Laboratoire de l'Informatique du Parallélisme

École Normale Supérieure de Lyon
Unité Mixte de Recherche CNRS-INRIA-ENS LYON
n° 5668

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

SPI

Lower Bounds Are not Easier over the Reals: Inside PH

Hervé Fournier and Pascal Koiran

March 1999

Research Report N° 1999-21

École Normale Supérieure de Lyon
46 Allée d'Italie, 69364 Lyon Cedex 07, France
Téléphone : +33(0)4.72.72.80.37
Télécopieur : +33(0)4.72.72.80.80
Adresse électronique : `lip@ens-lyon.fr`

INRIA

Lower Bounds Are not Easier over the Reals: Inside PH

Hervé Fournier and Pascal Koiran

March 1999

Abstract

We prove that all NP problems over the reals with addition and order can be solved in polynomial time with the help of a boolean NP oracle. As a consequence, the "P = NP?" question over the reals with addition and order is equivalent to the classical question. For the reals with addition and equality only, the situation is quite different since P is known to be different from NP. Nevertheless, we prove similar transfer theorems for the polynomial hierarchy.

Keywords: Transfer theorems, decision trees, point location, algebraic complexity, Blum-Shub-Smale model.

Résumé

On montre que tous les problèmes NP sur les réels avec addition et ordre peuvent être résolus en temps polynomial à l'aide d'un oracle NP booléen. Il en résulte que la question "P=NP ?" sur les réels avec addition et ordre est équivalente à la question classique. Pour les réels avec addition et égalité la situation est bien différente puisqu'il est connu que P $\neq$ NP. Cependant, nous démontrons des résultats de transfert similaires pour la hiérarchie polynomiale.

Mots-clés: Théorèmes de transfert, arbres de décision, localisation de point, complexité algébrique, modèle de Blum-Shub-Smale.

Lower Bounds Are not Easier over the Reals: Inside PH

Hervé Fournier and Pascal Koiran*
Laboratoire de l'Informatique du Parallélisme
Ecole Normale Supérieure de Lyon
46, allée d'Italie
69364 Lyon Cedex 07, France

19th March 1999

Abstract

We prove that all NP problems over the reals with addition and order can be solved in polynomial time with the help of a boolean NP oracle. As a consequence, the "P = NP?" question over the reals with addition and order is equivalent to the classical question. For the reals with addition and equality only, the situation is quite different since P is known to be different from NP. Nevertheless, we prove similar transfer theorems for the polynomial hierarchy.

Keywords: Transfer theorems, decision trees, point location, algebraic complexity, Blum-Shub-Smale model.

1 Introduction

Just as in discrete complexity theory, the main goal of algebraic complexity theory is to prove superpolynomial lower bounds for certain "natural" problems. In several algebraic settings this goal has not been achieved yet. For instance, it is not known whether the resultant of two sparse univariate polynomials can be computed by straight-line programs of polynomial length (see [11] for a motivation); the problem "VP = VNP?" in Valiant's model of computation [12, 13] is still open; and the same is true of the "P = NP?" problem in the most interesting versions of the Blum-Shub-Smale model. It is not always clear whether these algebraic questions are easier than the well-known open questions from discrete complexity theory. Indeed, it was shown in [3] that problems such as Knapsack can be solved in polynomial time on a real (multiplication-free) Turing machine under the hypothesis P = PSPACE. Therefore a superpolynomial lower bound (on the circuit

*Email: [Herve.Fournier,Pascal.Koiran]@ens-lyon.fr.

size, or on the running time of a real Turing machine) for Knapsack would imply a separation of P from PSPACE. In this paper we investigate similar questions for smaller complexity classes. Our main result is the following transfer theorem.

Theorem 1 $\mathrm{P}^0_{\mathbb{R}_{ovs}} = \mathrm{NP}^0_{\mathbb{R}_{ovs}}$ *if and only if* $\mathrm{P} = \mathrm{NP}$.

This implies for instance that Knapsack is in $\mathrm{P}^0_{\mathbb{R}_{ovs}}$ under the hypothesis $\mathrm{P} = \mathrm{NP}$, which is a weaker hypothesis than $\mathrm{P} = \mathrm{PSPACE}$. Here $\mathrm{P}^0_{\mathbb{R}_{ovs}}$ stands for the class of decision problems that can be solved in polynomial time by parameter-free real Turing machines over the structure $\mathbb{R}_{ovs}$ (i.e., the only legal operations are $+$, $-$ and $<$). The main result of [3] was that $\mathrm{P}^0_{\mathbb{R}_{ovs}} = \mathrm{PAR}^0_{\mathbb{R}_{ovs}}$ is equivalent to $\mathrm{P} = \mathrm{PSPACE}$ (PAR stands for "parallel polynomial time"). The complexity theory of $\mathbb{R}_{ovs}$ has been studied in [4, 2]. More background on computation over the reals and other algebraic structures can be found in the textbooks [1, 8].

In this paper, real Turing machines will be parameter-free unless stated otherwise. Results for machines with parameters follow from those for parameter-free machines. For instance:

Corollary 1 $\mathrm{P}_{\mathbb{R}_{ovs}} = \mathrm{NP}_{\mathbb{R}_{ovs}}$ *if and only if* $\mathrm{P/poly} = \mathrm{NP/poly}$.

