On real Turing machines that toss coins

Felipe Cucker[1]
Universitat Pompeu Fabra[2]

Marek Karpinski[3]
Universität Bonn[4]

Pascal Koiran[5]
DIMACS, Rutgers University[6]

Thomas Lickteig[7]
Universität Bonn[8]

Kai Werther
Universität Bonn[9]

NeuroCOLT Technical Report Series

NC-TR-95-018

February, 1995[10]

Produced as part of the ESPRIT Working Group
in Neural and Computational Learning,
NeuroCOLT 8556

NeuroCOLT Coordinating Partner

Royal Holloway
University of London

Department of Computer Science
Egham, Surrey TW20 0EX, England

For more information contact John Shawe-Taylor at the above address
or email neurocolt@dcs.rhbnc.ac.uk

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

[2]Balmes 132, Barcelona 08008 SPAIN e-mail: `cucker@upf.es`

[3]Partially supported by Leibniz Center for Research in Computer Science, by the DFG grant KA 673/4-1 and by the SERC grant GR-E 68297.

[4]Institut für Informatik Universität Bonn GERMANY e-mail: `marek@cs.uni-bonn.de`

[5]Supported by NeuroCOLT and by an INRIA fellowship.

[6]Piscataway, NJ 08855 USA e-mail: `koiran@dimacs.rutgers.edu`

[7]Supported by DFG Grant Li-405/2-1

[8]Institut für Informatik Universität Bonn GERMANY e-mail: `lickteig@cs.uni-bonn.de`

[9]Institut für Informatik Universität Bonn GERMANY e-mail: `kai@cs.uni-bonn.de`

[10]Received February 8, 1995

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.

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 [16]. After giving a deterministic counterpart to Schwarz's probabilistic algorithm for testing polynomials [25], these authors write that "This statement sounds much like Adleman's (1978) observation that every problem which is decidable in random polynomial time has polynomially 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. [27]). 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 [20] 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 retrieve 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 that section we work with non-uniform models since this can 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. We mainly prove by quite different methods a theorem which is very similar to Heintz and Schnorr's [16]. 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

[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.

characterize its Boolean part, which is defined to be

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

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 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 [19]). 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 [21] and [9] Boolean parts of several complexity classes for additive machines are characterized. Also, in [19] it is shown that $BP(P_W) = P/poly$ and in [12] that $BP(PAR_W) = PSPACE/poly$. Regarding unrestricted multiplications and divisions, it is proved in [10] that $BP(PAR_{\mathbb{R}}) = PSPACE/poly$ and in [19] that $BP(P^{=}_{\mathbb{R}}) \subseteq BPP$.

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 $P_{\mathbb{R}} = BPP_{\mathbb{R}}$. 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.

Due to space constraints, most proofs are given in the appendix.

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 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 $\mathcal{F}$.

If $\mathcal{F}$ is a class of indicator ({0,1}-valued) functions, its pseudo-dimension is usually called *Vapnik-Chervonenkis dimension*.

Goldberg and Jerrum gave bounds on the VC dimension of classes of subsets of $\mathbb{R}^n$ recognized by algebraic decision trees over the reals, or defined by formulas in the first-order theory of the reals [14].

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 ([15], Corollary 2).

Theorem 1 *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". At least at the beginning, we assume that only the usual four arithmetic operations and the order relation are authorized.

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

Theorem 2 *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

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

[2] $\mathcal{F}$ must satisfy a benign measurability condition called permissibility. See for instance Appendix A.1 in [3] or Appendix C in [24]. 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.

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 1, 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 ([27], p. 170) instead of Theorem 1 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 1 *Let C be a probabilistic arithmetic circuit of size s with inputs in $\mathbb{R}^n$ and random elements drawn from an arbitrary probability 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 2 with $X = \mathbb{R}^n$, $Y = \mathbb{R}^m$, g the function computed by C and f the characteristic function of the set accepted by C. In this case, the VC dimension of the resulting $\mathcal{F}$ is $O(sn)$ according to Theorem 2.3 in [14]. □

This result also holds in the more powerful algebraic decision tree model if the circuit size s is replaced by the tree depth T (that was actually the computational model used in Goldberg-Jerrum [14]). Hence we retrieve Meyer auf der Heide's $O(nT^2)$ bound for deterministically simulating probabilistic decision trees [20]. 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 1 remains true if the real number model is extended by adding new operations. Corollary 2 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 2

- 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 2 *Corollary 1 still holds if the real number model is extended by adding any fixed, finite set of semi-algebraic operations to the four arithmetic operations.*

It may also be possible to obtain good bounds in the case where the exponential function is allowed, using the recent results of [18]. 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 2 for approximation algorithms.

Theorem 3 *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 $x \in X$, $\left|\frac{1}{k}\sum_{i=1}^{k} g(x, y_i) - f(x)\right| \leq 2\epsilon$.

By Remark 1, corollaries 1 and 2 clearly hold in this approximation setting as well.

