On real Turing machines that toss coins

Felipe Cucker*
Universitat Pompeu Fabra
Balmes 132, Barcelona 08008
SPAIN
e-mail: cucker@upf.es

Marek Karpinski†
Institut für Informatik
Universität Bonn
GERMANY
e-mail: marek@cs.uni-bonn.de

Pascal Koiran‡
LIP, Ecole Normale Supérieure de Lyon
46 allée d'Italie, 69364 Lyon Cedex 07
FRANCE
e-mail: koiran@lip.ens-lyon.fr

Thomas Lickteig§
Institut für Informatik
Universität Bonn
GERMANY
e-mail: lickteig@cs.uni-bonn.de

Kai Werther
Institut für Informatik
Universität Bonn
GERMANY
e-mail: kai@cs.uni-bonn.de

May 7, 1996

Abstract

In this paper we consider real counterparts of classical probabilistic complexity classes in the framework of real Turing machines as introduced by Blum, Shub, and Smale [2]. We give an extension of the well-known "BPP $\subseteq$ P/poly" result from discrete complexity theory to a very general setting in the real number model. This result holds for real inputs, real outputs, and random elements drawn from an arbitrary probability distribution over $\mathbb{R}^m$. Then we turn to the study of Boolean parts, that is, classes of languages of zero-one vectors accepted by real machines. In particular we show that the classes BPP, PP, PH, and PSPACE are not enlarged by allowing the use of real constants and arithmetic at unit cost provided we restrict branching to equality tests.

*Partially supported by DGICyT PB 920498, the ESPRIT BRA Program of the EC under contracts no. 7141 and 8556, projects ALCOM II and NeuroCOLT.

†Partially supported by Leibniz Center for Research in Computer Science, by the DFG grant KA 673/4-1 ,and by the ESPRIT BR Grants 7097 and EC-US 030.

‡This work was done while this author was visiting DIMACS (Rutgers University), and was supported by the NeuroCOLT project and by an INRIA fellowship.

§Supported by DFG Grant Li-405/2-1

1 Introduction

This paper deals with probabilistic complexity classes in the real number model of computation. We consider both uniform and non-uniform models. The classical non-uniform models are straight-line programs, algebraic circuits and algebraic decision trees. Our uniform model is the real Turing machine introduced by Blum, Shub and Smale in the foundational paper [2].

In section 3 we present a generalization of the well-known "BPP $\subseteq$ P/poly" result from discrete complexity theory to the real number model. This problem implicitly goes back at least to Heintz and Schnorr [17]. After giving a deterministic counterpart to Schwarz's probabilistic algorithm for testing polynomials [30], these authors write that "This statement sounds much like Adleman's (1978) observation that every problem which is decidable in random polynomial time has polynomial network size. However in our situation Adleman's argument is not applicable since $W(d, n, v)$ is not finite[1]". The main tool for our proof is a result of probability theory based on the notion of Vapnik-Chervonenkis (VC) dimension (see section 2). The VC dimension was invented exactly to remedy the lack of finiteness pointed out by Heintz and Schnorr. This tool was originally developed to study some statistical aspects of pattern recognition, or learning theory as one would say nowadays (see e.g. [36]). It turned out to have important applications in other areas as well, especially in computational geometry (see e.g. the survey [6]).

A similar theorem was established by Meyer auf der Heide [25] for the algebraic decision tree model. Our result has a much wider range of applicability since it is only based on a VC dimension hypothesis and therefore, it is distribution independent. In the algebraic decision tree model, and for random elements drawn from $\{0,1\}$ we recover Meyer auf der Heide's $O(nT^2)$ bound for deterministically simulating a probabilistic algorithm that runs in time T on $\mathbb{R}^n$.

In section 3 we work with non-uniform models since this can only strengthen our results. Of course the classical proof that BPP $\subseteq$ P/poly also applies to non-uniform models, the corresponding result being that probabilistic boolean circuits are not more powerful than deterministic boolean circuits (up to a polynomial increase in size). Some applications of our simulation are presented in section 4. In particular, we prove by quite different methods a theorem which is very similar to Heintz and Schnorr's [17]. Lower bounds are discussed in section 5.

A recent issue in the theory initiated by Blum, Shub and Smale [2] is the comparison of the computational power of real and classical (discrete) Turing machines. This comparison is performed by feeding the real machines with binary inputs, i.e. with finite strings of zeros and ones. Depending on the resources allowed to the real machines, this defines a class of binary sets whose position in the taxonomy of Boolean complexity classes makes the desired comparison possible. In other words, for any real complexity class $\mathcal{C}$ we want to characterize its Boolean part, which is defined to be

$$\mathrm{BP}(\mathcal{C}) = \{S \cap \{0,1\}^* : S \in \mathcal{C}\}.$$

Typical examples of resource bounds in the real setting define versions of the well known classes P, NP, Σ_k, PH and PAR, the latter denoting parallel polynomial time (and being thus equivalent to PSPACE in the Boolean case). These resource bounds can apply to several kinds of real Turing

[1] $W(d, n, v)$ is the set of inputs considered by Heintz and Schnorr. It plays the same role as $\{0,1\}^n$ for boolean circuits or $\{0,1\}^*$ for Turing machines.

machines according to whether they can multiply or divide at unit cost, whether they cannot multiply or divide at all (additive machines) or whether they can do it in a restricted way (the weak model introduced in [21]). Finally, a last differentiation can be done according to whether these machines branch over order tests or only over equalities (in this case we will say that the machine is equational or order-free).

Results concerning Boolean parts have been established for all these settings. Thus, in [22] and [9] Boolean parts of several complexity classes for additive machines are characterized. Also, in [21] and [12] Boolean parts of some complexity classes in the weak model are characterized whereas for complexity classes defined with unrestricted multiplications and divisions, this task is carried out in [10]. We begin section 7 with a more detailed exposition of these previous results.

In the last sections of this paper, we pursue the study of Boolean parts for equational real Turing machines. In order to do so, we introduce in section 6 probabilistic real Turing machines along with the complexity classes they define, i.e. real versions of the classes ZPP, R, BPP and PP. The kind of randomization we use here is "coin tossing" i.e. random elements are chosen from $\{0,1\}$ with equal probability. For this computational model it is shown that randomization does not help in the sense that $\mathrm{P}_{\mathbb{R}} = \mathrm{BPP}_{\mathbb{R}}$. This is not necessarily true for the order-free case and for this case the study of Boolean parts is pursued. The main theorem is established in section 7 and characterizes the Boolean parts of several complexity classes.

2 Uniform convergence and the Vapnik-Chervonenkis dimension

In this section we recall a few well-known facts about the Vapnik-Chervonenkis dimension.

Definition 1 Let $\mathcal{F}$ be a class of indicator ($\{0,1\}$-valued) functions on a domain X. We say that $\mathcal{F}$ *shatters* a set $A \subseteq X$ if for every subset $E \subseteq A$, there exists some function $f_E \in \mathcal{F}$ satisfying:

- $f_E(x) = 0$ for every $x \in A \backslash E$;
- $f_E(x) = 1$ for every $x \in E$.

The *VC dimension* of $\mathcal{F}$ is the cardinality of the largest set that is shattered by $\mathcal{F}$.

Goldberg and Jerrum gave bounds on the VC dimension of classes of subsets of $\mathbb{R}^n$ defined by quantifier-free polynomial formulas over the reals ([15], Theorem 2.2).

Theorem 1 (Goldberg-Jerrum) *Let $\Phi_{k,n}(x,y)$ be a boolean formula containing s distinct atomic predicates, where each predicate is a polynomial equality or inequality over $n+k$ variables (representing $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^k$, respectively) of degree at most D. For any $y \in \mathbb{R}^k$, let $F_y \subseteq \mathbb{R}^n$ be the set of instances x such that $\Phi_{k,n}(x,y)$ is satisfied. The family $\{F_y; y \in \mathbb{R}^k\}$ has VC dimension at most $2k\log(8eDs)$.*

Since the behavior of arithmetic circuits can be characterized by such formulas, they went on to give bounds on the VC dimension of classes of sets recognized by these circuits ([15], Theorem 2.3). See Definition 3 in the next section if you are not familiar with this computation model.