Our proof of Theorem 1 relies on Meyer auf der Heide's construction of linear decision trees for point location in arrangements of hyperplanes [6, 7]. Roughly speaking, we show in Theorem 2 that his construction can be made uniform if a boolean NP oracle is available. This result is established in section 2, and complexity-theoretic consequences are drawn in section 3 (as a byproduct, we obtain the unexpected result that problems such as the real Traveling Salesman Problem or Knapsack are $\mathrm{NP}_{\mathbb{R}_{ovs}}$-complete for Turing machine reductions). Here Theorem 3 is a key result: problems in $\mathrm{NP}^0_{\mathbb{R}_{ovs}}$ can be solved in polynomial time with the help of a boolean NP oracle. Theorem 1, its corollary, and the completeness results just mentioned then follow immediately.

In the order-free structure $\mathbb{R}_{vs}$ (where addition, subtraction and equality tests are the only operations allowed) the situation is quite different than in $\mathbb{R}_{ovs}$ since it is possible to prove the unconditional result $\mathrm{P}_{\mathbb{R}_{vs}} \neq \mathrm{NP}_{\mathbb{R}_{vs}}$, as shown originally by Meer [5]. It would be interesting to obtain other separation results in this structure. Unfortunately, for several questions (such as the collapse of the polynomial hierarchy $\mathrm{PH}_{\mathbb{R}_{vs}}$ and the separation of $\mathrm{PH}_{\mathbb{R}_{vs}}$ from $\mathrm{PAR}_{\mathbb{R}_{vs}}$) this turns out to be impossible with current techniques: the transfer theorems in section 4 show that these questions are as hard as outstanding open problems from discrete complexity theory.

2 Point location in an arrangement of hyperplanes

We first recall some terminology regarding arrangements of hyperplanes. Let $H = \{h_1, \ldots, h_m\}$ be a set of hyperplanes in $\mathbb{R}^n$. We denote by h_i^+ and h_i^- the two open half-spaces defined by h_i. For a point $x \in \mathbb{R}^n$, we set $z_i(x) = 0$ if $x \in h_i$ and $z_i(x) = 1$ (respectively, -1) if $x \in h_i^+$ (respectively, $x \in h_i^-$). We define $\varphi(x) = (z_1(x), \ldots, z_m(x))$. The faces of the arrangement $\mathcal{A}(H)$ are by definition the classes of the equivalence relation $x \sim y \Leftrightarrow \varphi(x) = \varphi(y)$ on $\mathbb{R}^n$. The dimension of a face is the dimension of its affine closure; a face of dimension 0 is called a vertex, and a face of dimension n a cell.

We can now state the problem. Let $\mathcal{H}_n$ be the set of all hyperplanes defined by equations with integer coefficients in $\{-2^{t(n)}, \ldots, 2^{t(n)}\}$, where t is some fixed polynomial. We say that an algorithm solves the location problem (for the family of arrangements $H_t = (\mathcal{H}_n)_{n\in\mathbb{N}}$) if, on an input point $(x_1, \ldots, x_n) \in \mathbb{R}^n$, it computes a system

$$\mathcal{S} = \begin{cases} f_i(y) < 0 & i = 1, \ldots, p \\ g_j(y) = 0 & j = 1, \ldots, r \end{cases}$$

made up of $p + r = n^{O(1)}$ affine (in)equations with integer polynomial-size coefficients such that the set of points $P_{\mathcal{S}}$ of $\mathbb{R}^n$ satisfying $\mathcal{S}$ is included in a face of $\mathcal{A}(\mathcal{H}_n)$, and $x \in P_{\mathcal{S}}$.

Theorem 2 *The location problem for H_t is in $\mathrm{FP}^0_{\mathbb{R}_{ovs}}(\mathrm{NP})$ for any polynomial $t \in \mathbb{Z}[X]$. This means that a system locating the input point can be computed in polynomial time by a Turing machine over $\mathbb{R}_{ovs}$ using a boolean* NP *oracle.*

Defining formally the model of "real Turing machine with a boolean oracle" used in this theorem would be tedious but completely straightforward. The idea is that such a Turing machine can only use the instructions "write-0" or "write-1" to write on its oracle tape. This ensures that the oracle query is a word of $\{0,1\}^*$, despite the fact that the other tapes of the Turing machine may contain arbitrary real numbers.

Before proving the theorem we have to make an observation about (parameter-free) algorithms over the structure $\mathbb{R}_{ovs}$. By running such an algorithm on the formal input $(X_1, \ldots, X_n)$ and taking all possible paths into account, it is clear that each test is of the form $\sum_{i=1}^{n} a_i X_i + a_{n+1} > 0$ ($a_i \in \mathbb{Z}$). Thus, to a test on an input of length n corresponds an oriented hyperplane of $\mathbb{R}^n$ (having an equation with integer coefficients). This allows us to define a notion of size for the coefficients of tests:

Remark 1 *All tests performed by a program running in time $q(n)$ have coefficients bounded by $2^{q(n)}$.*

Let t be a polynomial in $\mathbb{Z}[X]$, and $\mathcal{L}_n \subseteq \mathbb{R}^n$ the union of the hyperplanes in the arrangement $\mathcal{H}_n$ defined in section 2. Before solving the point location problem for $\mathcal{H}_n$ we will describe an algorithm for recognizing $\mathcal{L}_n$. The union of the $\mathcal{L}_n$ for $n \geq 1$ is a language of $\mathbb{R}^\infty$ denoted L_t.

