}{s}) \leq de(x) + \Delta$.

Proof. $e(x + \frac{1}{s}) = |g(x + \frac{1}{s}) - h(x + \frac{1}{s})| \leq \Delta + |G[g(x), g(\frac{1}{s})] - G[h(x), h(\frac{1}{s})]|$. But, $|G[g(x), h(\frac{1}{s})] - G[h(x), h(\frac{1}{s})] = |H_2[g(x)] - H_2[h(x)]| = |H_2[h(x) + e(x)] - H_2[h(x)]| \leq \omega(H_2; e(x)) \leq de(x)$. □

We will show a scheme to bound $e(x)$ for all x when $d < 1$. This scheme can be thought of as an enumeration strategy, where at each step of the process, certain constraint equations have to be satisfied. First, we will show a canonical listing of elements in $\mathcal{D}^+_{n,s}$.

Lemma 24 *The elements in $\mathcal{D}^+_{n,s}$ can be generated in a canonical order starting from $\frac{1}{s}$ and using operations $2x$ and $x + \frac{1}{s}$ to obtain new elements on any of the elements already generated.*

Proof. Construct a binary tree T_k in the following manner. The nodes of the tree are labeled with elements from $\mathcal{D}^+_{n,s}$. The root of the tree is labeled $\frac{1}{s}$. If x is the label of a node, then $2x$ is the label of its left child (if $2x$ is not already in the tree). and $x + \frac{1}{s}$ is the label of its right child (if $x + \frac{1}{s}$ is not already in the tree). Using induction, we can prove that T_k contains all elements of $\mathcal{D}^+_{n,s}$. □

Corollary 25 *In T_k, if x is even (except root), then x is a left child; if x is odd, then x is a right child.*

A canonical way of listing elements in $\mathcal{D}^+_{n,s}$ arises from a preorder traversal of T_k. This ordering is used in our inductive argument.

Lemma 26 *For all $x \in \mathcal{D}^+_{n,s}$, if x is even, then $e(x) \leq \frac{1+c}{1-c}\Delta$ and if x is odd, then $e(x) \leq \frac{2}{1-c}\Delta$.*

Proof. We will prove this by induction on the preorder enumeration of the tree given by Lemma 24. Let x be the next element to be enumerated. By preorder listing, its parent has already been enumerated and hence, its error is known. If $x = 2y$ is even, from Corollary 25, it is a left child, and hence generated by a $2y$ operation. Hence, $e(y) \leq \frac{2}{1-c}\Delta$ by induction hypothesis. This together with Lemma 22 yields $e(x) \leq ce(y) + \Delta \leq c\frac{2}{1-c}\Delta + \Delta \leq \frac{1+c}{1-c}\Delta$, preserving the induction hypothesis. If $x = y + 1$ is odd, from Corollary 25, it is a right child, and hence generated by a $y + 1$ operation. But, y is even, $e(y) \leq \frac{1+c}{1-c}\Delta$ by induction hypothesis. This together with Lemma 23 and $d \leq 1$ yields $e(x) \leq de(y) + \Delta \leq e(y) + \Delta \leq \frac{1+c}{1-c}\Delta + \Delta \leq \frac{2}{1-c}\Delta$, preserving the induction hypothesis. □

This yields the following theorem:

Theorem 27 *For all $x \in \mathcal{D}^+_{n,s}$, $e(x) \leq \frac{2}{1-c}\Delta$.*

With our example, we have $H_1(x) = G[x, x] = x/2$ and so $c = 1/2$. Also, $H_2(x) = G[x, g(\frac{1}{s})]$ from which $H_2'(x) = g(\frac{1}{s})^2/(x + g(\frac{1}{s}))^2 \leq 1$ as $0 < L \leq x, g(\frac{1}{s})$. Thus, $d \leq 1$. By Theorem 27, we have $e(x) \leq 4\Delta$ for all $x \in \mathcal{D}^+_{n,s}$. To argue about $\mathcal{D}^-_{n,s}$, we pick a "pivot" point in $\mathcal{D}_{n,s}$ (0 for simplicity). Now, we have $h(-x) = G^{-1}[h(x), h(0)]$. Therefore, as before, we have $e(-x) \leq \omega(G^{-1}; \frac{2c}{1-c})$.