Corollary 1 *Let $\{C_{k,n};\ k,n \in \mathbb{N}\}$ be a family of arithmetic circuits of size $s = s(k,n)$ with inputs in $\mathbb{R}^{n+k}$. For any $y \in \mathbb{R}^k$, let $F_y \subseteq \mathbb{R}^n$ be the set of instances x such that (x,y) is accepted by C. The family $\{F_y; y \in \mathbb{R}^k\}$ has VC dimension at most $O(sk)$.*

This result was actually stated in the algebraic computation tree model, but in this paper we are mostly interested in circuits.

By quantifier elimination, one can also deal with formulas in the first-order theory of the reals ([15], Corollary 2.4).

Corollary 2 *Assume now that formula $\Phi_{k,n}$ in Theorem 1 is a first-order formula in the theory of the reals; assume further that the number of bound variables is polynomial in k and n, that the number of quantifier alternations is uniformly bounded and that the atomic predicates are bounded in number and degree by an exponential function of k and n. Then the VC dimension of $\{F_y; y \in \mathbb{R}^k\}$ is polynomial in k and n.*

The proof of Theorem 1 relies on a deep result from real algebraic geometry due to Warren [37]. Consider a family $f_1, \ldots, f_m$ of polynomials in $\mathbb{R}[X_1, \ldots, X_k]$. An m-tuple $(\sigma_1, \ldots, \sigma_m)$ of sign conditions (that is, σ_i is one of > 0, $= 0$, or > 0, for all $i \leq k$) is said to be *satisfied* by the family when the set of $x \in \mathbb{R}^n$ such that

$$f_1(x)\sigma_1 \ \& \ \ldots \ \& \ f_m(x)\sigma_m$$

is non-empty. An immediate upper bound for the number of satisfiable tuples of sign conditions is 3^m. Warren's result shows that the crucial parameter is not m but the number k of variables.

Theorem 2 (Warren) *Let $f_1, \ldots, f_m$ be polynomials in $\mathbb{R}[X_1, \ldots, X_k]$ whose degrees are bounded by d. The number of satisfiable m-tuples of sign conditions is bounded by $(8md/k)^k$.*

In his paper, Warren gave only a bound on the of non-zero sign assignements. The generalization to arbitrary sign assignements is easy, see for instance [15].

In fact, a slightly weaker bound is used by Meyer auf der Heide in [25]. There, the bound is easily derived from a well known result of Milnor [27] and Thom [34] that bounds the number of connected components of polynomial systems in $\mathbb{R}^n$. Note also that a related result in Warren's paper plays a crucial role in [20]. It seems thus that these geometric bounds are at the core of the problem.

The notion of VC dimension can be generalized to real-valued functions as follows.

Definition 2 Let $\mathcal{F}$ be a class of real valued functions on a domain X. We say that $\mathcal{F}$ *P-shatters* a set $A \subseteq X$ if there exists a function $s : A \to \mathbb{R}$ such that for every subset $E \subseteq A$, there exists some function $f_E \in \mathcal{F}$ satisfying:

- $f_E(x) < s(x)$ for every $x \in A \backslash E$;
- $f_E(x) \geq s(x)$ for every $x \in E$.

The *pseudo-dimension* of $\mathcal{F}$, denoted $P(\mathcal{F})$, is the cardinality of the largest set that is P-shattered by $\mathcal{F}$.

For indicator functions, the pseudo-dimension reduces to the VC dimension.

Remark 1 The pseudo-dimension of $\mathcal{F}$ is the VC dimension of the subgraph of $\mathcal{F}$ (the subgraph of $\mathcal{F}$ is the class of sets of the form $\{(x,y) \in X \times \mathbb{R};\ y \leq f(x)\}$ for some $f \in \mathcal{F}$).

The key property of classes of functions of finite pseudo-dimension is that they satisfy a uniform law of large numbers, as shown by the following result ([16], Corollary 2).

Theorem 3 *Let $\mathcal{F}$ be a class of functions from a set X into a bounded range $[0,M]$. Assume that $\mathcal{F}$ has finite pseudo-dimension d. Then, for any distribution P over X, any $\epsilon, \delta > 0$ and any*

$$k \geq \frac{64M^2}{\epsilon^2}(2d \ln \frac{16eM}{\epsilon} + \ln \frac{8}{\delta})$$

the following statement holds[2]: if k points $x_1, \ldots, x_k$ are independently drawn from P, then with probability at least $1-\delta$,

$$\left| \frac{1}{k} \sum_{i=1}^{k} f(x_i) - E_{x \in X} f(x) \right| \leq \epsilon$$

for every $f \in \mathcal{F}$ (E denotes the mathematical expectation).

3 Simulations of probabilistic circuits by deterministic circuits

For the sake of generality, we shall deal here with circuits that can draw random elements from an arbitrary probability distributions P on $\mathbb{R}^m$ (from section 6 onward, we will require P to be the uniform distribution on $\{0,1\}^m$). Thus we consider circuits with $n+m$ real inputs and the last m of them are to be interpreted as "random inputs". Let us define more formally our computational model.

To do so, we will consider the sign function

$$\mathrm{sg} : \mathbb{R} \to \{0,1\}$$

defined by $\mathrm{sg}(x) = 1$ if $x \geq 0$ and 0 otherwise. Thus, the sign function is the characteristic function of the set of positive elements in $\mathbb{R}$.

Definition 3 An *arithmetic circuit* C over $\mathbb{R}$ is a directed acyclic graph where each node has indegree 0, 1 or 2. Nodes with indegree 0 are either labeled as input nodes or with elements of $\mathbb{R}$ (we shall call them constant nodes). Nodes with indegree 2 are labeled with the binary operators

[2] $\mathcal{F}$ must satisfy a benign measurability condition called permissibility. See for instance Appendix A.1 in [3] or Appendix C in [28]. In the rest of the paper, we assume implicitly that this condition is satisfied. One can show that it is indeed satisfied in all concrete examples considered.

on $\mathbb{R}$, i.e., $+$, $\times$, $-$ or $/$. Nodes of indegree 1 are sign nodes; nodes of outdegree 0 are called output nodes.

For an arithmetic circuit C, the *size* $s(C)$ of C, is the number of nodes in C. The *depth* $d(C)$ of C, is the length of the longest path from some input node to some output node.

Let n be the number of input nodes of a circuit C. To each node g of C we inductively associate a function f_g from $\mathbb{R}^n$ to $\mathbb{R}$. We shall refer to the function f_C from $\mathbb{R}^n$ to $\mathbb{R}^s$ associated to the s output gates as the function *computed by the circuit*. It is immediate to note that f_C is a piecewise rational function from $\mathbb{R}^n$ to $\mathbb{R}^s$. If it is a characteristic function, i.e., if $s = 1$ and its image is included in $\{0, 1\}$, we shall say that the set $S \subset \mathbb{R}^n$ given by $S = \{x \in \mathbb{R}^n \mid f_C(x) = 1\}$ is *recognized* by C. We will also say that C is a *decision* circuit.

This way of recognizing a set is deterministic. We are now going to allow arithmetic circuits to draw random elements from $\mathbb{R}$.