4 Applications

We first give a deterministic version of Schwarz's lemma [25].

Theorem 4 *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 (and no comparisons), uses k real parameters (or "constants") and computes a rational fraction $R_L = P_L/Q_L$ in n variables (P_L and Q_L are polynomials, and $Q_L \not\equiv 0$).*
- *The degree of P_L in the n variables is at most d.*

$\diamond$ *The degree of P_L in the k parameters is at most d'.*

Set $u = 4nd$. Then there exist 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 $L \in \mathcal{L}$, $R_L \equiv 0$ if and only if $R_L(x) = 0$ for every $x \in S$.

In the case where R_L is a polynomial, Heintz and Schnorr [16] 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 4 also holds in this case since it is possible to recover the VC dimension bounds given in [14] 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 (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 [16], and then from Theorem 2.2 in [14].

Corollary 1 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 5 *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 2) by a probabilistic circuit of size $C\sqrt{s/n}$, where C is a universal constant.*

This result is a straightforward consequence of Corollary 1. As Corollary 1, 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 5 holds for arbitrary probability measures.

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

Theorem 6 *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 2 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 1. $\square$

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 2 and 3, 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 8 one of the second kind.

Theorem 7 *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 [26], 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 $i = 1, \ldots, k$.*

See the appendix for a proof.

Theorem 8 *Let s be the sign function: $s(x) = 1$ for $x \geq 0$, $s(x) = 0$ otherwise. For any finite set $\{(x_1, y_1), \ldots, (x_k, y_k)\} \subseteq [0,1]^2$,*

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

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 $PAR_{I\!R}$

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 $I\!R^\infty$ the direct sum $\bigoplus_1^\infty I\!R$ and we define the *size* $|x|$ of an element $x \in I\!R^\infty$ as the largest i such that its i^{th}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 $I\!R^\infty$ that we shall freely use.

We recall the following diagram of inclusions of complexity classes where every arrow stands for an inclusion (cf. [1]).

$$\begin{array}{cccccccc}
 & & R & \to & NP & & & PP \\
 & \nearrow & & \searrow & & \searrow \ \nearrow & & \\
P \to ZPP & & & & BPP & \to & & \\
 & \searrow & & \nearrow & & \nearrow \ \searrow & & \\
 & & co-R & \to & co-NP & & & \Sigma_2 \cap \Pi_2 \ \to
\end{array} \tag{3}$$

$$\begin{array}{cccccc}
 & & \Sigma_2 & & & \\
\to \ \Sigma_2 \cap \Pi_2 & {\nearrow \atop \searrow} & & {\searrow \atop \nearrow} & \Delta_3 \to \Sigma_3 \cap \Pi_3 & {\nearrow \atop \searrow} \ \cdots \to \ PH \to PAR = PSPACE \\
 & & \Pi_2 & & &
\end{array}$$

Borodin [4] has proved that the class of sets PAR accepted in parallel polynomial time coincides with $PSPACE$. In the real model the role of $PSPACE$

is then taken over by $PAR_{I\!R}$ since after a result of Michaux (cf.[23]) all recursive sets can be decided with constant work space and consequently, space is not a meaningful resource in the real model. A formal definition of $PAR_{I\!R}$ can be found in [8].

In [11] the subclass $DNP_{I\!R}$ of $NP_{I\!R}$ 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 3 We shall consider the classes $\Sigma_{k,I\!R}$, $\Delta_{k,I\!R}$, $ZPP_{I\!R}$, $R_{I\!R}$, $BPP_{I\!R}$ and $PP_{I\!R}$ that are all defined upon $DNP_{I\!R}$ 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 $\mathcal{C}^{=}_{I\!R}$ denote the corresponding class defined by equational machines.

Remark 2 In order to define real probabilistic complexity classes the real model should be equipped with the ability of picking random elements from $I\!R$, 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

$$\text{BSCP} = \{\mathcal{B} \mid \exists b \in \{0,1\}^n \text{ s.t. the decision circuit } \mathcal{B} \text{ is satisfied by } b\}$$

is $DNP_{I\!R}$-complete.

Proposition 1 *i)* $ZPP_{I\!R} \subseteq R_{I\!R} \subseteq DNP_{I\!R} \subseteq PP_{I\!R} \subseteq PAR_{I\!R}$.

ii) $ZPP_{I\!R}, BPP_{I\!R}, PP_{I\!R}$ *are closed under complementation, closed under polynomial time reductions, and the sets*

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

and

$$CMAJ = \{\mathcal{B} \mid \text{ s.t. the decision circuit } \mathcal{B} \text{ is satisfied by more than the half of the possible binary assignments}\}$$

are $PP_{I\!R}$*-complete under polynomial time reductions.*

iii) $BPP_{I\!R} \subseteq \Sigma_{2,I\!R} \cap \Pi_{2,I\!R}$.