Proposition 1 *For any polynomial t, L_t is in $\mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{NP})$.*

The remainder of this section is devoted to the proof of Proposition 1 and Theorem 2. The algorithms are based on a construction of Meyer auf der Heide [6, 7], who has described families of polynomial depth linear decision trees deciding unions of hyperplanes.[1] We shall build a uniform machine with an oracle in NP performing the same computation. The proof of Proposition 1 is in three steps: in section 2.1 we describe an algorithm for deciding a union of hyperplanes on $[-1,1]^n$. The size of the tests in this algorithm is analyzed in section 2.2, and the algorithm for deciding a union of hyperplanes in $\mathbb{R}^n$ is then obtained in section 2.3. Finally, this algorithm is turned into a point location algorithm in section 2.4.

2.1 Deciding a union of hyperplanes on a cube

We now describe a recursive method for deciding $\mathcal{L}_n = \bigcup_{h \in \mathcal{H}_n} h$ on $[-1,1]^n$.

Lemma 1 *There is a $\mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{NP})$ algorithm which, for any input $x \in \mathbb{R}^n$, decides whether $x \in \mathcal{L}_n \cap [-1,1]^n$.*

For the complete proof that the algorithm described below really is $\mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{NP})$ we need polynomial size bounds on the coefficients of tests. These bounds are established in section 2.2.

Given a point $y \in \mathbb{R}^n$ and a set $A \subseteq \mathbb{R}^n$, we denote by $\mathrm{Aff}(y, A)$ the affine closure of $\{y\} \cup A$, and by $P(y, A) = \{\lambda y + (1-\lambda)x, x \in A, \lambda < 1\}$ the pyramid of top y and base A. Recursion on the dimension n of the cube is made possible by the following lemma.

Lemma 2 (Meyer auf der Heide) *Let $S = \{h_1, \ldots, h_p\}$ be a set of hyperplanes in $\mathbb{R}^n$ such that the intersection $I = \bigcap_{i=1}^p h_i$ in nonempty. Let A be a polytope on a hyperplane h_0 which does not contain I, and let s be a point in $I \setminus h_0$. If a program decides $L' = \bigcup_{i=1}^p h_i \cap h_0$ on A, then the program obtained by replacing each test h' by $\mathrm{Aff}(s, h')$ (with the appropriate sign) decides $L = \bigcup_{i=1}^p h_i$ on $P(s, A)$.*

Let $x \in P(s, A)$. The previous lemma is clear when we notice the equivalence

$$x \in h_1 \cup \ldots \cup h_p \Leftrightarrow (sx) \cap h_0 \in (h_1 \cap h_0) \cup \ldots \cup (h_p \cap h_0)\ .$$

[1] As mentioned in the introduction, he has also described families of linear decision trees solving the whole location problem.

Now we need a definition. A number $r > 0$ is a *coarseness* of a set of hyperplanes $h_1, \ldots, h_k$ of $\mathbb{R}^n$ if, for any ball B of radius r, either $\{h_i, h_i \cap B \neq \emptyset\} = \emptyset$ or $\bigcap_{h_i \cap B \neq \emptyset} h_i \neq \emptyset$. Let r_n be a coarseness of $\mathcal{H}_n$. In [7] it is shown that one can take $1/r_n = n^{n^2} 2^{2n^2 t(n) + O(n^2)}$.

Here is how the algorithm works.
If $n = 1$ we can decide whether $x \in \mathcal{L}_n$ by binary search (no NP oracle is needed). We now assume that $n > 1$, and set $H_n^0 = \mathcal{H}_n$.
▷ Step 1.
We subdivide the cube $C_n^1 = [-1, 1]^n$ in little cubes with radius smaller than r_n (i.e., with edge length smaller than $2r_n/\sqrt{n}$). By binary search on each coordinate, we find a little cube c_n^1 such that $x \in c_n^1$. Let us call $H_n^1 = \{h \in H_n^0,\ h \cap c_n^1 \neq \emptyset\}$. There are two cases :

(i) $H_n^1 = \emptyset$.

(ii) Otherwise, $\bigcap_{h \in H_n^1} h \neq \emptyset$ by definition of coarseness.

We can check with a boolean NP algorithm whether (i) holds, and reject the input if this is the case. If it turns out that we are in case (ii) we compute in polynomial time with the help of a boolean NP oracle a point s_n^1 in $I^1 = \bigcap_{h \in H_n^1} h$. In order to do this we will in fact compute a strictly decreasing sequence $E_1, \ldots, E_j$ of affine subspaces such that E_1 is an element of H_n^1 and $E_j = I^1$ (note that $j \leq n$). Since the condition $H_n^1 \neq \emptyset$ is in NP, we can compute E_1 by prefix search with an NP oracle. E_{i+1} can be computed from E_i as follows. If it turns out that $E_i \subseteq h$ for all $h \in H_n^1$ we can halt with $I^1 = E_i$. This condition can again be checked with a boolean NP oracle. Otherwise there exists $h \in H_n^1$ such that $\dim(h \cap E_i) = \dim E_i - 1$ (since we are in case (ii) the case $h \cap E_i = \emptyset$ is not possible). Such an h can be determined by prefix search, and we set $E_{i+1} = E_i \cap h$. Finally, when $E_j = I^1$ has been determined we can easily compute a point s_n^1 on this affine space (e.g. by fixing some coordinates 0). If $x = s_n^1$ we accept the input. Otherwise we repeat the same procedure in dimension $n - 1$ as explained below.