Definition 4 Let P be a probabilistic distribution on $\mathbb{R}^m$ and let C be an arithmetic decision circuit with $n + m$ input nodes. Let also S be a subset of $\mathbb{R}^n$. We say that C recognizes S with *error probability* bounded by $\varepsilon < \frac{1}{2}$ when for any $x \in \mathbb{R}^n$ the following probability (with respect to random elements $y \in \mathbb{R}^m$)

$$\Pr(f_C(x, y) = 1 \ \& \ x \notin S) + \Pr(f_C(x, y) = 0 \ \& \ x \in S)$$

is bounded by ε. By abuse of language we shall say that the circuit C above is probabilistic, and if the value of ε is not relevant in the context, we shall simply say that C recognizes S.

Our positive results are based on the following "abstract" theorem, which in fact does not depend on the real number model at all.

Theorem 4 *Let P be an arbitrary probability distribution on a set Y; let $f : X \to Z$ and $g : X \times Y \to Z$ be two functions such that*

$$P\{y \in Y;\ g(x, y) = f(x)\} \geq 3/4$$

for every $x \in X$. Assume that the family of functions

$$\mathcal{F} = \{F_{x,z} : Y \to \{0, 1\};\ x \in X, z \in Z\}$$

defined by

$$F_{x,z}(y) = \begin{cases} 1 \text{ if } g(x, y) = z; \\ 0 \text{ if } g(x, y) \neq z \end{cases}$$

has finite VC dimension d. Then, there exists a universal constant C such that for every $k \geq Cd$, there are points $y_1, \ldots, y_k \in Y$ such that for every $x \in X$, at least 2/3 of the elements $g(x, y_1), \ldots, g(x, y_k)$ are equal to $f(x)$. In other words, $f(x)$ can be computed by taking a majority vote on $g(x, y_1), \ldots, g(x, y_k)$.

Proof. Let us draw k points $y_1, \ldots, y_k$ independently from P. According to Theorem 3, the probability that

$$\left| \frac{1}{k} \sum_{i=1}^{k} F_{x,z}(y_i) - E_{y \in Y} F_{x,z}(y) \right| > \epsilon \tag{1}$$

for at least one $(x, z) \in X \times Z$ is smaller than one for $k = \Omega(\frac{d}{\epsilon^2} \log \frac{1}{\epsilon})$. Hence for this value of k, there must exist at least one realization of $y_1, \ldots, y_k$ such that (1) does not hold for any $(x, z) \in X \times Z$. The result follows from the choice $z = f(x)$ and $\epsilon = 3/4 - 2/3$. □

Note that using Theorem A.2 in ([36], p. 170) instead of Theorem 3 yields the bound $k = O(d \log d)$ instead of $k = O(d)$. This sharper bound also follows implicitly from the proof of Theorem 4.2 in [13].

Corollary 3 *Let C be a probabilistic arithmetic circuit of size s with inputs in $\mathbb{R}^n$, outputs in $\mathbb{R}$ and random elements drawn from an arbitrary probability distribution on $\mathbb{R}^m$. Then C can be simulated by a deterministic arithmetic circuit of size polynomial in n, s and m.*

Proof. Follows immediately from Theorem 4 with $X = \mathbb{R}^n$, $Y = \mathbb{R}^m$, $g : \mathbb{R}^{n+m} \to \{0, 1\}$ the function computed by C and $f : \mathbb{R}^n \to \{0, 1\}$ the characteristic function of the set recognized by C. In this case, the VC dimension of the resulting $\mathcal{F}$ is $O(sn)$ by Corollary 1. □

For circuits with output in $\mathbb{R}^p$ it is well known that the class of real functions $f : \mathbb{R}^n \to \mathbb{R}^p$ computed by deterministic circuits is the class of piecewise rational functions. This fact readily extends to probabilistic circuits with random elements drawn from $\{0, 1\}$ with the uniform distribution. A straightforward consequence of Corollary 3 is that it also holds for arbitrary probability distributions.

Corollary 4 *Let C be a probabilistic arithmetic circuit with inputs in $\mathbb{R}^n$ output in $\mathbb{R}^p$ and random elements drawn from an arbitrary probability on $\mathbb{R}^m$. Then the function $f_C : \mathbb{R}^n \to \mathbb{R}^p$ computed by C is piecewise rational.*

This result also holds in the more powerful algebraic tree model if the circuit size s is replaced by the tree depth T (that was actually the computational model used in Goldberg-Jerrum [15]). Hence we retrieve Meyer auf der Heide's $O(nT^2)$ bound for deterministically simulating probabilistic decision trees [25]. Note that this author only considered the case where P is the uniform distribution over $\{0, 1\}^m$, and his argument cannot be generalized to arbitrary probability distributions. One big advantage of our approach is that it provides distribution-independent bounds. For instance, in the next section we work with the uniform probability distribution over $\{1, \ldots, u\}^n$, where u is some positive integer. Meyer auf der Heide's result can be applied in this case, but the resulting bounds increase with u. In contrast, our bounds are independent of u.

One may ask whether Corollary 3 remains true if the real number model is extended by adding new operations. Corollary 5 shows that the answer to this question is positive if these operations are semi-algebraic (we also give a negative result for the cosine function in the next section). We first recall the following definitions.

Definition 5

- A *basic semi-algebraic set* $S \subseteq \mathbb{R}^k$ is defined by the conjunction of a finite number of constraints of the form $P_i < 0$, $P_i = 0$ or $P_i > 0$, where $P_i : \mathbb{R}^k \to \mathbb{R}$ is a polynomial.
- A *semi-algebraic set* is a finite union of basic semi-algebraic sets.
- A function $s : \mathbb{R}^k \to \mathbb{R}$ is a *semi-algebraic function* if its graph is a semi-algebraic subset of $\mathbb{R}^{k+1}$.

Corollary 5 *Corollary 3 still holds if the real number model is extended by adding any fixed, finite set of semi-algebraic operations to the four arithmetic operations.*

Proof. The condition $F_{x,z}(y) = 1$ can be turned into an existential formula of the first-order theory of the reals by performing the following transformation on every instruction of the straight-line program evaluating this condition: replace an instruction $v := s(u_1, \ldots, u_k)$ by the subformula $\mathcal{R}(v, u_1, \ldots, u_k)$, where v is a new variable and $\mathcal{R}$ is the (non-quantified) formula defining s (s can be a sign gate, one of the four arithmetic operations or one of the new semi-algebraic functions). Then take the conjunction ψ of all these subformulas. The function computed by the straight-line program above is described by the formula $\exists v_1 \ldots \exists v_r \ \psi$ where $v_1, \ldots, v_r$ are the new variables added through this process.

The size of the resulting formula is exponentially (in fact, polynomially) bounded in s, n and m. The number of quantified variables is also polynomial in these parameters. Hence the VC dimension of the resulting $\mathcal{F}$ (defined as in Corollary 3) is polynomial according to Corollary 2. □

This corollary makes it possible to include in our computation model such natural functions as, for instance, the square root function, since the map $x \mapsto \sqrt{|x|}$ is semi-algebraic. It is also possible to obtain good bounds in the case where pfaffian functions (such as, e.g., the exponential function) are allowed, using the recent results of [20]. The field of p-addic numbers is a more exotic domain of computation where some of the results given here for the real number model may apply. To our knowledge, no explicit bound is known on the VC dimension of classes of concepts definable in the first-order theory of the p-addic numbers, but it is known to be finite.

The following result is the counterpart of Theorem 4 for approximation algorithms.

