On the Complexity of Quantifier Elimination: the Structural Approach

FELIPE CUCKER*

Universitat Pompeu Fabra, c/ Balmes 132, Barcelona 08008, Spain

The aim of this paper is to survey certain theoretical aspects of the complexity of quantifier elimination in the elementary theory of the real numbers with real constants, and to present some new results on the subject. We use the new model of computation introduced by L. Blum, M. Shub and S. Smale that accepts as inputs vectors of real numbers and allows the transfer of the stuctural approach to computability and complexity for computations with real numbers. More concretely, we give a proof of the existence of $\mathrm{NP}_{\mathbb{R}}$-complete problems. Also, we introduce a new complexity class $\mathrm{PAT}_{\mathbb{R}}$ which describes the complexity of the decision of quantified formulae and, in order to study its relationships with the already existing complexity classes, a model for parallel computations is also introduced. Using the classes resulting by bounding resources in this parallel model, some separation results are finally obtained. In particular, we show that the polynomial hierarchy over the reals is strictly contained in $\mathrm{PAT}_{\mathbb{R}}$.

Received December 1992, revised April 1993

1. INTRODUCTION

The aim of this paper is to survey certain theoretical aspects of the complexity of quantifier elimination in the elementary theory of the real numbers with real constants, and to present some new results on the subject.

Algorithms for quantifier elimination in the theory of the reals appear with the work of A. Tarski [27]. The irruption of computers together with the extent of applications that quantifier elimination has had, focused attention on the complexity of the problem. The complexity of Tarski's original algorithm was hyperexponential and in the early seventies Collins's Cylindrical Algebraic Decomposition (see [9]) reduced it to doubly exponential in the number of variables. Later a similar result was also obtained in [29] based on ideas of Monk and Solovay. More recently, several algorithms were developed in [17, 18, 20, 25], that eliminate quantifiers with a time bound which is singly exponential in the number of variables and doubly exponential in the number of quantifier alternations. Of course, a natural related question is to seek for lower bounds to this problem. An early result in [15] produced an exponential lower bound for the decision of quantified formulæ (i.e. to decide whether a quantified formula with no free variables is true) by nondeterministic Turing machines, and later, [28] and [14] gave a double exponential lower bound for the elimination of quantifiers by deterministic Turing machines.

At this point two important remarks arise. The first one is that the lower bound given by Davenport and Heintz or by Weispfenning does not apply for the decisional problem since it relies on the size of the output for a certain set of inputs. The second one concerns the model of computation considered. In general it is the Turing machine so the inputs considered are formulæ involving polynomials with integer coefficients. It is important to remark that although some of the mentioned algorithms also work with inputs given by real numbers (instead of integers) they were designed for the bit (or Boolean) model, i.e. the machine models used to describe and analyse them are Turing machines or families of Boolean circuits.

For decisional problems, a widespread approach for studying lower bounds is the structural one in which complexity classes are defined by bounding resources between different computational models and their relations are studied. This approach, however, seems to fail in our case since the complexities of basic problems of elementary logic and geometry over the reals are hard to capture by specific complexity models. For instance, the decision problem for existential formulæ is NP-hard (see the Appendix of [16], problem AN10) and the best upper bound we know for it [8, 19, 24] is that this problem belongs to PSPACE (but between this lower and upper bound there is the whole polynomial hierarchy).

A change in this scenery came very recently when [5] introduced a new model of computation that accepts as inputs vectors of real numbers. This theory intends to lay a theoretical ground for Numerical Analysis (see [26] for a discussion about this subject) and moreover provides a uniform machine model for the type of com-

*Partially supported by the ESPRIT BRA Programs of the EC n. 7141 and 6546, projects ALCOM II and PROMotion, and DGICyT PB 89/0379.

putations considered in Algebraic Complexity. These new ideas allowed the transfer of the stuctural approach to computability and complexity for computations with real numbers and the first completeness results as well as the existence of universal machines and undecidable problems appear in their seminal paper above quoted. In the future we shall refer to this approach as BSS theory.

It is also remarkable that, from a logical point of view, the complexity notion introduced in [5] reflects the complexity of the elementary theory of the reals with a constant symbol for every real number, and in this sense the complexity of Tarski's algorithm (or the complexities of Collins's or any subsequent ones that work also for real inputs) is exactly the number of arithmetical operations it performs as a function of the input size.

In the present paper we will mainly deal with complexity aspects of the BSS theory. Results concerning computability (that link BSS theory with dynamical systems, Julia sets and descriptive set theory) can be found in [5, 6, 10, 22]. In the following section, we briefly recall the definition of the BSS model of computation. We shall also define the complexity classes $P_{\mathbf{R}}$ (deterministic polynomial time) and $NP_{\mathbf{R}}$ (nondeterministic polynomial time) as they have been introduced in [5]. In section 3, we give a proof of the existence of $NP_{\mathbf{R}}$-complete problems and we establish its connection with the existential theory of the reals. In section 4, a new class $PAT_{\mathbf{R}}$ is introduced which describes the complexity of the decision of quantified formulæ and, in order to study its relationships with the already existing complexity classes, a model for parallel computations is introduced in the following section. Using the classes resulting by bounding resources in this model, some separation results are obtained in section 6. In particular, we show that the polynomial hierarchy over the reals is strictly contained in $PAT_{\mathbf{R}}$. This result (whose Boolean analog is not known) roughly says that deciding formulæ with a bounded number of alternations is strictly easier than doing so for formulæ with an unbounded one. Finally, in section 7, some open problems are discusssed.

2. THE BSS MODEL AND COMPLEXITY CLASSES

Subsequently we shall denote the direct sum of countably infinite many copies of $\mathbf{R}$ by $\mathbf{R}^\infty$. This direct sum is the set of sequences of real numbers having only a finite number of non-zero elements. For any such element x we will denote by $|x|$ its *size*, i.e. the largest n such that $x_n \neq 0$. Also, if $x = (x_1, x_2, \ldots, x_n, 0, 0 \ldots)$ and $x' = (x'_1, x'_2, \ldots, x'_m, 0, 0 \ldots)$ belong to $\mathbf{R}^\infty$ with sizes $|x| = n$ and $|x'| = m$ we denote by (x, x') the element $x = (x_1, x_2, \ldots, x_n, x'_1, x'_2, \ldots, x'_m, 0, 0 \ldots)$ of $\mathbf{R}^\infty$ of size $n + m$.