When $d \geq 1$, the error can no longer be bounded. In this case, we have $c \leq 1 < d$. Let $e = cd$. We can see from the structure of T_k that the maximum error can occur at $\frac{2^k-1}{s}$. By simple induction on the depth of the tree, the error is given by $e(\frac{2^k-1}{s}) \leq \sum_{i=0}^{k-2}(d^{i+1} + d^i)c^i\Delta = (d+1)\sum_{i=0}^{k-2} e^i\Delta = (d+1)\frac{e^{k-1}-1}{e-1}\Delta$. If $e < 1$, we obtain a constant error bound of $e(x) \leq (d+1)\frac{1}{1-e}\Delta$ by geometric summation. Otherwise, we obtain $e(x) = O(e^{\lg n})$.

4.3. Approximate Robustness for Functional Equations

As in [GLR+91, RS92], we test the program on the larger *test domain* $\mathcal{D}_{2p,s}$ and make conclusions about a smaller *safe domain* $\mathcal{D}_{n,s}$. The relationship between p and n will be determined later. The domain has to be such that G is analytic in it. Therefore, we consider the case when f is bounded on $\mathcal{D}_{2p,s}$, i.e., $0 < L \leq f(x) \leq U$. Let $\mathcal{G}$ be the family of functions that satisfy the following conditions:

(a) $\Pr_{x \in \mathcal{D}_{2p,s}}[f(x) \geq L] \geq 1 - \delta$,

(b) $\Pr_{x \in \mathcal{D}_{2p,s}}[f(x) \leq U] \geq 1 - \delta$,

(c) $\Pr_{x,y \in \mathcal{D}_{2p,s}}[G[f(x), f(y)] \geq L] \geq 1 - \delta$, and

(d) $\Pr_{x,y \in \mathcal{D}_{2p,s}}[G[f(x), f(y)] \leq U] \geq 1 - \delta$.

Note that membership in $\mathcal{G}$ is easy to determine by sampling. We can define a distribution $\mathcal{E}$ such that if P satisfies the functional equation on $\mathcal{E}$ with probability $\geq 1 - \delta/4$, then P also satisfies the following four properties:

(1) $\Pr_{x,y \in \mathcal{D}_{p,s}}[P(x + y) \approx_\Delta G[P(x), P(y)]] \geq 1 - \delta$,

(2) $\Pr_{x,y \in \mathcal{D}_{p,s}}[P(x) \approx_\Delta G[P(x - y), P(y)]] \geq 1 - \delta$,

(3) $\Pr_{x,y \in \mathcal{D}_{p,s}}[P(x) \approx_\Delta G[P(y), P(x - y)]] \geq 1 - \delta$, and

(4) $\Pr_{x \in \mathcal{D}_{n,s}, y \in \mathcal{D}_{p,s}}[P(x) \approx_\Delta G[P(x - y), P(y)]] \geq 1 - \delta$.

$\mathcal{E}$ is defined by flipping a fair four-sided die and on outcome

$i \in \{1,2,3,4\}$ checking the (i)-th property. We show that addition theorems on distribution $\mathcal{E}$ are $(\delta, 2\delta, \Delta, (9\widehat{G}^2 + 5\widehat{G})\Delta, \Delta, \mathcal{D}_{2p,s}, \mathcal{D}_{n,s})$-approximately robust in $\mathcal{G}$ (see Fact 20 for definition of $\widehat{G}$).

Define for $x \in \mathcal{D}_{p,s}$, $g(x) = \underset{y \in \mathcal{D}_{p,s}}{\text{median}}\{G[P(x-y), P(y)]\}$. By Markov's inequality, g's definition, and the properties of P, it is easy to show the following:

Theorem 28 $\Pr_{x \in \mathcal{D}_{n,s}}[g(x) \approx_\Delta P(x)] > 1 - 2\delta$.

For simplicity of notation, let P_x denote $P(x)$ for any $x \in \mathcal{D}_{p,s}$ and $G_{x,y}$ denote $G[P(x), P(y)]$ for any $x, y \in \mathcal{D}_{p,s}$. Since G is fixed, we will drop G from the modulus of continuity. Let $\gamma = n/2p$.