Theorem 5 *Let P be an arbitrary probability distribution on a set Y; let $f : X \to \mathbb{R}$ and $g : X \times Y \to \mathbb{R}$ be two functions such that $|E_{y \in P} g(x, y) - f(x)| \leq \epsilon$ for every $x \in X$. Assume that the family of functions $\mathcal{G} = \{G_x : Y \to \mathbb{R};\ x \in X\}$ defined by $G_x(y) = g(x, y)$ has finite pseudo-dimension d, and that $g(X \times Y) \subseteq [0, M]$ for some $M > 0$. Then, for every $\epsilon, \delta > 0$ and every*

$$k > \frac{64M^2}{\epsilon^2}\left(2d \ln \frac{16eM}{\epsilon} + \ln 8\right) \tag{2}$$

there are points $y_1, \ldots, y_k \in Y$ such that for every $x \in X$,

$$\left| \frac{1}{k} \sum_{i=1}^{k} g(x, y_i) - f(x) \right| \leq 2\epsilon.$$

Proof. Let us draw k points $y_1, \ldots, y_k$ independently from P. According to Theorem 3, the probability that

$$\left| \frac{1}{k} \sum_{i=1}^{k} g(x, y_i) - E_{y \in Y} g(x, y) \right| > \epsilon \tag{3}$$

for at least one $x \in X$ is smaller than one if (2) holds. Hence for such a value of k, there must exist at least one realization of $y_1, \ldots, y_k$ such that (3) does not hold for any $x \in X$. □

By Remark 1, corollaries 3 and 5 clearly hold in this approximation setting as well.

Remark 2 All the results in this section can be also shown for the algebraic tree model of computation. Moreover, in this case they can be strengthened in the following way. The complexity we have considered is the worst–case cost. However, we can provide additional power to algebraic trees by averaging the cost over all the random choices in $\mathbb{R}^m$. The application of theorem 4 is then still possible and allows us to obtain stronger results.

4 Applications

In this section, we give a few applications of the techniques developed in the previous section. We start with a deterministic version of Schwarz's randomized testing algorithm [30].

Theorem 6 *For any integers n, d, d', k, v, we define a set $\mathcal{L} = \mathcal{L}(n, d, d', k, v)$ of straight-line programs as follows. A straight-line program L is in $\mathcal{L}$ if:*

- *It is made of at most v arithmetic operations (no divisions or comparisons are allowed), uses k real parameters (or "constants") and computes a polynomial P_L in n variables.*
- *The degree of P_L in the n variables is at most d.*
- *The degree of P_L in the k parameters is at most d'.*

Set $u = 4nd$. Then there exists a set S of $s = O(v \log v + k \log d')$ points in $\{1, 2, \ldots, u\}^n$ which satisfies the property that for every $L \in \mathcal{L}$, $P_L \equiv 0$ if and only if $P_L(x) = 0$ for every $x \in S$.

By Schwarz's lemma, if $P_L \not\equiv 0$ and x is drawn from the uniform distribution on $\{1, \ldots, u\}^n$,

$$\Pr\{P_L(x) = 0\} \leq dn/u = 1/4.$$

This probabilistic test can be "derandomized" by Theorem 4 and Corollary 3, providing a polynomial bound for s. We explain below how one can obtain the sharper result claimed here by going through the proof of the VC dimension bounds in [15]. The only difference with [15] is that we have to count how many straight-line programs in $\mathcal{L}$ have a different syntactical structure, i.e., cannot be obtained from one another by changing numerical constants. This issue is dealt with in a perhaps more complicated way in ([29], Theorem 4.4), where the program syntax is encoded in a real constant.

Proof of Theorem 6. For $L \in \mathcal{L}$, let $F_L : \mathbb{R}^n \to \{0,1\}$ be defined by: $F_L(x) = 0$ if $P_L(x) = 0$, $F_L(x) = 1$ otherwise. The bound that can be obtained from Theorem 4 is $s = O(N)$, where N is the VC dimension of the family of functions $\mathcal{F} = \{F_L;\ L \in \mathcal{L}\}$. Take any fixed $L_0 \in \mathcal{L}$, and consider the class $\mathcal{L}(L_0)$ of straight-line programs that can be obtained from L_0 by letting the parameters in L_0 take arbitrary values in $\mathbb{R}^k$. It follows from the proof of Theorem 1 (see [15]) that the number of functions induced by $\{F_L; L \in \mathcal{L}(L_0)\}$ on any subset $\{x_1, \ldots, x_N\} \subseteq \mathbb{R}^n$ is bounded by $(8ed'N/k)^k$. Since there are at most $2^{O(v \log v)}$ distinct classes $\mathcal{L}(L_0)$ when L_0 ranges over $\mathcal{L}$, the number of functions induced by $\mathcal{F}$ can be bounded by $2^{O(v \log v)}(8ed'N/k)^k$. It can be shown by elementary calculations that this number can be larger or equal to 2^N only if $N = O(v \log v + k \log d')$. □

Heintz and Schnorr [17] gave the bounds $s = 6(v'+n)(v'+n+1)$ and $u = 2v'(d+1)^2$, where v' counts only the so-called *non-scalar* operations. Their result holds over any algebraically closed field K of characteristic zero. We can show that Theorem 6 also holds in this case since it is possible to recover the VC dimension bounds given in [15] for the real case. This follows from a $O((md/k)^{2k})$ bound on the number of consistent sign assignments for a set of m polynomials of degree d in k variables (in this context, by "sign assignment", we mean a condition of the form "$P = 0$" or "$P \neq 0$"). Indeed, one can write $K = R[i]$ where R is a real closed field. A $O((md/k)^k)$ bound for R follows from the Warren bound and the Tarski transfer principle. Then the claimed bound follows by separating the "real" and "imaginary" parts.

We can also prove a result which is even more similar to Heintz & Schnorr's, namely, $s = O((v'+n)^2 \log v'n)$ and $u = 4nd$. This follows from the fact that if a polynomial of degree d in n variables can be evaluated in v' non-scalar operations its coefficients are polynomials of degree $d(2v'+1)$ in $n + (v'+n)(v'+n+1)$ indeterminates ([17], originally [18]), and then from Theorem 1.

One advantage of our approach to Theorem 6 is that its relation with Schwarz's probabilistic testing algorithm is clearly established. That these two results are closely related was already pointed out in [17], but the nature of this relation remained somewhat mysterious. We have shown here that the existence of a deterministic testing method is not fortuitous: it follows from a general "derandomization" principle.

Corollary 3 can also be useful for proving lower bounds on the size of probabilistic circuits, since it reduces this problem to the (a priori easier) task of proving lower bounds for the size of deterministic circuits.

Theorem 7 *If a function $f : \mathbb{R}^n \to \mathbb{R}$ cannot be computed by a deterministic circuit of size s then it cannot be computed (in the sense of Theorem 4) by a probabilistic circuit of size $C\sqrt{s/n}$, where C is a universal constant.*

This result is a straightforward consequence of Corollary 3. As Corollary 3, it can be formulated in the algebraic decision tree model. Lower bounds for probabilistic decision trees were also obtained by Bürgisser et al. [5]. Their results hold only for the (quite large) class of algebraic probability measures, whereas Theorem 7 holds for arbitrary probability measures.

Finally, randomization does not help much for parallel computing with circuits.

Theorem 8 *If for some $l > 0$ a family of functions $f_n : \mathbb{R}^n \to \mathbb{R}$ cannot be computed by a family of deterministic circuits of polynomial size and $O(\log^l(n))$ depth, it cannot be computed by a family of polynomial-size probabilistic circuits of depth $O(\log^l(n))$ either.*

Proof. Assume by contradiction that f_n can be computed by a circuit of size $s = n^{O(1)}$ and depth $O(\log^l(n))$. The $k = O(d)$ elements $g(x, y_1), \ldots, g(x, y_k)$ in the statement of Theorem 4 can be computed in parallel by a deterministic circuit. Then a majority vote can be taken in depth $O(\log k)$. The resulting circuits are still of polynomial size and $O(\log^l(n))$ depth since $d = O(sn)$ as we have seen in Corollary 3. □

5 Lower bounds on deterministic samples

It would be most interesting to obtain lower bounds on the size of deterministic circuits simulating probabilistic circuits. This is often a very difficult task, to say the least. Here we will be more modest, and will consider only a very specific class of deterministic circuits: those that work exactly as suggested by theorems 4 and 5, i.e., replace the probabilistic sample by the same "deterministic sample" for every input. Thus, we are interested in lower bounds on the size of such a deterministic sample, or results of non-existence of such a sample. In the next theorem we give a result of the first kind and in theorem 10 one of the second kind.