We recall from [5] that a *real Turing machine* consists of an input space $I = \mathbf{R}^\infty$, an output space $O = \mathbf{R}^k$, ($k \leq \infty$) and a state space $S = \mathbf{N} \times \mathbf{N} \times \mathbf{R}^\infty$, together with a connected directed graph whose nodes labelled $1 \ldots N$ correspond to the set of different instructions of the machine and are of one of the five following types: input, output, computation, branching and move nodes. Let us describe them a bit more.

1. *Input nodes.* There is only one input node and it is labelled 1. Associated with this node there is a next node $\beta(1)$, and the input map $g_I : I \rightarrow S$.
2. *Output nodes.* There is only one output node and it is labelled N. It has no next nodes, once it is reached the computation halts, and the output map $g_O : S \rightarrow O$ places the result of the computation in the output space.
3. *Computation nodes.* Associated with a node m of this type there is a next node $\beta(m)$ and a map $g_m : S \rightarrow S$. The g_m is of the form $g_m(i, j, z) = (i'(i), j'(j), z'(z))$, with $i'(i) = i+1$ or 1, $j'(j) = j+1$ or 1, and z' is a polynomial or rational map on a finite number of coordinates, and the identity on the others.
4. *Branch nodes.* There are two nodes associated with a node m of this type: $\beta^+(m)$ and $\beta^-(m)$. The next node is $\beta^+(m)$ if $z_1 \geq 0$ and $\beta^-(m)$ otherwise.
5. *Move nodes or fifth nodes.* Nodes m of this type have a unique next node $\beta(m)$. If the current element of S is $(i, j, z_1, \ldots)$ it operates replacing z_j by z_i in the j^{th} place of the vector $\mathbf{R}^\infty$ in S.

The input and output maps can be any polynomial or rational map. However, in what follows we shall take as input map g_I the one that places the input in the z_s with s odd (thus we reserve the even coordinates to leave work space), and its size in z_2. When $O = \mathbf{R}^\infty$ we will take g_O as the identity map on the real coordinates of S. In the case $O = \mathbf{R}^k$ with finite k we shall take g_O as the identity restricted to the first k real coordinates of S.

An *instantaneous description* (or *configuration*) at any moment of the computation can be given by providing an element in S and the current node. After one computational step the first one changes according to the function associated with the current node and the node itself according to the function β.

For a given machine M, the function φ_M associating its output to a given input $x \in \mathbf{R}^\infty$ is called the ***input-output function*** in [5] and we shall say that a function $f : \mathbf{R}^\infty \rightarrow \mathbf{R}^\infty$ is computable when there is a machine M such that $f = \varphi_M$.

Also, a set $A \i \mathbf{R}^\infty$ is *decided* (or *accepted*) by a machine M if its characteristic function $\psi_A : \mathbf{R}^\infty \rightarrow \mathbf{R}$ coincides with φ_M. So, for decisional problems we consider machines whose output space is $\mathbf{R}$.

We can now introduce some central complexity classes.

DEFINITION 2.1. *A real Turing machine M is said* to work in polynomial time *when there are constants $c, q \in \mathbf{N}$ such that for every input $y \in \mathbf{R}^\infty$, M reaches its output node after at most $c|y|^q$ steps. The class $\mathrm{P}_\mathbf{R}$ is then defined as the set of all subsets of $\mathbf{R}^\infty$ that can be accepted by a machine working in polynomial time. Also, M is said* to work in exponential time *when there are constants $c, q \in \mathbf{N}$ such that for every input $y \in \mathbf{R}^\infty$, M reaches its output node after at most $2^{c \cdot |y|^q}$ steps, and the class $\mathrm{EXP}_\mathbf{R}$ consists of all the subsets of $\mathbf{R}^\infty$ that can be accepted by a machine working in exponential time. Finally, a set A belongs to $\mathrm{NP}_\mathbf{R}$ if there is a machine M satisfying the following condition: for all y, $y \in A$ iff there is a $y' \in \mathbf{R}^\infty$ such that M accepts the input (y, y') within time polynomial in $|y|$.*

In this model the element y' can be seen as the sequence of guesses used in the Boolean model. However, we note that in this definition no nondeterministic machines are introduced as a computational model, and nondeterminism appears here as a new acceptance definition for the deterministic machine. Also, we note that the length of y' can be easily bounded by the time bound $p(|y|)$.

According to standard terminology in complexity theory we shall freely use in the rest of the paper the word *problem* as a substitute for subset of $\mathbf{R}^\infty$.

A first trivial result is the following proposition.

PROPOSITION 2.1. *We have the inclusions* $\mathrm{P}_\mathbf{R}$ı $\mathrm{NP}_\mathbf{R}$ *and* $\mathrm{P}_\mathbf{R}$ı $\mathrm{EXP}_\mathbf{R}$.

3. EXISTENTIAL FORMULÆ AND $\mathrm{NP}_\mathbf{R}$-COMPLETENESS

A highly nontrivial (and to date open) question related to the proposition above is whether $\mathrm{P}_\mathbf{R} \neq \mathrm{NP}_\mathbf{R}$. A standard approach to this question is to look for $\mathrm{NP}_\mathbf{R}$-complete problems, since the existence of such a problem A reduces the equality of the classes to the membership of A to $\mathrm{P}_\mathbf{R}$.

One of the main results in [5] is the existence of $\mathrm{NP}_\mathbf{R}$-complete problems. In this section we shall give a proof of it.

DEFINITION 3.1. *Let $\mathcal{C}$ be a class of subsets of $\mathbf{R}^\infty$ and $\mathcal{D}$ a class of functions from $\mathbf{R}^\infty$ to $\mathbf{R}^\infty$. A set Aı$\mathbf{R}^\infty$ is* hard *for $\mathcal{C}$ under reductions in $\mathcal{D}$ when for every set $B \in \mathcal{C}$ there is a function $f : \mathbf{R}^\infty \to \mathbf{R}^\infty$ which belongs to $\mathcal{D}$ such that for every $x \in \mathbf{R}^\infty$, $x \in B$ iff $f(x) \in A$. The set is moreover* complete *for $\mathcal{C}$ if it belongs to $\mathcal{C}$.*

In general, the definition is used with complexity classes satisfying the condition that the resources allowed in $\mathcal{D}$ are smaller than those allowed in $\mathcal{C}$. In that case $\mathcal{C}$-complete problems characterize the class $\mathcal{C}$ in the sense that if a $\mathcal{C}$-complete problem belongs to $\mathcal{D}$, then $\mathcal{D} = \mathcal{C}$.