Fact 29 *For* $x \in \mathcal{D}_{2n,s}$, $\Pr_{y \in \mathcal{D}_{p,s}}[x + y \in \mathcal{D}_{p,s}] \geq 1 - \gamma$.

For $x \in \mathcal{D}_{2n,s}$, noting that $x - z$, y, z are all random, we have $\Pr_{y,z \in \mathcal{D}_{p,s}} [G_{x-y,y} \approx_{\omega(\Delta,0)} G[G_{x-z,z-y}, P_y] = G[P_{x-z}, G_{z-y,y}] \approx_{\omega(0,\Delta)} G_{x-z,z}] > 1 - 12\delta - 2\gamma$. The error in the first step (due to computation of P_{x-y}) is $\omega(\Delta, 0)$ and the equation holds with probability $\geq 1 - \delta$ by hypothesis (3). The bounds on $G_{x-z,z-y}$ also hold with probability $\geq 1 - 2\delta$ by hypotheses (c), (d) and so the error is just $\omega(\Delta, 0)$. There is a loss of 2γ probability for $x - y$, $x - z$ to be in $\mathcal{D}_{p,s}$ by Fact 29. The next line is just rewriting. In a similar manner, the final equation holds with probability $\geq 1 - \delta$ by hypothesis (2) and the error bound is $\omega(0, \Delta)$ The bounds on random points $P_y, P_z, P_{x-z}, P_{z-y}$ hold with probability $\geq 1 - 8\delta$ by hypotheses (a), (b) on P to make the error $\omega(0, \Delta)$. Hence, the total error is $\omega(\Delta, 0) + \omega(0, \Delta) = 2\omega(\Delta, 0)$ by Fact 20 and the equality holds with probability $\geq 1 - 12\delta - 2\gamma$. We can then show:

Theorem 30 *For all* $x, y \in \mathcal{D}_{n,s}$, $g(x + y) \approx_{\Delta''} G[g(x), g(y)]$ *with probability* $\geq 1 - 56\delta - 9\gamma$, *where* $\Delta'' = (9\widehat{G}^2 + 5\widehat{G})\Delta$.

If $\delta < 1/56, p > 11n$, we have $1 - 56\delta - 9\gamma > 0$ and so the above lemma is true with probability 1. In the case of our example function, we already calculated $\widehat{G} = U^2/(L+U)^2$. Hence, $\Delta'' = \Delta \left(\frac{9U^4}{(L+U)^4} + \frac{5U^2}{(L+U)^2} \right)$.

4.4. Reductions Between Functional Equations

This section explores the idea of using reductions among functions (as in [BK89, ABC+93]) to obtain approximate self-testers for new functions. Consider any pair of functions f_1, f_2 that are interreducible via functional equations. Suppose we have an approximate self-tester for f_1 and let there exist continuous functions F, F^{-1} such that $f_2(x) = F[f_1(x)]$ and $f_1(x) = F^{-1}(f_2(x))$. Given a program P_2 computing f_2, construct program P_1 computing f_1 via F^{-1}. We can then self-test P_1. Suppose P_1 is Δ-close to f_1 on a large portion of the domain. Then for every x for which $P_1(x)$ is Δ-close to $f_1(x)$, we bound the deviation of $P_2(x)$ from $f_2(x)$ by $\Delta' = F[f_1(x) + \Delta] - f_2(x)$. Then $\Delta' = F[f_1(x) + \Delta] - F[f_1(x)] \leq \omega(F; \Delta)$. If we can bound the right hand side by a constant (at least for a portion of the domain), we can bound the maximum deviation Δ' of P_2 from f_2. This idea can be used to give simple and alternative approximate self-testers for functions like $\sin x, \cos x, \sinh x, \cosh x$ which can be reduced to e^x.

ACKNOWLEDGEMENTS. We would like to thank Janos Aczel (U. of Waterloo), Peter Borwein (Simon Fraser U.), Gian Luigi Forti (U. of Milan), D. Sivakumar (SUNY, Buffalo), Madhu Sudan (IBM, Yorktown), and Nick Trefethen (Cornell U.), for their suggestions and pointers.

References

[Acz66] J. Aczel. *Lectures on Functional Equations and their Applications*. Academic Press, 1966.