First we determine a face f_n^1 of the cube C_n^1 such that x is in $P(s_n^1, f_n^1)$, the pyramid of top s_n^1 and base f_n^1. Let g_n^1 be the affine closure of f_n^1 (the equation of g_n^1 is of the form $x_i = \pm 1$). Then Lemma 2 applies, so it remains to solve a $(n-1)$-dimensional problem: decide whether the point $(s_n^1 x) \cap f_n^1$ lies on $\bigcup_{h \in H_n^1} (g_n^1 \cap h)$ on the $(n-1)$-dimensional cube $[-1, 1]^{n-1}$ of g_n^1. An important point is that if r is a coarseness of a set $\{h, h_1, \ldots, h_p\}$ of hyperplanes, then r is a coarseness of $h \cap h_1, \ldots, h \cap h_p$ on h. Since the hyperplane which plays the role of h (namely g_n^1) is an element of $\mathcal{H}_n$, this will allow us to subdivide the $(n-1)$-dimensional cube with the same r_n (and this remains true at further steps of the recursion).

$\triangleright$ Step k $(1 < k \leq n)$.
At this step, we work on the cube $C_n^k = [-1,1]^{n-k+1}$ of the affine space $\{x_{i_1} = \varepsilon_1, \ldots, x_{i_{k-1}} = \varepsilon_{k-1}\}$ with a projected point x^k (the values of $\varepsilon_j \in \{-1,1\}$ and of the i_j depend on which face of C_n^j was chosen as base of the pyramid at step j). We subdivide C_n^k in smaller cubes, and then locate x^k in a little cube c_n^k of C_n^k. Note that the coordinates of x^k need not be computed explicitly. Instead, a test h' on x^k is done by performing the test $\mathrm{Aff}(s_n^1, (\mathrm{Aff}(s_n^2, \mathrm{Aff}(\ldots, \mathrm{Aff}(s_n^{k-1}, h') \ldots)$ on x. Let H_n^k be the subset of hyperplanes of $\mathcal{H}_n$ that intersect all little cubes $c_n^1, \ldots, c_n^k$ found up to step k. We know that if x lies on an hyperplane of $\mathcal{H}_n$, this point must in fact lie on an hyperplane of H_n^k. If $H_n^k = \emptyset$ we reject x as in step (i). Otherwise we compute a point $s_n^k \in \bigcap_{h \in H_n^k} h$ as in step (ii). If $k = n$ we accept x if $x = s_n^k$, and reject otherwise. If $k < n$ we determine i_k and ε_k, and go one step further into the recursion.

2.2 Coefficients of tests in the location algorithm

What is the running time of the procedure described in section 2.1? As $r_n = 2^{n^{O(1)}}$, locating a point in a small cube by binary search always takes polynomial time. Moreover, it is clear that the number of operations performed (over $\mathbb{Z}$) to compute the coefficients of tests is polynomially bounded. To make sure that the algorithm is really $\mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{NP})$, it just remains to check that the coefficients of these tests are of polynomial size. For this we need a bound on coefficient size for hyperplanes of $\mathbb{R}^n$ given as affine hulls of n points.

Lemma 3 *Let $v_1, \ldots, v_{n-1}$ be linearly independent vectors of $\mathbb{R}^n$ with integer coordinates bounded by A in absolute value. The hyperplane $\mathrm{Vect}(v_1, \ldots, v_{n-1})$ has an equation with integer coefficients bounded by $A^{n-1}(n-1)^{(n-1)/2}$ in absolute value.*

Proof. An equation of this hyperplane is $\det(v_1, \ldots, v_{n-1}, x) = 0$. Its coefficients are $(n-1) \times (n-1)$ minors with entries bounded by A in absolute value. The result then follows from the Hadamard inequality (e.g., [9] p. 7). $\square$

Corollary 2 *Let $a_1, \ldots, a_n$ be n points of $\mathbb{R}^n$ such that their affine hull H is an hyperplane.*

(i) If the a_i's have integer coordinates bounded by B in absolute value, H has an equation with integer coefficients bounded by $B^n n^{n/2}$ in absolute value.

(ii) If the a_i's have rational coordinates with numerators and denominators bounded by C in absolute value, H has an equation with integer coefficients bounded by $C^{n^2+n} n^{n/2}$ in absolute value.

Proof. For (i), let $\overline{a_i}$ be the point of $\mathbb{R}^{n+1}$ with its first n coordinates equal to those of a_i, and its last coordinate equal to 1. By Lemma 3, $\mathrm{Vect}(\overline{a_1}, \ldots, \overline{a_n})$ has an equation of the form $\alpha_1 x_1 + \cdots + \alpha_n x_n + \alpha_{n+1} x_{n+1} = 0$ where the α_i's are integers bounded by $B^n n^{n/2}$ in absolute value. An equation of H is then $\alpha_1 x_1 + \cdots + \alpha_n x_n + \alpha_{n+1} = 0$.

In case (ii), write $a_i = (p_{ij}/q_{ij})_{1 \leq j \leq n}$ where p_{ij} and q_{ij} are integers bounded by C in absolute value. Next, perform the change of variable $y_j = \prod_{i=1}^{n} q_{ij} \cdot x_j$ for $j = 1, \ldots, n$. In the y coordinates, the components of the a_i's are integers bounded by C^n in absolute value. It follows from (i) that in these new coordinates, H has an equation of the form $\alpha_1 y_1 + \cdots + \alpha_n y_n + c = 0$ with coefficients bounded by $C^{n^2} n^{n/2}$ in absolute value. Going back to the x coordinates yields the desired bound. $\square$

When the algorithm is run in dimension n, test hyperplanes are of the form $f_n \circ \ldots \circ f_{s+1}(h_s)$ with $f_i : h \mapsto \mathrm{Aff}(s_n^{n+1-i}, h)$ and h_s a hyperplane created in dimension s (that is, at step $n - s + 1$ of the algorithm). We call "initial hyperplane" such an hyperplane.