A subset S of $\mathbf{R}^n$ is called *semi-algebraic* if it is defined by a Boolean combination of inequalities of the form

$$f > 0$$

where f is a n-variate polynomial with coefficients in $\mathbf{R}$. Such a Boolean combination is called a *semi-algebraic* system and it is said to be *satisfiable* when the semi-algebraic set it defines is non-empty. We shall denote by $\mathcal{L}(d)$ the set of all semi-algebraic systems in any number of variables such that all their polynomials have degree bounded by d. Analogously, the space of polynomials having degree bounded by d will be denoted by $\mathcal{P}(d)$.

Semi-algebraic systems correspond to quantifier-free formulæ in the elementary theory of the reals with constant symbols. Since this theory has quantifier elimination we conclude that semi-algebraic sets are just the definable sets of the theory. Comprehensive introductions to the geometry of semi-algebraic sets can be found in [3] and in [7].

Let us now consider the problems

$$\text{dSAS}=\{\varphi \in \mathcal{L}(d) \mid \varphi \text{ is a satisfiable system}\}$$

for $d \in \mathbf{N}$, and

$$\text{dFEAS}= \{f \in \mathcal{P}(d) \mid f \text{ has a real zero}\}.$$

THEOREM 3.1. *i) For every $d \geq 2$ the problem* dSAS *is $\mathrm{NP}_\mathbf{R}$-complete for reductions in $\mathrm{P}_\mathbf{R}$.*

ii) For every $d \geq 4$ the problem dFEAS *is $\mathrm{NP}_\mathbf{R}$-complete for reductions in $\mathrm{P}_\mathbf{R}$.*

Proof. i) For the membership to $\mathrm{NP}_\mathbf{R}$ we consider a machine that guesses values for the variables of the input formula and then evaluates this formula in the guessed values. We shall prove the hardness for $d = 2$ since for greater values of d the result easily follows.

Let us consider a machine M solving a problem B in nondeterministic polynomial time. Without loss of generality, we can suppose that the functions associated to the computational nodes are the arithmetic ones and that they operate z_2 and z_3 placing the result in z_1 (i.e. these functions are of the form $z_1 = z_2 * z_3$ for $* \in \{+, -, \cdot, /\}$). It is clear that such simplification only modifies the complexity of M within a constant multiplicative factor.

Now, let $x \in \mathbf{R}^\infty$ be an element of size n and let $p = p(n)$ be a bound for the running time of M over x. We shall consider the variables $I_t, J_t, Z_{t,i}$ and Y_t for $1 \leq i, t \leq p$ which represent the values of the state space coordinates i, j and z_i and the current node y after t steps. Let us also denote in the following by $\mathcal{C}$ the subset of $\{1, \ldots, N\}$ of computational nodes, and by $\mathcal{B}$ and $\mathcal{M}$ the subsets of branching and move nodes respectively.

The initial configuration (which is obtained after the input is mapped in the state space, and corresponds to the values of the variables with subindex $t = 1$) can be described by the formula Φ_{init}

$$I_1 = 1 \wedge J_1 = 1 \wedge Z_{1,2} = n \wedge Z_{1,4} = x_1 \wedge$$
$$Z_{1,6} = x_2 \wedge \ldots \wedge Z_{1,2n+2} = x_n \wedge Y_1 = \beta \ (1)$$

Also, for any t the change of configuration after t computational steps can be described by the formula Φ_t

$$\bigvee_{m\in\mathcal{C}} [Y_t = m \wedge (I_{t+1}, J_{t+1}, Z_{t+1,1}) = g_m(I_t, J_t, Z_{t,2}, Z_{t,3})]$$
$$\vee \bigvee_{m\in\mathcal{B}} [Y_t = m \wedge ((Z_{t,1} \geq 0 \wedge Y_{t+1} = \beta_+(m) \vee$$
$$(Z_{t,1} < 0 \wedge Y_{t+1} = \beta_-(m))] \vee$$
$$\bigvee_{m\in\mathcal{M}} \bigvee_{\substack{1\leq r\leq p\\ 1\leq s\leq p}} [Y_t = m \wedge I_t = r \wedge J_t = s \wedge Z_{t+1,s} = Z_{t,r}] \vee$$
$$Y_t = 2 \wedge Y_{t+1} = 2$$

In the preceding lines we must add additional equations to require that the values of the rest of the variables remain unchanged (we do not do it here to avoid unnecessary complication of the formulæ).

Moreover, we can describe the fact that M accepts x with Φ_{accept}

$$Y_p = 2 \wedge Z_{p,1} = 1$$

Thus, it turns out that M accepts x iff the sentence

$$\exists Z_{0,1} \exists Z_{0,2} \ldots \exists Z_{p,p} (\Phi_{init} \wedge (\bigwedge_{1\leq t\leq p} \Phi_t) \wedge \Phi_{accept})$$

evaluates to TRUE.