Theorem 9 *Let X and Y be two arbitrary sets and let $g : X \times Y \to \{0, 1\}$ be such such that the family $\mathcal{G} = \{G_x : Y \to \{0, 1\};\ x \in X\}$ defined by $G_x(y) = g(x, y)$ has VC-dimension at least d. Let $S = \{Y_1, \ldots, Y_d\}$ be a shattered set and let P be the uniform distribution on S. Let the "target function" be $f(x) = E_{y \in P} g(x, y)$.*

- *Let us draw k points $y_1, \ldots, y_k$ independently from P. For any $\delta > 0$, the probability that $|\sum_{i=1}^{k} g(x, y_i)/k - f(x)| \geq \epsilon$ is smaller than δ for any $k > \frac{3}{\epsilon^2} \ln \frac{1}{\delta}$.*
- *A "deterministic sample" $S' = \{y_1, \ldots, y_k\} \subseteq S$ satisfies the property that $|\sum_{i=1}^{k} g(x, y_i)/k - f(x)| < \epsilon$ for every $x \in X$ if and only if $k > d(1 - \epsilon)$.*

Proof. Part one follows from the Chernoff bounds since they bound the probability that $|\sum_{i=1}^{k} g(x, y_i)/k - f(x)| \geq \epsilon$ by $e^{-k\epsilon^2/3}$ and this value is smaller than δ for $k > \frac{3}{\epsilon^2} \ln \frac{1}{\delta}$.

For part two, the worst case is given by an x such that $g(x, y) = 1$ if $y \in S'$ and $g(x, y) = 0$ if $y \in S \backslash S'$. In this case $|\sum_{i=1}^{k} g(x, y_i)/k - f(x)| = (d - k)/d$. This error is smaller than ϵ if and only if $k > d(1 - \epsilon)$. Note that the same argument applies if S' is a multiset rather than an ordinary set. □

We now consider a problem involving the cosine function. As shown in [33], one has infinite VC dimension in this case. On an input $w \in \mathbb{R}$, we wish to estimate the area of the subset E_w of $Y = [0, 1]^2$ defined by the condition $y \leq (1 + \cos wx)/2$. By e.g. the Chernoff bounds, this can be done efficiently by drawing points $(x_1, y_1), \ldots, (x_k, y_k)$ from the uniform distribution on $[0, 1]^2$ and counting how many belong to E_w. On the other hand, we shall see that no finite deterministic sample can achieve a worst-case error better than $1/4$ when w ranges over $\mathbb{R}$. Again, this does not mean that there are no other efficient ways of determining the area. Indeed, this area is simply

$$\int_0^1 \frac{1 + \cos wx}{2} dx = \frac{1}{2}(1 + \frac{\sin w}{w}).$$

Our result is based on the following lemma.

Lemma 1 *For every $\epsilon > 0$, every $w_0 > 0$ and every finite set $\{x_1, \ldots, x_k\} \subseteq \mathbb{R}$, there exists $w > w_0$ such that $\cos wx_i > 1 - \epsilon$ for every $i = 1, \ldots, k$.*

Proof. Let $f(w) = (wx_1 \bmod 2\pi, \ldots, wx_k \bmod 2\pi)$, and assume first that the x_i's are rationally independent. As pointed out in [33], it follows from Theorem 3.2 and Lemma 2.7 in [24] that $f(\mathbb{R})$ is dense in $[0, 2\pi]^k$. This implies in particular that for any $\epsilon > 0$, any $\eta < \epsilon/2$, and any $x \in \mathbb{R}^k$ such that $||x|| < \epsilon/2$, some $f(w)$ falls in the ball of center x and radius η. Since the set $f([-w_0, w_0])$ is supported by a finite number of lines, we can choose this ball so that it does not intersect $f([-w_0, w_0])$. A corresponding w satisfies $\cos wx_i > 1 - \epsilon$ for every $i = 1, \ldots, k$, and the lemma is proved since cosine is even.

If the points are not rationally independent, we can assume without loss of generality that $\{x_1, \ldots, x_l\}$ is a basis of the vector space over $\mathbb{Q}$ spanned by $\{x_1, \ldots, x_k\}$. Thus there are rationals α_{ij} such that $x_i = \sum_{j=1}^{l} \alpha_{ij} x_j$ for every $i \in \{l+1, \ldots, k\}$. Let A be a sufficiently large integer. By setting $y_i = x_i/A$ for $i = 1, \ldots, l$, one can ensure that $x_i = \sum_{j=1}^{l} \beta_{ij} y_j$ for every $i = 1, \ldots, k$, where $\beta_{ij} \in \mathbb{N}$. Since $y_1, \ldots, y_l$ are rationally independent, one can find for every $\eta > 0$ some $w \in]-\infty, -w_0[\cup]w_0, +\infty[$ such that $wy_i \bmod 2\pi \in [0, \eta[$ for $i = 1, \ldots, l$. Hence $wx_i \bmod 2\pi \in [0, \epsilon[$ if $\sum_{j=1}^{l} \beta_{ij} \eta \leq \epsilon$ for every $i = 1, \ldots, k$. □

Theorem 10 *Let s be the sign function: $s(x) = 1$ for $x \geq 0$, $s(x) = 0$ otherwise. For any finite set $\{(x_1, y_1), \ldots, (x_k, y_k)\} \subseteq [0, 1]^2$,*

$$\sup_{w \in \mathbb{R}} \left| \frac{1}{k} \sum_{i=1}^{k} s\left(\frac{1 + \cos wx_i}{2} - y_i \right) - \int_0^1 \frac{1 + \cos wx}{2} dx \right| \geq 1/4.$$

Proof. According to Lemma 1, for every $w_0 > 0$ there exist $w > w_0$ such that $s\left(\frac{1+\cos wx_i}{2} - y_i\right) = 1$ for every i such that $y_i < 1$. Two cases can be distinguished.

- If more than 3/4 of the y_i's are smaller than 1, then there are "large" values of w for which the difference between the "empirical area" and the true area is at least 1/4. This is because $\lim_{w \to +\infty} \int_0^1 \frac{1+\cos wx}{2} dx = 1/2$.
- If at least 1/4 of the y_i's are equal to 1, then the difference between the "empirical area" and the true area converges to 1/4 or more when $w \to 0$ ($w \neq 0$).

□

6 Probabilistic complexity classes within $\mathrm{PAR}_{\mathbb{R}}$

In the rest of this paper we shall deal with the theory of computability and complexity over the reals introduced in [2]. The ground concepts and notations in what follows are taken from there. In particular, we denote by $\mathbb{R}^\infty$ the direct sum $\bigoplus_1^\infty \mathbb{R}$ and we define the *size* $|x|$ of an element $x \in \mathbb{R}^\infty$ as the largest i such that its ith coordinate x_i is different from zero. We note that if $\{0,1\}^*$

denotes the set of all finite sequences of elements in $\{0,1\}$ there is a natural inclusion of $\{0,1\}^*$ in $\mathbb{R}^\infty$ that we shall freely use.

Borodin [4] has proved that the class of sets PAR recognized in parallel polynomial time coincides with PSPACE. In the real model the role of PSPACE is then taken over by $\mathrm{PAR}_{\mathbb{R}}$ since after a result of Michaux (cf.[26]) all recursive sets can be recognized with constant work space and consequently, space is not a meaningful resource in the real model. A formal definition of $\mathrm{PAR}_{\mathbb{R}}$ can be found in [8].