The tops of the pyramids are the points y that appear in the functions $f : h \mapsto \mathrm{Aff}(y, h)$. Such a point y lies in $I = \bigcap_{h \in H_n^{n-s+1}} h$. If $\dim I > 0$, we add some equations of the form $x_i = 0$ to obtain an intersection reduced to a point. We obtain y by Cramer's rule: $y = (n_1/d, \ldots, n_s/d)$ where $|d|, |n_i| \leq (2^{t(n)})^n n^{n/2}$ since $H_n^{n-s+1} \subseteq \mathcal{H}_n$.

The initial hyperplanes are of three kinds:

(a) The intersection x_0 of $h \in H_n^{n-s+1}$ with a one-dimensional affine space of the form $\{x_{i_1} = \varepsilon_1, \ldots, x_{i_{n-1}} = \varepsilon_{n-1}\}$ (with $\varepsilon_i \in \{-1, 1\}$).

(b) The hyperplanes used to test in which pyramid the point lies. They are of the form $h = \mathrm{Aff}(y, f)$ where f is the intersection of two faces of the cube $[-1, 1]^s$ in the s-dimensional space $\{x_{i_1} = \varepsilon_{i_1}, \ldots, x_{i_{n-s}} = \varepsilon_{i_{n-s}}\}$ (with $\varepsilon_i \in \{-1, 1\}$).

(c) The hyperplanes used to decompose the unit cube in smaller cubes.

As the bound for (c) dominates the bounds for (a) and (b), we conclude that in $\mathbb{R}^n$ each test hyperplane is the affine hull of n points with rational coordinates having numerators and denominators bounded by $\lceil \sqrt{n}/2 \rceil (1/r_n) \leq n^{n^2} 2^{2n^2 t(n) + O(n^2)}$ in absolute value. Call C this bound. By Corollary 2, each test hyperplane has an equation with integer coefficients bounded by $C^{n^2+n} n^{n/2}$ in absolute value. As a consequence, the length of these coefficients is polynomial in n.

2.3 Deciding a union of hyperplanes on $\mathbb{R}^n$

Let $\tilde{H}_n$ be the family of hyperplanes of $\mathbb{R}^{n+1}$ defined by equations of the form $\sum_{i=1}^n a_i x_i - b x_{n+1} = 0$ where b and the a_i's are integers bounded in absolute value by $2^{t(n)}$, and $(a_1, \ldots, a_n, b) \neq 0$. $\tilde{H}_n$ is obtained from H_n by transforming each hyperplane $h \in H_n$ of $\mathbb{R}^n$ of equation $\sum_{i=1}^n a_i x_i = b$ into the hyperplane $\tilde{h}$ of $\mathbb{R}^{n+1}$ of equation $\sum_{i=1}^n a_i x_i - b x_{n+1} = 0$. We describe below a $\mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{NP})$ algorithm which recognizes the union of the hyperplanes in $\tilde{H}_n$. Proposition 1 will follow by setting $x_{n+1} = 1$.

On a nonzero input $\tilde{x} \in \mathbb{R}^{n+1}$ this algorithm first determines a face f of the cube $[-1,1]^{n+1}$ such that $\tilde{x} \in P(0,f)$ (if $\tilde{x} = 0$ we can accept directly). Let $\overline{f}$ be the affine closure of f and $y = (0\tilde{x}) \cap f$. Since the hyperplanes of $\tilde{H}_n$ have a common intersection point in $0_{\mathbb{R}^{n+1}}$, $\tilde{x}$ lies on an hyperplane of this family if and only if y lies on an hyperplane of $\overline{f}$ in the family $\overline{H_n} = \{\tilde{h} \cap \overline{f};\ \tilde{h} \in \tilde{H}_n\}$. Note that $\overline{H_n}$ is exactly the family of hyperplanes of $\overline{f}$ defined by equations with integers coefficients bounded by $2^{t(n)}$ in absolute value. Given y one can therefore use the $\mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{NP})$ algorithm of Lemma 1 to decide whether this point lies on an hyperplane of $\overline{H_n}$. Let T be the corresponding oracle Turing machine. It is not possible to compute explicitly the coordinates of y from those of $\tilde{x}$ since we work with multiplication-free machines. However, on input $\tilde{x}$ we can emulate the behavior of T on input y. This is done as in section 2.1: in order to test the position of y with respect to an hyperplane h of $\overline{f}$, we test the position of $\tilde{x}$ with respect to $\mathrm{Aff}(0,h)$. The corresponding algorithm remains $\mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{NP})$. This completes the proof of Proposition 1.

2.4 Proof of the location theorem

We are now ready for the proof of Theorem 2. Let $x = (x_1, \ldots, x_n)$ be the point to be located. We use a perturbation method to turn the algorithm of section 2.3 into a point location algorithm. Set $\tilde{x} = (x_1 + \varepsilon_1, \ldots, x_n + \varepsilon_n)$ where $\epsilon_1, \ldots, \epsilon_n$ are infinitely small, positive, and $\varepsilon_1 \ll \epsilon_2 \ll \cdots \ll \varepsilon_n$. Now we run on input $\tilde{x}$ the algorithm of Proposition 1 for deciding $\mathcal{L}_n = \bigcup_{h \in \mathcal{H}_n} h$. Of course we know in advance that $\tilde{x}$ will be rejected since this input does not lie on any hyperplane with integer coefficients; the point is that the collection of all tests performed on $\tilde{x}$ during the execution of this algorithm is a system which locates $\tilde{x}$ in $\mathcal{A}(\mathcal{H}_n)$. Let $\tilde{S} = \{f_1(y) < 0, \ldots, f_q(y) < 0\}$ be this system. Then for each i we test whether $f_i(x) < 0$ or $f_i(x) = 0$: this yields a new system S, which locates x in $\mathcal{A}(\mathcal{H}_n)$. Moreover the size conditions are fulfilled: S is made of $n^{O(1)}$ affine (in)equations with integers coefficients of size $n^{O(1)}$.

