Coin Flipping Cannot Shorten Arithmetic Computations

Stasys Jukna

Abstract. We use elementary arguments to show that randomization cannot spare even one single ring operation to compute real multivariate polynomials.

1. INTRODUCTION. Let R be any of the rings $\mathbb{Z}$ (integers), $\mathbb{Q}$ (rational numbers), or $\mathbb{R}$ (real numbers). We are interested in how many ring operations $+, -, \times$ must we apply to compute a given polynomial f over R, when starting from input variables and ring elements. Note that we only want to compute the polynomial f as a *function*, not to produce it as a formal expression.

Every such *procedure* F for computing f (known also as a *straight-line program* or *arithmetic circuit*) is just a sequence $f_1, \ldots, f_l$ of polynomials, where each f_i is obtained by applying one of the ring operations $+, -, \times$ to some two previous (not necessarily distinct) polynomials in $R \cup \{x_1, \ldots, x_n, f_1, \ldots, f_{i-1}\}$; elements of R are also polynomials (of zero degree). The *length* of such a procedure is the number l of polynomials in the sequence, and the function $F : R^n \to R$ *computed* by the procedure is the function computed by the last polynomial f_l.

Examples. A naive procedure to compute a univariate polynomial $f(x) = \sum_{k=0}^{d} a_k x^k$ has length $3d - 1$: we need $d - 1$ additions and $2d$ multiplications (to compute powers and multiply them with coefficients). But we can do better: the Horner procedure $f_1 = a_{d-1} + a_d \times x$, $f_2 = a_{d-2} + f_1 \times x, \ldots, f_d = a_0 + f_{d-1} \times x$ also computes f, and has length only $2d$; for ease of description, we here perform two operations in one step. Some polynomials can be computed by even exponentially shorter procedures. For example, the univariate polynomial $f(x) = \sum_{k=0}^{d} \binom{d}{k} x^k$ of degree $d = 2^m$ can be computed by the procedure $f_1, \ldots, f_l$ of length only $l = m + 1$. Namely, we can take $f_1 = x + 1$ and $f_{i+1} = f_i \times f_i$; then $f_{m+1}(x) = (x + 1)^d$ which, by the binomial theorem, is the same function as $f(x)$.

A natural question is: can *randomization* shorten the procedures for computing polynomials? Our goal is to give a *negative* answer, in a very strong sense: flipping a coin during the computation will not spare even one single ring operation!

A *random* procedure is a (deterministic) procedure which, besides the string $x = (x_1, \ldots, x_n)$ of input variables, can use an additional string $\boldsymbol{r} = (\boldsymbol{r}_1, \ldots, \boldsymbol{r}_m)$ of *random* variables taking their values in the underlying ring R; the actual probability distributions of these random variables will be irrelevant in our argument. What such a procedure then computes is a *random polynomial*, an object which attracted the attention of many generations of mathematicians.

A random procedure F *computes* a given n-variate polynomial $f(x)$ with a *positive success probability* if there is an $\varepsilon = \varepsilon(n) > 0$ such that, for every input $a \in R^n$, $F(a, \boldsymbol{r}) = f(a)$ holds with probability at least ε. That is, we allow an arbitrarily small nonzero success probability but, on every input $a \in R^n$, the correct *value* $f(a)$ must be computed with this (or larger) probability.

doi.org/10.1080/00029890.2019.1565883
MSC: Primary 68W20, Secondary 12Y05

Theorem. *If a polynomial f can be computed by a random procedure of length l with a positive success probability, then f can be also computed by a deterministic procedure of the same length l.*

The message of this theorem is: randomization *cannot* save even one single ring operation. This—the optimality of the result, together with the extreme simplicity of its proof—is our main contribution. That randomization cannot help "much" when computing polynomials was already shown by Cucker et al. [1]. They consider more general procedures that, besides the ring operations $+, -, \times$, can also use the division and signum operations, and show that also then random procedures can be simulated by at most quadratically longer deterministic procedures. They obtained this result via a cute combination of deep known results in statistical learning theory, algebraic geometry, and quantifier elimination over the reals. In contrast, we show that if the random procedures do not use the signum operation, then an elementary argument, based on the following two simple lemmas, gives an optimal derandomization.