[AB83] M. Albert and J.A. Baker. Functions with bounded n-th differences. *Ann. Polonici Mathematici*, 43:93 – 103, 1983.

[ABC+93] S. Ar, M. Blum, B. Codenotti, and P. Gemmell. Checking approximate computations over the reals. *Proc. 25th STOC*, pp. 786–795, 1993.

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

[AS92] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *Proc. 33rd FOCS*, pp. 2–13, 1992.

[BFL91] L. Babai, L. Fortnow, and C. Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, pp. 3–40, 1991.

[BK89] M. Blum and S. Kannan. Designing programs that check their work. *Proc. 21st STOC*, pp. 86–97, 1989.

[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.

[BW94] M. Blum and H. Wasserman. Program result-checking: A theory of testing meets a test of theory. *Proc. 35th FOCS*, pp. 382–392, 1994.

[BW94] M. Blum and H. Wasserman. Reflections on the Pentium division bug. *Proc. 8th Intl. Software Quality Week*, 1994.

[CR92] E. Castillo and M.R. Ruiz-Cobo. *Functional Equations and Modeling in Science and Engineering*. Marcel Dekker Inc., 1992.

[Cho84] P.W. Cholewa. The stability problem for a generalized Cauchy type functional equation. *Rev. Roumaine Math. Pures Appl.*, 29:457–460, 1984.

[Dji69] D.Z. Djokovic. A representation theorem for $(X_1 - 1)(X_2 - 1)\ldots(X_n - 1)$ and its applications. *Ann. Polonici Mathematici*, 22:189 – 198, 1969.

[For80] G.L. Forti. An existence and stability theorem for a class of functional equations. *Stochastica*, 4:23–30, 1980.

[For95] G.L. Forti. Hyers–Ulam stability of functional equations in several variables. *Aeq. Mathematicae*, 50:143–190, 1995.

[FS89] G.L. Forti and J. Schwaiger. Stability of homomorphisms and completeness. *C.R. Math. Rep. Acad. Sci. Canada*, 11:215–220, 1989.

[Gaj90] Z. Gajda. Local stability of the functional equation characterizing polynomial functions. *Ann. Polonici Mathemtici*, 52:119 – 137, 1990.

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

[Hye41] D.H. Hyers. On the stability of the linear functional equation. *Proc. Nat. Acad. Sci. U.S*, 27:222–224, 1941.

[Lip91] R. Lipton. New directions in testing. *Proc. DIMACS Workshop on Distr. Comp. and Cryptography*, pp. 191–202, 1991.

[Lor66] G. G. Lorentz. *Approximation of Functions*. Holt, Rinehart and Winston, 1966.

[Lun91] C. Lund. *The Power of Interaction*. Ph.D. Thesis, U. of Chicago, 1991.

[MO34] S. Mazur and W. Orlicz. Grundlegende Eigenschaften der Polynomischen Operationen. *Erste Mitteilung, Studia Math.*, 5:50 – 68, 1934.

[Rub94] R. Rubinfeld. Robust functional equations with applications to self-testing/correcting. *Proc. 35th FOCS*, pp. 288–299, 1994.

[RS92] R. Rubinfeld and M. Sudan. Testing polynomial functions efficiently and over rational domains. *Proc. 3rd SODA*, pp. 23–43, 1992.

[RS96] R. Rubinfeld and M. Sudan. Robust characterizations of polynomials and their applications to program testing. *SIAM J. of Computing*, 25(2):252–271, 1996.

[Sko83] F. Skof. Sull'approssimazione delle applicazioni localmente δ-additive. *Atti della Accademia delle Scienze di Torino*, 117:377–389, 1983.

[Sud91] M. Sudan. *Personal Communication*, 1991.

[Tim63] A.F. Timan. *Theory of Approximation of Functions of a Real Variable*. Pergamon Press, 1963.

[Tit47] E.C. Titchmarsh. *The Theory of Functions*. Oxford U. Press, 1947.