3 Transfer theorems for $\mathbb{R}_{ovs}$

We first recall some notations and results on the structure $\mathbb{R}_{ovs} = (\mathbb{R}, +, -, <)$ of the reals with addition and order. A real language (or *real problem*) is a subset of $\mathbb{R}^\infty = \coprod_{n\in\mathbb{N}} \mathbb{R}^n$. The boolean part $\mathrm{BP}(L)$ of a language $L \subseteq \mathbb{R}^\infty$ is by definition $L \cap \{0,1\}^\infty$. For a class $\mathcal{C}$ of real languages, $\mathrm{BP}(\mathcal{C}) = \{\mathrm{BP}(L),\ L \in \mathcal{C}\}$.

Fact 1 $\mathrm{BP}(\mathrm{P}^0_{\mathbb{R}_{ovs}}) = \mathrm{P}$, $\mathrm{BP}(\mathrm{NP}^0_{\mathbb{R}_{ovs}}) = \mathrm{NP}$, $\mathrm{BP}(\mathrm{P}_{\mathbb{R}_{ovs}}) = \mathrm{P/poly}$ *and* $\mathrm{BP}(\mathrm{NP}_{\mathbb{R}_{ovs}}) = \mathrm{NP/poly}$.

We recall that $\mathrm{NDP}^0_{\mathbb{R}_{ovs}}$ is a "digital" version of $\mathrm{NP}^0_{\mathbb{R}_{ovs}}$ where certificates are required to be boolean. More precisely, a problem $A \subseteq \mathbb{R}^\infty$ is in $\mathrm{NDP}^0_{\mathbb{R}_{ovs}}$ if there exists $B \in \mathrm{P}^0_{\mathbb{R}_{ovs}}$ and a polynomial p such that

$$x = (x_1, \ldots, x_n) \in A \Leftrightarrow \exists z \in \{0,1\}^{p(n)} \langle x, z\rangle \in B\ .$$

Fact 2 $\mathrm{NP}^0_{\mathbb{R}_{ovs}} = \mathrm{NDP}^0_{\mathbb{R}_{ovs}}$.

The proofs of these results can be found in [4] and [2]. We can now state and prove a key transfer theorem.

Theorem 3 $\mathrm{NP}^0_{\mathbb{R}_{ovs}} \subseteq \mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{NP})$.

Proof. Let $L \in \mathrm{NP}^0_{\mathbb{R}_{ovs}}$. By Fact 2, there exists $A \in \mathrm{P}^0_{\mathbb{R}_{ovs}}$ and a polynomial r such that

$$(x_1, \ldots, x_n) \in L \Leftrightarrow \exists z \in \{0,1\}^{r(n)} \langle x, z\rangle \in A.$$

For any $x \in \mathbb{R}^n$ and $z \in \{0,1\}^{r(n)}$, the condition $\langle x, z\rangle \in A$ can be checked in polynomial time $t(n)$ by a Turing machine T over $\mathbb{R}_{ovs}$, and the polynomial t depends only on A. Let $\mathcal{H}_n$ be the set of all hyperplanes of $\mathbb{R}^n$ with coefficients in $\{-2^{t(n)}, \ldots, 2^{t(n)}\}$. For any $z \in \{0,1\}^{r(n)}$, if we run $A\langle ., z\rangle$ on the formal input $(X_1, \ldots, X_n)$, each test is of the form $\sum_{i=1}^n a_i X_i + a_{n+1} > 0$, with a_i in $\{-2^{t(n)}, \ldots, 2^{t(n)}\}$. As a consequence, $L \cap \mathbb{R}^n$ is a union of faces of $\mathcal{A}(\mathcal{H}_n)$. The $\mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{NP})$ algorithm deciding L works in two steps.

First, the input $x = (x_1, \ldots, x_n)$ is located in $\mathcal{A}(\mathcal{H}_n)$. By Theorem 2, this can be done in $\mathrm{FP}^0_{\mathbb{R}_{ovs}}(\mathrm{NP})$. The output is a system $\mathcal{S}$ of $n^{O(1)}$ (in)equations of the form $h_i(x) < 0$ or $h_i(x) = 0$ such that the h_i's have polynomial size coefficients and the set $P_{\mathcal{S}}$ of the points of $\mathbb{R}^n$ satisfying $\mathcal{S}$ is included in a face of $\mathcal{A}(\mathcal{H}_n)$.