If we define ψ_x to be the semi-algebraic system obtained from the above sentence by "deleting" its quantifiers, we then have that M accepts x iff ψ_x is satisfiable. Moreover, all the polynomials appearing in ψ_x are of degree smaller than or equal to 2 since the g_m are so. Thus, the polynomial time computable function that associates ψ_x to x reduces A to 2SAS.

ii) We now reduce the quantifier-free part of the sentence above to a degree 4 polynomial. The reduction performs the following steps.

a) *Elimination of inequalities.* We replace

$$p(X) \geq 0 \quad \text{by} \quad p(X) = Y^2$$

and

$$p(X) > 0 \quad \text{by} \quad p(X) \cdot Y^2 = 1$$

since the existence of an element Y satisfying the right hand equations is equivalent to their left hand counterparts.

b) *Elimination of connectives.* We iteratively replace a disjunction like

$$\bigvee_{1\leq i\leq s} p_i(X) = 0 \quad \text{by} \quad \prod_{1\leq i\leq s} p_i(X) = 0$$

and a conjunction like

$$\bigwedge_{1\leq i\leq q} p_i(X) = 0 \quad \text{by} \quad \sum_{1\leq i\leq q} p_i(X)^2 = 0$$

Since the original formula has a constant number of connective alternations the size of the resulting polynomial is polynomially bounded.

c) *Reduction to a system of quadrics.* At this moment we have a single polynomial equality

$$\sum_{\nu} a_\nu X_1^{\nu_1} \ldots X_r^{\nu_r} = 0$$

that we replace by the conjunction of a linear equation

$$\sum_{\nu} a_\nu T_\nu = 0$$

in the new variables T_ν, toghether with a set of quadratic equations of the form

$$T_{\alpha+\beta} = T_\alpha T_\beta$$

The number of new variables and equations needed in this process is polynomial in n.

d) *Reduction to a quartic.* We finally obtain a single polynomial of degree 4 by replacing the conjunction of quadrics by the sum of its squares as in step b). We then have that, if F_x is the resulting polynomial, then ψ_x is satisfiable iff F_x has a real root. ∎

It is important to remark that the degree of the polynomials in our complete problems must be bounded by some constant. This need is related to the sparse representation of polynomials together with the fact that in the BSS model large numbers have unit size. Thus, a problem FEAS whose inputs were polynomials of arbitrary degree would not belong to $\mathrm{NP}_{\mathbb{R}}$. In fact, it would not belong to any complexity class given by a deterministic or nondeterministic time bound.

Some other complete problems with an algebraic or geometric meaning are exhibited in [13]. More precisely, it is shown there that the following sets

$$\text{dCONVEX} = \{\varphi \in \mathcal{L}(d) \mid \text{the semi-algebraic set defined by } \varphi \text{ is convex}\},$$
$$\text{dFINITE}_k = \{\varphi \in \mathcal{L}(d) \mid \varphi \text{ has at most } k \text{ solutions}\},$$
$$\text{dPOSITIVE} = \{f \in \mathcal{P}(d) \mid \text{for every } x \in \mathbb{R}^n \ f(x) > 0\},$$
$$\text{dREGULAR} = \{f \in \mathcal{P}(d) \mid f \text{ is regular}\}$$

are $\mathrm{coNP}_{\mathbb{R}}$-complete for every $d \geq 2$ in the first two cases, (and for any $k \in \mathbb{N}$ in the second one) and for every $d \geq 4$ in the other two. Also, [30] shows that problems coming from neural net learning are $\mathrm{NP}_{\mathbb{R}}$-complete.

4. DECISION OF QUANTIFIED FORMULÆ

In the Boolean setting the complexity of the decision of existential formulæ is captured by the class NP, and a similar thing happens with quantified Boolean formulæ and the class PSPACE. In the real case, the use of space bounds does not produce natural complexity classes since it has been shown in [21] that all recursive sets can be decided in linear space. However, since

PSPACE coincides with polynomial alternating time (see [2, ch. 3]) we can define alternation for real Turing machines (modifying the acceptance definition in a similar way as is done for nondeterminism) and an analogous result easily follows.

On the other hand, a hierarchy of complexity classes is defined between P and PSPACE that reflects the complexity of deciding Boolean formulæ with a bounded number of quantifier alternation (see [1, ch. 7]). In this section we shall introduce the real analogy of these classes.

DEFINITION 4.1. *We shall say that a set S is accepted in* Polynomial Alternating Time *if there exists a polynomial p and a machine M such that for every $y \in \mathbf{R}^\infty$, $y \in S$ iff*

$$\exists x_1 \forall z_1 \dots \exists x_{p(|y|)} \forall z_{p(|y|)} \qquad M \text{ accepts}$$

$$(y, x_1, z_1, \dots, x_{p(|y|)}, z_{p(|y|)}) \text{ in time } p(|y|)$$

and we shall denote this fact by $S \in \mathrm{PAT}_{\mathbf{R}}$.

We now define for any $d \geq 1$ the problem dQRF to be the set of true formulæ like

$$Q_1 X1, Q_2 X_2, \dots, Q_n X_n \varphi(X_1, X_2, \dots, X_n)$$

where the Q_i are quantifiers—either existential or universal—and $\varphi(X_1, X_2, \dots, X_n)$ is a semi-algebraic system whose polynomials have degree bounded by d.

Similar arguments as those used in the proof of theorem 3.1 imply the following result.

THEOREM 4.1. *For every $d \geq 2$ the set* dQRF *is* $\mathrm{PAT}_{\mathbf{R}}$*-complete for reductions in* $\mathrm{P}_{\mathbf{R}}$.

We consider the set S_k (resp. P_k) for $k \geq 1$ consisting of the formulæ in 2QRF that have at most $k-1$ alternations of quantifiers and the first one is existential (resp. universal).

DEFINITION 4.2. *For every $k \geq 1$ we define the classes Σ_k^P and Π_k^P as the class of subsets of $\mathbf{R}^\infty$ that reduce in polynomial time to S_k and P_k respectively. Also, we define Δ_k^P to be the intersection of Σ_k^P and Π_k^P, and the* polynomial hierarchy $\mathrm{PH}_{\mathbf{R}}$ *to be the union of all the Σ_k^P.*

REMARK 4.1. It is possible to introduce the classes Σ_k^P and Π_k^P using oracles, just defining Σ_{k+1}^P to be the class of sets accepted by a nondeterministic machine that asks queries of an oracle in Σ_k^P, and Π_k^P to be the class of sets whose complements are in Σ_k^P. We have not done it here in such a way (much more conceptual) to avoid defining the oracle version of the real Turing machine.

5. PARALLEL COMPUTATIONS AND COMPLEXITY CLASSES

DEFINITION 5.1. *An* algebraic circuit *C over $\mathbf{R}$ is a directed acyclic graph where each node has indegree 0, 1, or 2. Nodes with indegree 0 are either labeled as input or with elements of $\mathbf{R}$ (we shall call the last ones constant nodes). Nodes with indegree 2 are labeled with the arithmetic operations of $\mathbf{R}$, i.e. "+", "·", "−" and "/". Finally, nodes with indegree 1 are of a unique kind and are called sign nodes. There is a set of $m \geq 1$ nodes with outdegree 0 called output nodes. In the sequel the nodes of a circuit will be called* gates.