Lemma 1. *Suppose that an n-variate polynomial f can be computed by a random procedure of length l with a positive success probability $\varepsilon > 0$. Then, for every finite nonempty set $A \subset R^n$, there is a deterministic procedure F of length l such that $F(a) = f(a)$ holds for more than $\frac{\varepsilon}{2}|A|$ inputs $a \in A$.*

Proof. Take a random procedure of length l computing f with a success probability $\varepsilon > 0$. Let $A \subset R^n$ be a finite set, and take $m := \lceil 4\varepsilon^{-2} \ln |A| \rceil$ independent copies of this random procedure. For an input $a \in A$, let $X_{a,i}$ be the Bernoulli 0/1-random variable with $X_{a,i} = 1$ if and only if the ith copy outputs the correct value $f(a)$ on input a. Since $\Pr\left[X_{a,i} = 1\right] \geq \varepsilon$ holds for every i, the expected value μ of the sum $X_a = X_{a,1} + \cdots + X_{a,m}$ is $\mu \geq \varepsilon m$. So, for $\delta := \varepsilon/2$, we have $\Pr[X_a \leq \delta m] \leq \Pr[X_a \leq \mu - \delta m]$. By Chernoff's bound (see, for example, [2, Theorem 1.1]), the latter probability is at most $p = \mathrm{e}^{-2\delta^2 m} = \mathrm{e}^{-\varepsilon^2 m/2} \leq |A|^{-2}$. By the union bound, the probability that $X_a \leq \delta m$ will hold for *at least one* input $a \in A$ is at most $p \cdot |A|$, which is strictly smaller than 1. Thus, the probability that, for *every* input $a \in A$, more than δm of the m copies of our random procedure will output the correct value $f(a)$ is nonzero.

There must therefore exist m realizations $F_1, \ldots, F_m$ of our random procedure (deterministic procedures of length l) such that, on *every* input $a \in A$, more that δm of them will output the correct value $f(a)$. By double counting, at least one of the procedures $F_1, \ldots, F_m$ must then output correct values $f(a)$ on more than $\delta|A| = \frac{\varepsilon}{2}|A|$ inputs $a \in A$. ∎

For finite fields $\mathbb{F}$, and $S = \mathbb{F}$, the following extension of the fundamental theorem of algebra to *multivariate* polynomials dates back to 1922, and was originally proved by Ore [3]. Various extensions to arbitrary fields and arbitrary finite subsets $S \subseteq \mathbb{F}$ were then found by other authors. We will use that due to Schwartz [4].

Lemma 2 (Ore–Schwartz). *Let $\mathbb{F}$ be a field, $f(x)$ a nonzero n-variate polynomial of degree $d \leq |\mathbb{F}|$ over $\mathbb{F}$, and $S \subseteq \mathbb{F}$ a finite subset of $|S| \geq d$ field elements. Then $|\{a \in S^n : f(a) = 0\}| \leq d|S|^{n-1}$.*

Proof. We induct on the number n of variables. The statement is true for $n = 1$ since then the number of roots of f cannot exceed its degree. For the induction step, write the polynomial f as $f(x) = \sum_{i=0}^{d} f_i \cdot x_n^i$, where each f_i is some polynomial in the first $n-1$ variables. Let $t = \max\{i : f_i \neq 0\}$. So, $f_t(x_1, \ldots, x_{n-1})$ is a nonzero polynomial of degree at most $d - t$.