In [11] the subclass $\mathrm{DNP}_{\mathbb{R}}$ of $\mathrm{NP}_{\mathbb{R}}$ is defined by requiring the guesses to belong to the set $\{0,1\}$. Based on this class of digital non-deterministic polynomial time one can construct a polynomial time hierarchy as in the Boolean case, and likewise real probabilistic polynomial time classes can be defined.

Definition 6 We shall consider the classes $\Sigma_{k,\mathbb{R}}$, $\Delta_{k,\mathbb{R}}$, $\mathrm{ZPP}_{\mathbb{R}}$, $\mathrm{R}_{\mathbb{R}}$, $\mathrm{BPP}_{\mathbb{R}}$ and $\mathrm{PP}_{\mathbb{R}}$ that are all defined upon $\mathrm{DNP}_{\mathbb{R}}$ by just mimicking their definitions in the Boolean case (see chapters 6 and 8 of [1]). Moreover, if $\mathcal{C}$ is any one of these classes then $\mathcal{C}_{\mathbb{R}}^{=}$ denote the corresponding class defined by equational machines. In the Boolean case the class R is also denoted by RP.

Remark 3 In order to define real probabilistic complexity classes the real model should be equipped with the ability of picking random elements from $\mathbb{R}$, and an issue is the question of which sort of probability measure fits into the BSS model. For the probabilistic classes above considered we have chosen a finite distribution since each randomly chosen element belongs to $\{0,1\}$ with equal probability. This choice has the advantage of its simplicity and it is not far from applications since in the majority of numerical problems where randomization helps it suffices to pick the random elements from a bounded interval of integers.

Some basic properties of the above considered complexity classes follow again in the same manner as in the Boolean case (see Chapters 6 and 8 of [1]). We state them without proof in the following proposition. Recall before from [11] that the set

$$\mathrm{BSCP} = \{\mathcal{B} \mid \exists b \in \{0,1\}^n \text{ that satisfies the decision circuit } \mathcal{B}\}$$

is $\mathrm{DNP}_{\mathbb{R}}$-complete with decision circuit as defined in Section 3.

Proposition 1

i) $\mathrm{ZPP}_{\mathbb{R}} \subseteq \mathrm{R}_{\mathbb{R}} \subseteq \mathrm{DNP}_{\mathbb{R}} \subseteq \mathrm{PP}_{\mathbb{R}} \subseteq \mathrm{PAR}_{\mathbb{R}}$.

ii) $\mathrm{ZPP}_{\mathbb{R}}, \mathrm{BPP}_{\mathbb{R}}, \mathrm{PP}_{\mathbb{R}}$ *are closed under complementation, closed under polynomial time reductions, and the sets*

$$\#\mathrm{BSCP} = \{(\mathcal{B}, i) \mid \text{ s.t. the decision circuit } \mathcal{B} \text{ has more than } i \text{ satisfying binary assignments}\}$$

and

$$\mathrm{CMAJ} = \{\mathcal{B} \mid \text{ s.t. } \mathcal{B} \text{ is satisfied by more than one half of all possible binary assignments}\}$$

are $\mathrm{PP}_{\mathbb{R}}$*-complete under polynomial time reductions.*

iii) $\mathrm{BPP}_{\mathbb{R}} \subseteq \Sigma_{2,\mathbb{R}} \cap \Pi_{2,\mathbb{R}}$.

iv) The above statements also hold for the corresponding equational classes, reductions by equational polynomial time machines and equational decision circuits. □

A first difference with the discrete setting is that for the standard case (machines that branch on order conditions) randomization with bounded probability error does not help.

Theorem 11 *The equality* $\mathrm{P}_{\mathbb{R}} = \mathrm{BPP}_{\mathbb{R}}$ *holds.*

Proof. Let M be a probabilistic machine accepting a set S in polynomial time and with bounded probability error. According to the proof of Corollary 3, for every input size n there are a polynomial number of bits that can act as a deterministic sample for all inputs of size n with respect to the behaviour of M. We consider the real number α whose binary expansion encodes the sequence of all these deterministic samples. A deterministic machine $\tilde{M}$ having α as a built-in constant can then be designed running in polynomial time and accepting the set S. □

Remark 4 Note that the equality above is essentially theoretical in the sense that it does not provide a constructive way for designing $\tilde{M}$. In fact, for some problems, the best deterministic algorithm known up to date (without using non-constructive real constants) takes exponential time while a probabilistic polynomial time solution is possible. This is the case for instance for the problem of deciding whether a given straight-line program computes the zero function.

On the other hand, there are sets for which coin flipping can help (see [5]).

7 On Boolean parts of subclasses of $\mathrm{PAR}_{\mathbb{R}}^{=}$

One research direction that has been worked out intensively during the last few years is the study of the computational power of real Turing machines over binary inputs. The general problem can be roughly stated in the following way. Let us consider a class $\mathcal{C}$ of real Turing machines that work under some resource bound (for instance polynomial time, branching only on equality, etc.). If we restrict these machines to work on binary inputs (i.e. finite words over $\{0,1\}$) they define a class of binary languages $\mathcal{D}$. The question is, what can we say about $\mathcal{D}$ depending on $\mathcal{C}$?

More formally, let us consider the subset $\{0,1\}^*$ of $\mathbb{R}^\infty$ consisting of those vectors whose components belong to $\{0,1\}$. Given any complexity class $\mathcal{C} \subseteq \mathcal{P}(\mathbb{R}^\infty)$ (the set of parts of $\mathbb{R}^\infty$), we define its Boolean part to be the class of binary languages

$$\mathrm{BP}(\mathcal{C}) = \{X \cap \{0,1\}^* : X \in \mathcal{C}\}$$

Our problem now can be stated as: given a complexity class of real sets $\mathcal{C}$ characterize $\mathrm{BP}(\mathcal{C})$.

A possible origin of the problem is the recent interest in the computational power of neural networks. The first results characterized the power of nets with rational weights working within polynomial time by showing that they recognize exactly the sets in P (cf. [31]). The same problem

was then considered for neural networks with real weights and it was shown that they recognize in polynomial time exactly the sets in P/*poly* (cf. [32] and [23]).

It is important to notice that this latter problem deals in a natural way with a framework in which an algebraic model having real constants operates over binary inputs. Other references where this setting is also considered are the papers by Gashkov [14] and by Turán and Vatan [35]. Here the authors consider analog circuits that compute Boolean functions and they focus on lower and upper bounds for the size of those circuits with respect to some Boolean functions. Quoting the latter paper, "It appears that analog circuits form a natural model of computation, and studying the complexity of Boolean functions in this framework may, among other things, contribute to the understanding of the relation between the discrete and continuous aspects of computations".

A further step was taken in [21] which passed from a structured model —the neural net— to a general one —the real Turing machine. However, that paper did not deal with the real Turing machine as introduced in [2] but with a restricted version of it that can do only a moderate use of multiplication namely, all rational functions intermediately computed (in the input variables as well as in the machine's constants) must have degree and coefficient size bounded by the running time. It was shown that $\mathrm{BP}(\mathrm{P}_W) = \mathrm{P}/poly$, where P_W is the class of sets accepted in polynomial time in this weak model.