To each gate we inductively associate a function of the input variables in the usual way (note that sign gates return 1 if their input is greater than or equal to 0, and 0 otherwise). In particular, we shall refer to the function associated to the output gates as the function computed by the circuit.

DEFINITION 5.2. *For an arithmetic circuit C, the* size *$s(C)$ of C, is the number of gates in C. The* depth *$d(C)$ of C, is the length of the longest path from some input gate to some output gate.*

Acceptance of subsets of $\mathbf{R}^\infty$ in nonuniform parallel polynomial time can be now easily defined using for a given set S a family of circuits C_n having depth polynomial in n such that C_n computes the characteristic function of S restricted to inputs of size n. We introduce now a uniformity condition and its corresponding class of uniform parallel polynomial time, and we show that this class is contained in $\mathrm{EXP}_{\mathbf{R}}$.

DEFINITION 5.3. *Given an algebraic circuit C, the* canonical encoding *of C is a sequence of 4-tuples of the form $(g, op, g_l, g_r) \in \mathbf{R}^4$ where g represents the gate label, op is the operation performed by the gate, g_l is the gate which provides the left input to g and g_r its right input. By convention g_l and g_r are 0 if gate g is an input gate, and g_r is 0 if gate g is a sign gate (whose input is then given by g_l) or a constant one (the associated constant being then stored in g_l). Also, we shall suppose that the first n gates are the input ones and the last m the output ones.*

DEFINITION 5.4. *Let $\{C_n\}_{n \in \mathbf{N}}$ be a family of circuits. We shall say that the family is* P-uniform *if there exists a real Turing machine M that generates the i^{th} coordinate of the encoding of C_n with input*

$$(i, \overbrace{1, \dots, 1}^{n-1})$$

in time polynomial in n. We shall say that the family is EXP-uniform *when there is a real Turing machine M as above but working in time exponential in n.*

REMARK 5.1. Note that an input like the one in the preceding definition is introduced in a real Turing machine together with its size n in the fourth coordinate of the state space. Thus, the only relevant information of the input (the numbers i and n) are reachable without cost. An equivalent condition is considering a two-dimensional real Turing machine that performs the same computation with input (n, i) also in time $n^{O(1)}$ (but this choice has the undesirable feature of making time depend not on the size of the input but on the input itself).

DEFINITION 5.5. *We shall say that a set S can be decided in* parallel polynomial time *($S \in \mathrm{PAR}_{\mathbf{R}}$ for short) when there is a P-uniform family of circuits $\{\mathcal{C}_n\}$ having depth polynomial in n and such that $\mathcal{C}_n$ computes the characteristic function of S restricted to inputs of size n. In the same way, we define the class $\mathrm{PEXP}_{\mathbf{R}}$ of sets decided in* parallel exponential time.

An alternative definition of parallel polynomial time could be given by assembling an exponential number of real Turing machines and letting them work together within polynomial time. It is however not difficult to prove (see [12] for details) that we would obtain the same class $\mathrm{PAR}_{\mathbf{R}}$ we have defined here.

The defined parallel complexity classes give the formalism within which the algorithms for eliminating quantifiers given in [20] and in [25] are analysed. In fact the following theorem—whose proof is out of the scope of our paper—is shown in the quoted papers.

THEOREM 5.1. *The following inclusions between complexity classes hold*

$$\mathrm{PH}_{\mathbf{R}} \subseteq \mathrm{PAR}_{\mathbf{R}} \text{ and } \mathrm{PAT}_{\mathbf{R}} \subseteq \mathrm{PEXP}_{\mathbf{R}}$$

THEOREM 5.2. *We have the inclusion*

$$\mathrm{PAR}_{\mathbf{R}} \subseteq \mathrm{PAT}_{\mathbf{R}}.$$

Proof. It closely follows some results of the Boolean case (see [2] 4.3 and [1] 3.9) and is a straightforward consequence of the two following lemmas. ∎

LEMMA 5.3. *Every set in $\mathrm{PAR}_{\mathbf{R}}$ can be decided by a real Turing machine working in exponential time and using polynomial space.*

Proof. Let S be a set belonging to $\mathrm{PAR}_{\mathbf{R}}$ and let M be the real Turing machine that generates the circuit gates.

Given a circuit $\mathcal{C}$ with only one output gate let us denote this gate by $\mathrm{out}(\mathcal{C})$. Moreover, if $\mathrm{out}(\mathcal{C})$ is a gate with indegree 2, let us denote by $\mathcal{C}^l$ and $\mathcal{C}^r$ the two subcircuits of $\mathcal{C}$ that have as output node the left and right predecessors of $\mathrm{out}(\mathcal{C})$ respectively, and if $\mathrm{out}(\mathcal{C})$ has indegree 1, let us denote by $\mathcal{C}^u$ the subcircuit of $\mathcal{C}$ that has as output node the unique predecessor of $\mathrm{out}(\mathcal{C})$.

```
EVAL($\mathcal{C}_n, x$)
if $\mathcal{C}_n$ is a leaf then RETURN its value
elsif out($\mathcal{C}_n$) has indegree 1
    then $y$ :=EVAL($\mathcal{C}_n^u, x$);
         RETURN sign($y$)
else $y^l$ :=EVAL($\mathcal{C}_n^l, x$);
     $y^r$ :=EVAL($\mathcal{C}_n^r, x$);
     operate $y^l$ and $y^r$ acording with the type
     of out($\mathcal{C}_n$) and RETURN the result
fi
```

The space used by M' is bounded by the depth of the circuit $\mathcal{C}_n$ (which is polynomial on n) plus the space used by M (which is also polynomial on n). We note that the circuit $\mathcal{C}_n$ is never entirely stored along the computation. Instead, its nodes are generated by M when they are needed. ∎

LEMMA 5.4. *If a set $S \subseteq \mathbf{R}^\infty$ is decided by a real Turing machine in exponential time and using polynomial space, then it reduces to* 2QRF.