Then it remains to check whether $P_{\mathcal{S}}$ is included in L or in its complement. This can be done by a standard NP algorithm: we guess a certificate $z \in \{0,1\}^{r(n)}$ and an accepting computation path of T. The set of inputs of $\mathbb{R}^n$ which given z follow this computation path is a polyhedron P defined by a system of linear inequalities of polynomial size. We accept x if

$P \cap P_{\mathcal{S}} \neq \emptyset$. This linear programming problem can be solved in polynomial time. (Another method consists in guessing a rational point $q \in P_{\mathcal{S}}$ with small coordinates – such a point exists, see [4] – and then running T on $\langle q, z \rangle$). $\square$

Remark 2 *Other inclusions can be obtained with this method. The point location algorithm described above can be used for any real language L in a complexity class $\mathcal{C}^0_{\mathbb{R}_{ovs}} \subseteq \mathrm{PAR}^0_{\mathbb{R}_{ovs}}$. Then it remains to check whether the resulting polyhedron $P_{\mathcal{S}}$ is included in L or in its complement. If $\mathcal{C}$ is "reasonable", this will be feasible in $\mathcal{C}_{\mathbb{Z}_2}$. For example, we have $\mathrm{PP}^0_{\mathbb{R}_{ovs}} \subseteq \mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{PP})$, $(\Sigma^k_{\mathbb{R}_{ovs}})^0 \subseteq \mathrm{P}^0_{\mathbb{R}_{ovs}}(\Sigma^k)$ for $k \in \mathbb{N}$ and $\mathrm{PAR}^0_{\mathbb{R}_{ovs}} \subseteq \mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{PSPACE})$. Of course we have $(\Pi^k_{\mathbb{R}_{ovs}})^0 \subseteq \mathrm{P}^0_{\mathbb{R}_{ovs}}(\Pi^k)$ too. For BPP, we only obtain $\mathrm{BPP}^0_{\mathbb{R}_{ovs}} \subseteq \mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{NP} \oplus \mathrm{BPP})$ where $\oplus$ is the join operation.*

The results stated in the introduction (Theorem 1 and its corollary) are direct consequences of Theorem 3.

Proof of Theorem 1. If $\mathrm{P}^0_{\mathbb{R}_{ovs}} = \mathrm{NP}^0_{\mathbb{R}_{ovs}}$, $\mathrm{P} = \mathrm{NP}$ by Fact 1. The converse follows immediately from Theorem 3. $\square$

Proof of Corollary 1. If $\mathrm{P}_{\mathbb{R}_{ovs}} = \mathrm{NP}_{\mathbb{R}_{ovs}}$, P/poly = NP/poly by Fact 1. For the converse, consider a problem A in $\mathrm{NP}_{\mathbb{R}_{ovs}}$. There exists $B \in \mathrm{NP}^0_{\mathbb{R}_{ovs}}$ and parameters $\alpha_1, \ldots, \alpha_p$ such that $(x_1, \ldots, x_n) \in A \Leftrightarrow (x_1, \ldots, x_n, \alpha_1, \ldots, \alpha_p) \in B$. By Theorem 3 $B \in \mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{NP})$, hence $B \in \mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{P/poly})$ by the assumption P/poly = NP/poly. Encoding the advice function in an additional parameter yields $B \in \mathrm{P}_{\mathbb{R}_{ovs}}$, therefore $A \in \mathrm{P}_{\mathbb{R}_{ovs}}$ too. $\square$

We will now give a completeness result. For a real language $L \subseteq \mathbb{R}^\infty$, let us define the integer part of L as

$$\mathrm{IP}(L) = \bigcup_{n \in \mathbb{N}} \{\langle p_1, \ldots, p_n \rangle,\ (p_1, \ldots, p_n) \in L \text{ and } p_1, \ldots, p_n \in \mathbb{Z}\}.$$

For a real complexity class $\mathcal{C}$, we set $\mathrm{IP}(\mathcal{C}) = \{\mathrm{IP}(L),\ L \in \mathcal{C}\}$.

Lemma 4 *Let $A \subseteq \mathbb{R}^\infty$ be such that $\mathrm{IP}(A)$ is Turing NP-hard. Then A is Turing $\mathrm{NP}^0_{\mathbb{R}_{ovs}}$-hard.*

Proof. By Theorem 3, $\mathrm{NP}^0_{\mathbb{R}_{ovs}} \subseteq \mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{NP})$. As $\mathrm{IP}(A)$ is NP-hard, $\mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{NP}) \subseteq \mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{IP}(A))$, and of course $\mathrm{P}^0_{\mathbb{R}_{ovs}}(\mathrm{IP}(A)) \subseteq \mathrm{P}^0_{\mathbb{R}_{ovs}}(A)$. Here $\mathrm{IP}(A)$ is a boolean language used as a boolean oracle: as explained after Theorem 2, such an oracle only handles inputs made of 0's and 1's. We conclude that $\mathrm{NP}^0_{\mathbb{R}_{ovs}} \subseteq \mathrm{P}^0_{\mathbb{R}_{ovs}}(A)$. $\square$

Let us recall the definitions of two classical real languages. The real knapsack problem $\text{Knapsack}_{\mathbb{R}}$ is defined by

$$\text{Knapsack}_{\mathbb{R}} \cap \mathbb{R}^{n+1} = \{(x_1, \ldots, x_n, s),\ \exists u_1, \ldots, u_n \in \{0,1\},\ \sum_{i=1}^{n} u_i x_i = s\}\ .$$

The traveling salesman problem $\text{TSP}_{\mathbb{R}}$ is the set of pairs (A, d) where A is a distance matrix of $\{1, \ldots, n\}$ and $d \in \mathbb{R}^+$, such that there exists a Hamiltonian tour over $\{1, \ldots, n\}$ of length at most d. The final result of this section follows immediately from Lemma 4.