Subsequently, several papers exhibited new results on Boolean parts. In [12] it was shown that $\mathrm{BP}(\mathrm{PAR}_W) = \mathrm{PSPACE}/poly$ where PAR_W is the class of subsets of $\mathbb{R}^\infty$ recognized in weak parallel polynomial time. Also, for additive machines (i.e. real Turing machines that do not perform multiplications at all), it was shown in [22] that $\mathrm{BP}(\mathrm{P}_{\mathrm{add}}) = \mathrm{P}/poly$ and that $\mathrm{BP}(\mathrm{NP}_{\mathrm{add}}) = \mathrm{NP}/poly$. Here $\mathrm{P}_{\mathrm{add}}$ and $\mathrm{NP}_{\mathrm{add}}$ denote the obvious classes but we recall that the nondeterministic guesses in this model are real numbers. Moreover, if the machines are order-free, i.e. they are required to branch only on equality tests, we now have that $\mathrm{BP}(\mathrm{P}^{=}_{\mathrm{add}}) = \mathrm{P}$ and that $\mathrm{BP}(\mathrm{NP}^{=}_{\mathrm{add}}) = \mathrm{NP}$ ([22]). These results were subsequently generalized in [9] to all the levels of the polynomial hierarchy constructed upon $\mathrm{NP}_{\mathrm{add}}$ (or $\mathrm{NP}^{=}_{\mathrm{add}}$) as well as to the class $\mathrm{PAR}_{\mathrm{add}}$ (or $\mathrm{PAR}^{=}_{\mathrm{add}}$) of sets computed in parallel polynomial time. The Boolean part of the latter are proven there to be PSPACE/*poly* (respectively PSPACE). Finally, in [10] it was shown that the Boolean part of $\mathrm{PAR}_{\mathbb{R}}$ (the class of sets computed in parallel polynomial time with no restrictions on the use of multiplication) is PSPACE/*poly*.

One notes that, in the results quoted above, a clear distinction appears depending on whether the real Turing machines are allowed to branch on inequalities or only on equalities. In the former case, the ability of the machine to code a polynomial advice in a single real number gives to the real Turing machine a computational power proper of non-uniform models. On the contrary, in the latter, the use of an analog model does not yield non-uniformity when restricted to binary inputs. As a matter of fact, the study of the equational case has mostly been done for additive machines and in this case, the Boolean part of a class $\mathcal{C}^{=}_{\mathrm{add}}$ is the Boolean class $\mathcal{C}$ (here $\mathcal{C}$ can be P, NP, EXP, or any level in the polynomial hierarchy). The only previous results on machines with multiplication can be found in [21], where it is shown that $\mathrm{BP}(\mathrm{P}^{=}_{W}) = \mathrm{P}$ and $\mathrm{BP}(\mathrm{P}^{=}_{\mathbb{R}}) \subseteq \mathrm{BPP}$.

The aim of this section is to compute the computational power of real Turing machines (that can freely use multiplication) branching on equalities. Again, the Boolean parts computed will be uniform complexity classes but the extra power of the multiplication in the analog model will

appear under the form of an oracle in R (the class of one-side error probabilistic polynomial time).

We begin with an easy lemma taken from [21] that will be helpful in the sequel.

Lemma 2 *For every equational real Turing machine M with time bound $p : \mathbb{N} \to \mathbb{N}_+$ there exists a constant $c > 0$ and an equational real Turing machine M' such that*

1. *the machine constants $\beta_1, \ldots, \beta_k, \gamma$ of M' are such that $\beta_1, \ldots, \beta_k$ are algebraically independent over $\mathbb{Z}$*
2. *M' does not perform divisions,*
3. *M' has running time bounded by $c \cdot p$,*
4. *the accepted languages in $\mathbb{R}^\infty$ of M and M' coincide.*

Proof. Let $\alpha_1, \ldots, \alpha_p$ be the machine constants of M. If k is the trascendence degree of $\mathbb{Q}(\alpha)$ over $\mathbb{Q}$ we can find elements $\beta_1, \ldots, \beta_k, \gamma$ s.t. $\beta_1, \ldots, \beta_k$ is a transcendence basis of $\mathbb{Q}(\alpha)$ over $\mathbb{Q}$ and $\mathbb{Q}(\alpha) = \mathbb{Q}(\beta)[\gamma]$. Moreover, we can choose $\beta_1, \ldots, \beta_k$ to be any k algebraically independent elements among the α's and the remaining α's can be written as rational functions on $\beta_1, \ldots, \beta_k, \gamma$.

Therefore, calculation of α's from the β's and γ can be done in constant time and one may assume the machine constants to be the β's and γ.

Finally, representing elements $r \in \mathbb{Q}(\beta)$ by pairs $(s,t) \in \mathbb{Z}[\beta]^2$ such that $t \neq 0$ and $r = s/t$, division can be avoided within a constant factor as well. □

We can now state the main theorem of this section.

Theorem 12 *Let $\mathcal{C}$ denote any complexity class appearing in Definition 6. Then*

$$\mathrm{BP}(\mathcal{C}_{\mathbb{R}}^{=}) \subseteq \mathcal{C}^{\mathrm{R}}.$$

One also has the $\mathrm{BP}(\mathrm{PAR}_{\mathbb{R}}^{=}) = \mathrm{PSPACE}^{\mathrm{R}}$.

For the classes $\mathcal{C}$ closed under R-oracles, i.e. satisfying $\mathcal{C}^{\mathrm{R}} = \mathcal{C}$, we get "stability" under Boolean parts.

Corollary 6 *The classes* BPP, PP, PH, *and* PAR *satisfy*

$$\mathrm{BP}(\mathrm{BPP}_{\mathbb{R}}^{=}) = \mathrm{BPP}, \qquad \mathrm{BP}(\mathrm{PP}_{\mathbb{R}}^{=}) = \mathrm{PP},$$
$$\mathrm{BP}(\mathrm{PH}_{\mathbb{R}}^{=}) = \mathrm{PH}, \qquad \mathrm{BP}(\mathrm{PAR}_{\mathbb{R}}^{=}) = \mathrm{PAR} = \mathrm{PSPACE}.$$

Proof. This immediately follows from the well known fact that the above classes are closed under R-oracles. □

Corollary 7 *For the levels of the polynomial hierarchy the following inclusions hold:*

$$\begin{aligned} \mathrm{BP}(\Sigma^{=}_{k,\mathbb{R}}) &\subseteq \Sigma_{k+1}, \\ \mathrm{BP}(\Pi^{=}_{k,\mathbb{R}}) &\subseteq \Pi_{k+1}, \\ \mathrm{BP}(\Delta^{=}_{k,\mathbb{R}}) &\subseteq \Delta_{k+1}. \end{aligned}$$

Proof. The inclusions are consequences of the fact that $\mathrm{R} \subset \Sigma_1 = \mathrm{NP}$. Therefore, we use the fact that, by definition $\Sigma_k^{\mathrm{NP}} = \Sigma_{k+1}$, $\Pi_k^{\mathrm{NP}} = \Pi_{k+1}$, and $\Delta_k^{\mathrm{NP}} = \Delta_{k+1}$. □

We can now return to the proof of Theorem 12.

Let $k \geq 1$ be fixed. Consider the language $\mathrm{ZSLP}_k \subset \{0,1\}^\infty$ of (suitable encodings of) $\{0,1,+,-,*\}$-straight-line programs with k inputs and last result equal to the zero polynomial when executed on input $(X_1,\ldots,X_k) \in \mathbb{Z}[X]^k$. Using the modular techniques from [30], Ibarra and Moran [19] have located it in $\mathrm{co}-\mathrm{R}$.