Proof. Let S and M be as above with $p(n)$ the polynomial that bounds the space used by M. Then, for any $x \in \mathbf{R}^\infty$ of size n the configuration of the machine M at time t can be described by the vector $(Y, I, J, X_1, \ldots, X_p)$ containing the actual node and the contents of the state space (but only the $p = p(|x|)$ first real coordinates since the machine does not use more coordinates by hypothesis). Now, given $x \in \mathbf{R}^n$ and vectors α, β with $p+3$ coordinates we consider the formulæ

$$\mathrm{Next}(\alpha,\beta),\ \mathrm{Equal}(\alpha,\beta),\ \mathrm{Initial}(\alpha,x),\text{ and }\mathrm{Accepts}(\alpha)$$

whose respective meanings are "β is the configuration resulting from α after one step of M", "α and β are the same configuration", "α is the initial configuration of M with input x" and "α is an accepting configuration".

In the same way as in theorem 3.1 we see that the five formulæ can be constructed by a machine—that depends on M—in polynomial time. We will now construct a new formula $\mathrm{Access\_2}^m(\alpha,\beta)$ describing the fact that the configuration β is reached from α after 2^m steps of M.

If $m = 0$ then, we have

$$\mathrm{Access\_2}^0(\alpha,\beta) = \mathrm{Equal}(\alpha,\beta) \vee \mathrm{Next}(\alpha,\beta)$$

For $m > 0$ our first choice would be

$$\exists\gamma\ \mathrm{Access\_2}^{m-1}(\alpha,\gamma) \wedge \mathrm{Access\_2}^{m-1}(\gamma,\beta)$$

but, it is easy to realize that the resulting expanded formula would have a size exponential in m. Thus, we take advantadge of the fact that the last two terms in the formula above have the same format to avoid writing both of them. In fact, we define $\mathrm{Access\_2}^m(\alpha,\beta)$ in the following way

$$\exists\gamma\forall\alpha'\forall\beta'\ [(\mathrm{Equal}(\alpha',\alpha) \wedge \mathrm{Equal}(\beta',\gamma)) \vee$$
$$(\mathrm{Equal}(\alpha',\gamma) \wedge \mathrm{Equal}(\beta',\beta)) \Rightarrow \mathrm{Access\_2}^{m-1}(\alpha',\beta')]$$

The length of this formula, once recursively expanded, is linear in m and it can be written down in a number of steps polynomial in m.

The reduction from S to 2QRF can be now simply described since, given an $x \in \mathbf{R}^\infty$ it associates to x the formula $\mathrm{Accept}(x)$ defined as

$$\exists\alpha\exists\beta\ (\mathrm{Initial}(\alpha,x) \wedge \mathrm{Accepts}(\beta) \wedge \mathrm{Access2}^{|x|^{O(1)}}(\alpha,\beta))$$

which is constructed in polynomial time in $|x|$, and whose polynomials have degree bounded by 2. ■

6. LOWER BOUNDS AND SEPARATIONS

We show in this section a general lower bound for parallel time given in [23] and deduce from it some other results. The lower bound is an extension for parallel computations to the well known bound of [4].

We begin by a lemma whose proof can be found in [17, 20, 23].

LEMMA 6.1. *Let $\mathcal{F}$ be a finite family of polynomials in $\mathbf{R}[X_1, \dots, X_n]$ and $D = \sum_{f \in \mathcal{F}}$ degree(f). Then we have:*

i) the number of satisfiable systems of the form

$$f_1(X_1, \dots, X_n) \geq 0 \wedge \dots \wedge f_s(X_1, \dots, X_n) \geq 0$$
$$f_{s+1}(X_1, \dots, X_n) > 0 \wedge \dots \wedge f_r(X_1, \dots, X_n) > 0$$
$$f_{r+1}(X_1, \dots, X_n) = 0 \wedge \dots \wedge f_t(X_1, \dots, X_n) = 0$$

where $f_1, \dots, f_t$ are all the polynomials of $\mathcal{F}$, is bounded by $D^{O(n)}$.

ii) the number of connected components of the solution set in $\mathbf{R}^n$ of any of the above systems is bounded by $D^{O(n)}$.

REMARK 6.1. The preceding lemma also holds if we replace the polynomials for rational functions in $\mathbf{R}(X_1, \dots, X_n)$ and define the degree of such a function f/g to be the sum of the degrees of f and g.

We give now a proof of the main result of [23].

THEOREM 6.2. *Let $S\imath\mathbf{R}^\infty$ be a set recognizable in parallel time $t(n)$. If we denote by S_n the subset of S consisting of its elements of size n, then*

$$t(n) = \Omega\left(\sqrt{\frac{\log_2(\#\text{ connected components }(S_n))}{n}}\right)$$

Proof. Let $\mathcal{C}_n$ be the circuit accepting S_n and let $t = t(n)$ its depth.

For each $i \leq t$ let $g_{i,1}, \dots, g_{i,s_i}$ be the sign gates of $\mathcal{C}_n$ whose depth is i, and let $f_{i,1}, \dots, f_{i,s_i}$ be the functions of $(X_1, \dots, X_n)$ they respectively receive as input. Note that for each input $(x_1, \dots, x_n) \in \mathbf{R}^n$, $f_{i,j}$ is a rational function of $(x_1, \dots, x_n)$.

Now, at level i we know that the number s_i of sign gates is bounded by 2^{t-i} (since the indegree of the gates is bounded by 2) and that the degree of their input functions is bounded by 2^i (since at each parallel step the degree can at most get doubled). Thus, applying part i) of the preceding lemma we see that at level 1 there are at most

$$(2^{t-1} \cdot 2)^{O(n)}$$

satisfiable s_1-tuples of sign conditions for $f_{1,1}, \dots, f_{1,s_1}$. Let η_1 be this number.

Each s_1-tuple of satisfiable sign conditions determines the output of the s_1 sign gates at level 1 and thus, determines specific rational functions $f_{2,1}, \dots, f_{2,s_2}$ in the variables $(X_1, \dots, X_n)$ as inputs for the sign gates at level 2. Since the number s_2 of these gates is bounded by 2^{t-2} and the degree of their input functions is bounded by 2^2, applying again the preceding lemma we deduce that for each s_1-tuple σ we have at most

$$(2^{t-2} \cdot 2^2)^{O(n)}$$

satisfiable s_2-tuples of sign conditions for $f^\sigma_{2,1}, \dots, f^\sigma_{2,s_2}$ (the superscript σ is written to recall that these input functions depend on the preceding tuple of sign conditions σ).