A. Constructing Approximate Self-Correctors

We illustrate how to build approximate self-correctors for functional equations. Suppose P (Δ, ϵ)-approximates f for $\epsilon < 1/8$ and $f(x+y) = G[f(x), f(y)]$. Then the self-corrector P_{sc} at input x is constructed as follows. To obtain a confidence of β, we choose random points $y_1, y_2, \ldots, y_N (N = O(\ln 1/\beta))$ and let $P_{\text{sc}}(x)$ to be the median of $G[P(x-y_1), P(y_1)], \ldots, G[P(x-y_N), P(y_N)]$. By assumption, both the calls to P are within Δ of f with probability $> 3/4$. In this case, the value of $G[P(x-y), P(y)]$ is $\Delta' = 2\Delta\widehat{G}$ away from $f(x)$ (see Section 4.1 for $\widehat{G}$). Using Chernoff bounds, we can see that at least half of the values $G[P(x-y), P(y)]$ are at most Δ' away from $f(x)$. Thus, their median $P_{\text{sc}}(x)$ is also at most Δ' away from $f(x)$.

For degree d polynomials, a similar self-corrector works with $\Delta' = O((d+1)2^d\Delta)$. In order to pass good programs, this is the best Δ' possible using the evenly spaced interpolation equation. Using interpolation equations that do not use evenly spaced points seem to require Δ' that is dependent on the size of the domain.

B. Constructing Approximate Self-Testers

The following is a self-tester for an addition theorem $f(x+y) = G[f(x), f(y)]$ computing the function family $\mathcal{F}$ over $\mathcal{D}_{n,s}$. The following uses notations from Section 4.1. To obtain a confidence of β, we choose random points $x_1, y_1, \ldots, x_N, y_N (N = O(\max\{1/\delta, 48\} \ln 1/\beta))$ and verify the assumptions on program P in the beginning of Section 4.3. If P passes the test, then using Chernoff bounds, approximate robustness, and stability of the property, we are guaranteed that P approximates some function in $\mathcal{F}$. We next perform the equality test to ensure that P approximates the given $f \in \mathcal{F}$. Assume that $f(\frac{1}{s})$ when $c < 1$ (resp. $f(\frac{n}{s})$ when $c > 1$) is given. Modifying the proofs in Section 4.2 , we can show that if $P_{\text{sc}}(\frac{1}{s}) \approx f(\frac{1}{s})$ when $c < 1$ ($P_{\text{sc}}(\frac{n}{s}) \approx f(\frac{n}{s})$ when $c > 1$), the error between P_{sc} and f can be bounded by a constant on the rest of $\mathcal{D}_{n,s}$. Since P_{sc} approximates P, the correctness of the self-tester follows.

For polynomials, we verify the assumptions (see full version) on program P required for approximate robustness. Using the approximate robustness and stability of the evenly spaced interpolation equation, P is guaranteed to approximate some degree d polynomial h. To perform the equality test that determines if P approximates the correct polynomial f, we assume that the tester is given the correct value of the polynomial f at $\ell = (d+1)/\epsilon$ evenly spaced points $x_1 = -\frac{n}{s}, \ldots, x_\ell = \frac{n}{s} \in \mathcal{D}_{n,s}$. Using the self-corrector P_{sc} from Appendix A, we have $|P_{\text{sc}} - h| \leq \Delta' = (d+1)2^d 2^{d \lg d}\Delta$. The equality tester now tests that for all x_i, $|f(x_i) - P_{\text{sc}}(x_i)| \leq (d+1)2^d\Delta$. Call an input x *bad* if $|f(x) - h(x)| > \Delta'' = \Delta' + (d+1)2^d\Delta$. If x is bad then $|f(x) - P_{\text{sc}}(x)| > (d+1)2^d\Delta$. If x is a sample point, and x is bad, then the test would have failed. If the test passes, then any *bad interval* (sequence of consecutive bad points) in the domain can be of length at most $(2n+1)/\ell$, because any longer interval would contain at least one sample point. The two points immediately preceding and following the bad interval satisfy $|f(x) - h(x)| \leq \Delta''$. This implies that there must be a local maximum of $f - h$ (a degree d polynomial) inside the bad interval. Since there are only d extrema of $f - h$, the total number of bad points is $\leq d(2n+1)/\ell$. Thus, on $1 - \epsilon$ fraction of $\mathcal{D}_{n,s}$, P_{sc}'s error is $\leq \Delta' + \Delta''$. These arguments can be extended to the multivariate case.