By the induction hypothesis, f_t can have at most $(d-t)|S|^{n-2}$ roots $a \in S^{n-1}$ and, hence, there can be at most $(d-t)|S|^{n-1}$ points (a, b) in $S^{n-1} \times S$ with $f(a, b) = 0$ *and* $f_t(a) = 0$. For every point $a \in S^{n-1}$ with $f_t(a) \neq 0$, the polynomial $f(a, x_n)$ is a nonzero univariate polynomial of degree t in one variable x_n, and can have at most t roots $b \in S$. So, there can be at most $t|S|^{n-1}$ points (a, b) in $S^{n-1} \times S$ with $f(a, b) = 0$ *and* $f_t(a) \neq 0$. Overall, the number of points $(a, b) \in S^n$ with $f(a, b) = 0$ is at most $(d-t)|S|^{n-1} + t|S|^{n-1} = d|S|^{n-1}$. ∎

2. PROOF OF THE THEOREM. Let $f(x)$ be an n-variate polynomial over R, and suppose that f can be computed by a random procedure of length l with a success probability $\varepsilon > 0$. Set $d := \max\{\deg(f), 2^l\}$, and take a subset $S \subseteq R$ of size $|S| \geq 2d/\varepsilon$; this is the only place where we need $|R|$ to be unbounded. By Lemma 1 (applied with $A = S^n$), there must be a subset $X \subseteq S^n$ of $|X| > \frac{\varepsilon}{2}|S|^n$ input vectors and a *deterministic* procedure F of length l such that $F(a) = f(a)$ holds for all $a \in X$.

The procedure F is a sequence of only l applications of ring operations to previously obtained polynomials. Since we start from polynomials of degree 0 and 1 (scalars and single variables), and since the degree after each application can only be doubled, the polynomial $F(x)$ has degree at most 2^l. So, the degree of the polynomial $g(x) := f(x) - F(x)$ cannot be larger than d.

Were g a nonzero polynomial, then Lemma 2 (applied with $\mathbb{F} = \mathbb{R}$) would require the set X to have cardinality $|X|$ at most $d|S|^{n-1}$. But then we would have $\frac{\varepsilon}{2}|S|^n < |X| \leq d|S|^{n-1}$ and, hence, also $|S| < 2d/\varepsilon$, which contradicts our choice of S. So, $g(x) = f(x) - F(x)$ must be the zero polynomial, meaning that $F(a) = f(a)$ must hold for *all* inputs $a \in R^n$, as desired. ∎

Remark. The result also holds when the domain R is $\mathbb{Q}$ or $\mathbb{R}$, and all four field operations $+, -, \times, \div$ can be used, that is, when procedures can also divide. What such procedures then compute are rational functions $f(x) = p(x)/q(x)$, where p and q are polynomials. In this case, we can argue as above with $d := r + 2^l$, where r is the maximum degree of p and q. The deterministic procedure F obtained in the proof of the theorem then also computes some rational function $F(x) = P(x)/Q(x)$, where the polynomials P and Q have degrees at most 2^l. So, the degree of the polynomial $g(x) := p(x) \cdot Q(x) - q(x) \cdot P(x)$ cannot exceed d, and the same argument shows that g must be the zero polynomial.

ACKNOWLEDGMENT. Research supported by the DFG grant JU 3105/1-1 (German Research Foundation).

REFERENCES

[1] Cucker, F., Karpinski, M., Koiran, P., Lickteig, T., Werther, K. (1995). On real Turing machines that toss coins. In: Leighton, F. T., Borodin, A., eds. *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*. New York: Association for Computing Machinery, pp. 335–342.

[2] Dubhashi, D., Panconesi, A. (2009). *Concentration of Measure for the Analysis of Randomized Algorithms*. New York: Cambridge Univ. Press.

[3] Ore, Ø. (1922). Über höhere Kongruenzen. *Norsk Mat. Forenings Skrifter Ser. I*, 7: 15 pages.

[4] Schwartz, J. T. (1980). Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*. 27(4): 701–717.

Institute of Computer Science, Goethe University Frankfurt, Frankfurt am Main, Germany
Affiliated with Institute of Data Science and Digital Technologies, Vilnius University, Vilnius, Lithuania
stjukna@gmail.com