We then conclude that the number of satisfiable $(s_1 + s_2)$-tuples of sign conditions is bounded by

$$\eta_1 \cdot (2^{t-2} \cdot 2^2)^{O(n)} = 2^{t \cdot O(n)} 2^{t \cdot O(n)}$$

Iterating t times this argument we obtain a bound for the number of sign conditions satisfied for all the sign gates of $\mathcal{C}_n$ of

$$\overbrace{2^{t \cdot O(n)} \cdot \dots \cdot 2^{t \cdot O(n)}}^{t \text{ times}} = 2^{t^2 \cdot O(n)}$$

By part ii) of the preceding lemma the sets defined by one of these $(s_1 + \dots + s_n)$-tuples of sign conditions have at most

$$(2^t \cdot 2^t)^{O(n)} = 2^{t \cdot O(n)}$$

connected components.

Since S_n is a finite union of some of these sets, we deduce that

$$\#\text{ connected components }(S_n) \leq 2^{t^2 \cdot O(n)}$$

Solving t from this inequality we get the desired bound. ■

We can now use this lower bound to produce two separations, the first of them with a proof different from its original one given in [11].

THEOREM 6.3. *The inclusions* $\mathrm{PAR}_{\mathbf{R}}\imath\,\mathrm{EXP}_{\mathbf{R}}$ *and* $\mathrm{PAR}_{\mathbf{R}}\imath\,\mathrm{PAT}_{\mathbf{R}}$ *are strict.*

Proof. Let us consider the set

$$S = \{x \in \mathbf{R}^\infty \mid (x_1 + ix_2)^{2^{2^{|x|}}} = 1\}$$

For a given size n, S_n is the set of points whose first two coordinates are a 2^{2^n}th root of the unity. Thus, S_n has 2^{2^n} connected components and from the preceding theorem we deduce that $S \notin \mathrm{PAR}_{\mathbf{R}}$.

The membership of S to $\mathrm{EXP}_{\mathbf{R}}$ is given by the algorithm

```
input($x$)
$n := |x|$
$m := 2^n$
for $j = 1 \dots m$ do
```

```
        $a_1 := x_1$
        $x_1 := x_1^2 - x_2^2$
        $x_2 := 2a_1x_2$
    od
    if $x_1 = 1 \wedge x_2 = 0$ then ACCEPT
        else REJECT
    fi
```

whose running time is in $O(2^{|x|})$.

For the membership of S to $\mathrm{PAT}_{\mathbf{R}}$, we shall reduce S to 2QRF using the idea of lemma 5.4 as it is laid out in [14].

Thus, we shall construct a sequence of formulæ $\Phi_0, \Phi_1, \ldots$ in two free (complex) variables z and t the meaning of Φ_n being "t is a 2^{2^n}th root of z".

The formula $\Phi_0(t, z)$ simply says

$$z = t^2$$

As in lemma 5.4 we define for $m \geq 1$ the formula $\Phi_m(t, z)$ to be

$$\exists y \forall v \forall w[(t = v \wedge w = y) \vee (v = y \wedge w = z) \Rightarrow \Phi_{m-1}(w, v)]$$

When expanded, $\Phi_n(t, z)$ is a formula whose length is polynomial (in fact linear) in n and logically equivalent to

$$z = t^{2^{2^n}}$$

Now, replacing each complex variable in $\Phi_n(t, z)$ by two real ones and the equalities

$$z = t^2 \qquad \text{by} \qquad y_1 = x_1^2 - x_2^2 \wedge y_2 = 2x_1x_2$$

and

$$v = w \qquad \text{by} \qquad r_1 = s_1 \wedge r_2 = s_2$$

we obtain a formula in four real variables $\Psi_n(x_1, x_2, y_1, y_2)$ whose length is also linear in n and such that the meaning of $\Psi_n(x_1, x_2, 1, 0)$ is "the complex point $x_1 + ix_2$ is a 2^{2^n}th root of the unity".

The function

$$(x_1, \ldots, x_n) \rightarrow \Psi_n(x_1, x_2, 1, 0)$$

is the reduction we look for. ■

The two separations shown in the theorem above have an additional importance since their Boolean analogues are not known. Thus, the classical question of whether NP equals EXP (which usually comes togheter with the P vs. NP one) has in the real case a negative answer. In fact, one of the weak sides of the structural approach to complexity has been up to now the lack of nontrivial separations. In this sense, theorem 6.3 can give us the hope that in the real setting things will behave better.

We can summarize the relations of the complexity classes considered in this paper in the following diagram, where the arrows mean inclusion and an arrow $\overset{\neq}{\rightarrow}$ means a strict inclusion.

$$\begin{tikzcd}
 & \mathrm{NP}_{\mathbf{R}} \arrow[d, equal] & & & & \\
 & \Sigma_1^P \ldots \arrow[dr] & & & \mathrm{EXP}_{\mathbf{R}} \arrow[dr] & \\
\mathrm{P}_{\mathbf{R}} \arrow[ur] \arrow[dr] & & \mathrm{PH}_{\mathbf{R}} \arrow[r] & \mathrm{PAR}_{\mathbf{R}} \arrow[ur, "\neq"] \arrow[dr, "\neq"] & & \mathrm{PEXP}_{\mathbf{R}} \\
 & \Pi_1^P \ldots \arrow[ur] \arrow[d, equal] & & & \mathrm{PAT}_{\mathbf{R}} \arrow[ur] & \\
 & \text{co-NP}_{\mathbf{R}} & & & &
\end{tikzcd}$$

7. OPEN PROBLEMS

Together with the decision of whether the inclusions in the preceding diagram are strict or not, one of the main problems that remains open is to understand the relation between $\mathrm{EXP}_{\mathbf{R}}$ and $\mathrm{PAT}_{\mathbf{R}}$. In the Boolean case, the decision of quantified formulæ is a PSPACE-complete problem and it is known that PSPACE ı EXP. In our case, however, the decision of quantifier formulæ does not seem to be a problem solvable in exponential time, and nothing is known concerning the inclusions $\mathrm{PAT}_{\mathbf{R}}$ ı $\mathrm{EXP}_{\mathbf{R}}$ and $\mathrm{EXP}_{\mathbf{R}}$ ı $\mathrm{PAT}_{\mathbf{R}}$. An open problem strongly related with this one is to find natural complete problems for $\mathrm{EXP}_{\mathbf{R}}$, since such a problem would give us a good insight about the difficulty of the problems lying in $\mathrm{EXP}_{\mathbf{R}}$.