Proposition 2 [19] *For all $k \geq 1$, $\mathrm{ZSLP}_k \in \mathrm{co}-\mathrm{R}$.* □

Proof of Theorem 12. Let $\mathcal{C}$ one of these classes, M an equational real Turing machine (that can be nondeterministic, probabilistic, parallel, etc) with respective resource bounds recognizing a language L in $\mathcal{C}^{=}_{\mathbb{R}}$. By Lemma 2, M can be assumed to have machine constants $\alpha_1,\ldots,\alpha_k,\gamma$ the first k of them being algebraically independent over $\mathbb{Z}$. We describe now a classical oracle machine $\tilde{M}$ (that, as M can be nondeterministic, probabilistic, parallel, etc) accepting the Boolean part of L. The (finite) "program text" of M is coded into the program of $\tilde{M}$ which behaves as follows: given an input $x_1,\ldots,x_n \in \{0,1\}^n$, $\tilde{M}$ writes down the straight-line program S_1 with $k+1$ "input variables" $\alpha_1,\ldots,\alpha_k,\gamma$ corresponding to a run of M up to the first zero test. Note that we can work modulo the minimal polynomial of γ to keep the degree in γ of the resulting polynomial bounded by a constant. Moreover, in this case, the resulting polynomial is zero if and only if all its coefficients are the zero polynomial in $\mathbb{Z}[\alpha_1,\ldots,\alpha_k]$. Therefore, to decide the outcome of this test, $\tilde{M}$ queries the oracle ZSLP_k. Then $\tilde{M}$ extends S_1 to the straight-line program S_2 corresponding to a run of M up to the second zero test and continues this simulation accordingly until M halts. The machine generates polynomially (in n) many straight-line programs, all of them being of length polynomially (in n). Clearly, $\tilde{M}$ recognizes $L \cap \{0,1\}^*$. □

Acknowledgements

The problem of deterministically simulating probabilistic algorithms over the reals was brought to our attention by Nader Bshouty during a talk at COLT'94. We would like to thank Angus Macintyre for useful discussions on the exponential function and p-addic numbers.

The anonymous referees suggested several improvements in the presentation of the paper, pointed out reference [29] and found a mistake in an earlier version of Theorem 6.

References

[1] J.L. Balcázar, J. Díaz and J. Gabarró. *Structural Complexity I*, EATCS Monographs on Theoretical Computer Science, Springer Verlag, 1988.

[2] L. Blum, M. Shub and S. Smale. On the theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines, *Bulletin of the AMS*, **21**(1), pp. 1–46, 1989.

[3] A. Blumer, A Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, **36**(4), pp. 929–965, 1990.

[4] A. Borodin. On relating time and space to size and depth, *SIAM J. on Computing*, **6**, pp. 733–744, 1977.

[5] P. Bürgisser, M. Karpinski and T. Lickteig. On randomized semi–algebraic test complexity, *J. of Complexity*, **9**, pp. 231–251, 1993.

[6] B. Chazelle. Computational geometry: a retrospective. In *Proc. 26th ACM Symposium on Theory of Computing*, pages 75–94, 1994.

[7] M. Coste and M.F. Roy. Thom's lemma, the coding of real algebraic numbers and the topology of semi-algebraic sets, *J. of Symbolic Computation*, **5**, pp. 121–129, 1988.

[8] F. Cucker. On the Complexity of Quantifier Elimination: the structural approach, *The Computer Journal*, **36**, pp. 400–408, 1993.

[9] F. Cucker and P. Koiran. Computing over the reals with addition and order: higher complexity classes, to appear in *J. of Complexity*.

[10] F. Cucker and D. Grigoriev. On the power of real Turing machines over binary inputs, to appear in *SIAM J. on Computing*.

[11] F. Cucker and M. Matamala. On digital non-determinism, to appear in *Math. Systems Theory*.

[12] F. Cucker, M. Shub and S. Smale. Complexity separations in Koiran's weak model, *Theoretical Computer Science*, **133**, pp. 3–14, 1993.

[13] C. Darken, M. Donahue, L. Gurvits, and E. Sontag. Rate of approximation results for neural network learning. Technical Report 93-07, Siemens Corporate Research, Princeton, NJ, 1993.

[14] S.B. Gashkov. The complexity of the realization of Boolean functions by networks of functional elements and by formulas in bases whose elements realize continuous functions, *Prob. Kibernetiki*, **37**, pp. 52–118, 1980 (in Russian).

[15] P. Goldberg and M. Jerrum. Bounding the Vapnik-Chervonenkis dimension of concept classes parameterized by real numbers. *Machine Learning*, **18**, pp. 131-148, 1995.

[16] D. Haussler. Decision theoretic generalizations of the PAC model for neural nets and other learning applications. *Information and Computation*, **100**, pp. 78–150, 1992.

[17] J. Heintz and C.-P. Schnorr. Testing polynomials which are easy to compute. In *Logic and Algorithmic (an International Symposium held in honour of Ernst Specker)*, pp 237–254. Monographie n^o 30 de L'Enseignement Mathématique, 1982. A preliminary version appeared in *Proc. 12th ACM Symposium on Theory of Computing*, pp. 262-272, 1980.

[18] J. Heintz and M. Sieveking. Lower bounds for polynomials with algebraic coefficients. *Theoretical Computer Science*, **11**, pp. 321-330, 1980.

[19] O.H. Ibarra and S. Moran. Probabilistic algorithms for deciding equivalence of straight-line programs, *J. of the ACM*, **30**, pp. 217–228, 1983.

[20] M. Karpinski and A. Macintyre. Polynomial bounds for VC dimension of sigmoidal neural networks. *Proc. 27th ACM Symposium on Theory of Computing*, 1995, pp.200-208.

[21] P. Koiran. A weak version of the Blum, Shub & Smale model, NeuroColt Report 94–5. Available by anonymous ftp from cscx.cs.rhbnc.ac.uk in /pub/neurocolt/tech_reports. A preliminary version appeared in *Proc. 34th IEEE Symposium on Foundations of Computer Science*, pp. 486-495, 1993.

[22] P. Koiran. Computing over the reals with addition and order, *Theoretical Computer Science*, **133**, pp. 35–47, 1993.

[23] W. Maass. Bounds for the computational power and learning complexity of analog neural nets. In *Proc.* 25^{th} *STOC*, pages 335–344, 1993.

[24] R. Mañé. *Ergodic Theory and Differentiable Dynamics.* Springer-Verlag, NY, 1987.

[25] F. Meyer auf der Heide. Simulating probabilistic by deterministic algebraic computation trees. *Theoretical Computer Science*, **41**, pp. 325–330, 1985.

[26] C. Michaux. Une remarque à propos des machines sur $\mathbb{R}$ introduites par Blum, Shub et Smale, *C.R. Acad. Sci. Paris Série I*, **309**, pp. 435–437, 1989.

[27] J. Milnor. On the Betti numbers of real varieties, *Proc. Amer. Math. Soc.*, **15**, pp. 275–280, 1964.

[28] D. Pollard. *Convergence of Stochastic Processes.* Springer Verlag, 1984.

[29] K. Romanik and J.S. Vitter. Using Vapnik-Chervonenkis dimension to analyse the testing complexity of program segments. Technical Report DUKE-TR-1994-28, Department of Computer Science, Duke University, 1994. This report is available online: `http://www.cs.duke.edu/`.

[30] J.T. Schwartz. Fast probabilistic algorithms for verification of polynomials identities, *J. of the ACM*, **27**, pp. 701–717, 1980.

[31] H. T. Siegelmann and E. D. Sontag. On the computational power of neural nets. In *Proc. Fifth ACM Workshop on Computational Learning Theory*, July 1992.

[32] H. T. Siegelmann and E. D. Sontag. Analog computation via neural networks. *Theoretical Computer Science*, **131**(2):331–360, 1994.

[33] E. D. Sontag. Feedforward nets for interpolation and classification. *J. Comp. Syst. Sci.*, **45**, pp. 20–48, 1992.

[34] R. Thom. Sur l'homologie des variétés algébriques réelles. In *Differential and combinatorial topolology,* edited by S. Cairns, Princeton University Press, 1965.

[35] G. Turán and F. Vatan. On the computation of Boolean functions by analog circuits of bounded fan-in, *Proc. 35th IEEE Symposium on Foundations of Computer Science*, pp. 553–564, 1994.

[36] V. Vapnik. *Estimation of dependences based on empirical data.* Springer Series in Statistics. Springer-Verlag, 1982.

[37] H.E. Warren. Lower bounds for approximation by non-linear manifolds. *Trans. Amer. Math. Soc.*, **133**, pp. 167–178, 1968.