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 9 *The equality $P_{I\!R} = BPP_{I\!R}$ 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 1, 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. $\square$

Remark 3 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 $PAR^{=}_{I\!R}$

We pass now to the computation of the Boolean parts of some equational classes. In order to do so, the following easy lemma inspired in [19] will be helpful.

Lemma 2 *For every real Turing machine M with time bound $p : I\!N \to I\!N_+$ there exists a constant $c > 0$ and a real Turing machine M' such that*

1. *the machine constants of M' 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 $I\!R^\infty$ of M and M' coincide.*

We can now state the main theorem of this section.

Theorem 10 *Every Boolean complexity class $\mathcal{C}$ appearing in the diagram (3) satisfies*

$$BP(\mathcal{C}^{=}_{I\!R}) \subseteq \mathcal{C}^{R}.$$

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

Corollary 3 *The classes BPP, PP, PH, and PAR satisfy*

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

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

Corollary 4 *For the levels of the polynomial hierarchy the following inclusions hold:*

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

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

The proof of Theorem 10 is based on Proposition 2, and can be found in the appendix.

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 [25], Ibarra and Moran [17] have located it in $co - R$.

Proposition 2 [17] *For all $k \geq 1$, $ZSLP_k \in co - R$.* □

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.

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, preprint, 1994.

[11] F. Cucker and M. Matamala. On digital non-determinism, preprint, 1993.

[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] P. Goldberg and M. Jerrum. Bounding the Vapnik-Chervonenkis dimension of concept classes parameterized by real numbers. *Machine Learning*, 1994. (to appear.) A preliminary version appeared in *Proc. 6th ACM Workshop on Computational Learning Theory*, pp. 361-369, 1993.

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

[16] 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.

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

[18] M. Karpinsky and A. Macintyre. Good bounds for VC dimension of sigmoidal neural networks. preprint, 1994.

[19] P. Koiran. A weak version of the Blum, Shub & Smale model, Technical Report 94–10, DIMACS, 1994. A preliminary version appeared in *Proceedings of 34^{th} IEEE Found. of Comp. Sc.*, pp. 486-495, 1993.

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

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

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

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

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

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

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

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

Appendix: More proofs

Proof of Corollary 2. 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 $u := s(u_1, \ldots, u_k)$ by the subformula $\exists u\ \mathcal{R}(u, u_1, \ldots, u_k)$, where u is a new bound 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 of all these subformulas.

The size of the resulting formula is bounded by an exponential (in fact, polynomial) function of 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 1) is polynomial according to Corollary 2.4 of [14]. □

Proof of Theorem 3. Let us draw k points $y_1, \ldots, y_k$ independently from P. According to Theorem 1, the probability that

$$\left| \frac{1}{k} \sum_{i=1}^{k} g(x, y_i) - E_{y \in Y}\, g(x, y) \right| > \epsilon \tag{4}$$

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 (4) does not hold for any $x \in X$. □

Proof of Theorem 4. Obviously, $R_L \equiv 0$ if and only if $P_L \equiv 0$; and for any $x \in I\!R^n$, $R_L(x) = 0$ implies $P_L(x) = 0$. Hence by Schwarz's lemma, if $R_L \not\equiv 0$ and x is drawn from the uniform distribution on $\{1, \ldots, u\}^n$,

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

This probabilistic test can be "derandomized" by Theorem 2 and Corollary 1, 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 [14].

For $L \in \mathcal{L}$, let $F_L : I\!R^n \to \{0, 1\}$ 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 2 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 2.2 in [14] 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 $N = O(v \log v + k \log d')$. $\square$

Proof of Lemma 1. 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 [26], it follows from Theorem 3.2 and Lemma 2.7 in [22] 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 $i = 1, \ldots, k$. $\square$

Proof of Lemma 2. Let $\alpha_1, \ldots, \alpha_p$ be the machine constants of M. There are $\mathbb{Q}$-linear combinations $\beta_1, \ldots, \beta_k, \gamma$ of the α's such that $\beta_1, \ldots, \beta_k$ is a transcendence basis of $\mathbb{Q}(\alpha)$ over $\mathbb{Q}$ and $\mathbb{Q}(\alpha) = \mathbb{Q}(\beta)[\gamma]$.

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

Using $\mathbb{Q}(\beta)$-symbolic computation in the simple extension $\mathbb{Q}(\beta)[\gamma]$ of $\mathbb{Q}(\beta)$ one may furthermore assume the constants to be the β's alone; a $\leq$-comparison in $\mathbb{Q}(\beta)[\gamma]$ can be simulated in $\mathbb{Q}(\beta)$ by using coding *à la Thom* techniques [7]. The blow up of such a simulation is within a constant factor.

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. $\square$

Proof of Theorem 10. Let $\mathcal{C}$ one of these classes, M a 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 $\alpha_1, \ldots, \alpha_k$ 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 "input variables" $\alpha_1, \ldots, \alpha_k$ corresponding to a run of M up to the first zero test. 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 $L \cap \{0,1\}^\infty$. $\square$