ACKNOWLEDGEMENT

Thanks are due to Ricard Gavaldà for his very careful reading of the manuscript and his useful comments and suggestions.

REFERENCES

[1] J.L. Balcázar, J. Díaz and J. Gabarró (1988), *Structural Complexity I*, EATCS Monographs of Theoretical Computer Science, n. 11, Springer Verlag.

[2] J.L. Balcázar, J. Díaz and J. Gabarró (1990), *Structural Complexity II*, EATCS Monographs of Theoretical Computer Science, n. 22, Springer Verlag.

[3] R. Benedetti and J.J. Risler (1990), *Real Algebraic and Semialgebraic Sets*, Hermann, Paris.

[4] M. Ben-Or (1983), "Lower bounds for algebraic computation trees", *15th A.C.M. Symp. on Theory of Computing*, pp.80–86.

[5] L. Blum, M. Shub and S. Smale (1989), "On a theory of computation and complexity over the real numbers: *NP*-completeness, recursive functions and universal machines", *Bulletin of the Amer. Math. Soc.*, vol.**21**, n.**1**, pp.1–46.

[6] L. Blum (1990), "Lectures on a theory of computation and complexity over the reals (or an arbitrary ring)", *Lectures in the Sciences of Complexity*, Addison Wesley, pp.1–47.

[7] J. Bochnak, M. Coste and M.-F. Roy (1987), *Géométrie algébrique réelle*. Ergebnisse der Math., 3.Folge, Band **12**, Springer Verlag.

[8] J. Canny (1988), "Some algebraic and geometric computations in PSPACE", *20th A.C.M. Symp. on Theory of Computing*, pp.460–467.

[9] G.E. Collins (1975), "Quantifier elimination for real closed fields by cylindrical algebraic decomposition", *Proc. 2nd. GI Conference Automata Theory and Formal Languages*, LNCS 33, pp.134–183, Springer Verlag.

[10] F. Cucker (1992a), "The arithmetical hierarchy over the reals", *J. of Logic and Comp.*, **2**, pp. 375–395.

[11] F. Cucker (1992b), "$P_R \neq NC_R$", *J. of Complexity*, **8**, pp. 230–238.

[12] F. Cucker, J.L.Montaña and L.M. Pardo (1992), "Models for parallel computations with real numbers", Preprint, Universidad de Cantabria.

[13] F. Cucker and F. Rosselló (1992), "On the complexity of some problems for the Blum, Shub & Smale model", *Proc. LATIN'92*, LNCS 583, pp.117–129, Springer Verlag.

[14] J.H. Davenport and J. Heintz (1988), "Real quantifier elimination is doubly exponential", *J. of Symb. Comp.*, **5**, pp. 29–35.

[15] M.J. Fisher and M.D. Rabin (1974), "Super-exponential complexity of Presburger arithmetic", *Complexity of Computing*, ed. by R.M. Karp, AMS, pp. 27–41.

[16] M.R. Garey and D.S. Johnson (1979), *Computers and Intractability: A Guide to the theory of NP-completeness*, Freeman, San Francisco.

[17] D. Grigor'ev (1988), " Complexity of deciding Tarski algebra". *J. of Symb. Comp.*, **5**, pp.65–108.

[18] D. Grigor'ev and N.N. Vorobjov (Jr.) (1988), "Solving systems of polynomials inequalities in subexponential time", *J. of Symb. Comp.*, **5**, pp.37–64.

[19] J. Heintz, M.-F. Roy and P. Solerno (1989), "On the complexity of semialgebraic sets", *IFIP 89*, Elsevier Science Publishers, North Holland, pp.293–298.

[20] J. Heintz, M.-F. Roy and P. Solerno (1990), "Sur la complexité du principe de Tarski-Seidenberg", *Bull. Soc. Math. France*, **118**, pp.101–126.

[21] C. Mìchaux (1989), "Une remarque à propos des machines sur **R** introduites par Blum, Shub et Smale", *C. R. Acad. Sci. Paris*, t.**309**, Série I, pp.435–437.

[22] C. Michaux (1991), "Ordered rings over which output sets are recursively enumerable", *Proc. Amer. Math. Soc.*, **112**, pp.569–575.

[23] J.L.Montaña and L.M. Pardo (1993), "Lower bounds for arithmetical networks", *Applic. Alg. in Eng. Comm. and Comput.*, **4**, pp. 1–24.

[24] J. Renegar (1988), "A faster PSPACE algorithm for deciding the existential theory of the reals", *Proc. of the 29th Annual Symp. on Foundations of Comp. Sci.*, pp.291–295.

[25] J. Renegar (1992), "On the computational complexity and geometry of the first order theory of the reals", parts I, II and III, *J. of Symb. Comp.*, **13**, pp.255–352.

[26] S. Smale (1990), "Some remarks on the foundations of numerical analysis", *SIAM Review*, **32**, pp. 211–220.

[27] A. Tarski (1951), *A Decision Method for Elementary Algebra and Geometry*, 2nd. ed., Univ. Calif. Press, Berkeley.

[28] V. Weispfenning (1988), "The complexity of linear problems in fields", *J. of Symb. Comp.*, **5**, pp.3–27.

[29] H.R. Wüthrich (1976), "Ein Entscheidungsverfahren für die Theorie der reell-abgeschlossenen Körper", *Komplexität von Entscheidungsproblemen*, ed. by E. Specker, and V. Strassen, LNCS 43, pp.138–162, Springer Verlag.

[30] X.D. Zhang (1992), "Complexity of neural network learning in the real number model", Preprint, University of Massachusets at Amherst.