Proposition 2 $\text{Knapsack}_{\mathbb{R}}$ *and* $\text{TSP}_{\mathbb{R}}$ *are Turing* $\text{NP}^0_{\mathbb{R}_{ovs}}$*-complete.*

4 The polynomial hierarchy over $\mathbb{R}_{vs}$

The result $\text{P}_{\mathbb{R}_{vs}} \neq \text{NP}_{\mathbb{R}_{vs}}$ was proved by Meer [5]. In this section we show that similar arguments can be used to separate the lowest levels of the polynomial hierarchy $\text{PH}_{\mathbb{R}_{vs}}$. Separating the higher levels of the hierarchy is essentially "impossible" due to the transfer theorems established in section 4.2. These results rely on elementary observations on the structure of subsets of $\mathbb{R}^n$ definable in $\mathbb{R}_{vs}$ (see Lemma 6 and Lemma 7 in particular). In the following, these subsets will simply be called "definable sets". Remember that definable sets are definable without quantifiers since $\mathbb{R}_{vs}$ admits quantifier elimination. Consequently, a definable set is nothing but a boolean combination of hyperplanes.

As in the rest of the paper, we work with parameter-free machines unless stated otherwise. We point out in Remark 4 at the end of the paper that it is straightforward to generalize our transfer theorems to the case of machines with parameters.

We first recall the generic path method. Let M be a machine over $\mathbb{R}_{vs}$ stopping on all inputs, and L the language decided by M. Given $n \in \mathbb{N}\setminus\{0\}$, we set $L_n = L \cap \mathbb{R}^n$. The generic path in M for inputs of size n is the path obtained by answering *no* to all tests of the form $''h(x) = 0\ ?''$ (unless $h = 0$, in which case we answer *yes*).

This definition is effective in the sense that the generic path can be followed by running M on the formal input $(X_1, \ldots, X_n)$. If M is parameter-free this computation can be carried out on a classical Turing machine. Moreover, if M works in time $t(n)$, it takes time $O(nt(n)^2)$ to apply the generic path method, and the tests that are performed by M along this path can be computed effectively. Let us call $\{h_1, \ldots, h_r\}$ these tests. We have $r \leq t(n)$, and these hyperplanes have the following property: if the inputs following the generic path are rejected, $L_n \subseteq h_1 \cup \ldots \cup h_r$; otherwise these inputs are accepted and $L_n^c \subseteq h_1 \cup \ldots \cup h_r$.

Note that the generic path method can be applied to an affine subspace $X \subseteq \mathbb{R}^n$ instead of $\mathbb{R}^n$, in which case we answer *yes* to a test "$h(x) = 0$?" if and only if $X \subseteq h$. Remember also that a definable subset A of $\mathbb{R}^n$ is dense on X iff it contains an open dense subset of X, and that this is equivalent to $\dim A \cap X = \dim X$. We summarize these observations in a lemma which will be used in section 4.2. In this lemma, Sys^n denotes the set of systems of affine equations in n variables with coefficients in $\mathbb{Z}$. For $S \in \mathrm{Sys}^n$, P_S denotes the affine subspace of $\mathbb{R}^n$ defined by S.

Lemma 5 *Let A be a language of $\mathbb{R}^\infty$ and $A^n = A \cap \mathbb{R}^n$. We denote by L^n the set of systems $S \in \mathrm{Sys}^n$ such that A^n is dense in P_S, and by L the language $\bigcup_{n \geq 1} L^n$. Assume that $A \in \mathcal{C}^0_{\mathbb{R}_{vs}}$, with $\mathcal{C} = \mathrm{PAR}$ or $\mathcal{C} = \Sigma^k$ for some $k \in \mathbb{N}$. Then $L \in \mathcal{C}_{\mathbb{Z}_2}$.*

Proof. Note that A^n is definable for any $A \in \mathcal{C}^0_{\mathbb{R}_{vs}}$ (this is in fact true for any recursive language of $\mathbb{R}^\infty_{vs}$). This set is dense in P_S iff a generic point of P_S belongs to A. We can therefore apply the generic point method described above. More precisely, consider first the case $A \in \mathrm{P}^0_{\mathbb{R}_{vs}}$. Given a test hyperplane h, we can decide in polynomial time whether $P_S \subseteq h$ by linear algebra (for instance, we precompute $d = \dim(P_S)$ and $d+1$ points $x_1, \ldots, x_{d+1}$ such that $P_S = \mathrm{Aff}(x_1, \ldots, x_{d+1})$; then we declare that $P_S \subseteq h$ if $x_i \in h$ for all $i = 1, \ldots, d+1$). The same remark applies to the case $\mathcal{C} = \mathrm{PAR}$ since test hyperplanes still have polynomial-size coefficients in this case. We conclude that L is in P if $A \in \mathrm{P}^0_{\mathbb{R}_{vs}}$, and L is in $\mathrm{PAR}_{\mathbb{Z}_2} = \mathrm{PSPACE}$ if $A \in \mathrm{PAR}^0_{\mathbb{R}_{vs}}$.

If $A \in (\Sigma^k_{\mathbb{R}_{vs}})^0$ for some $k \geq 1$ we use the equivalence between real and boolean alternation for $\mathbb{R}_{vs}$ [2]: there exists a polynomial p and $B \in \mathrm{P}^0_{\mathbb{R}_{vs}}$ such that for any $x \in \mathbb{R}^n$, $x \in A$ iff

$$Q_1 y_1 \in \{0,1\}^{p(n)} \cdots Q_k y_k \in \{0,1\}^{p(n)} \langle x, y_1, \ldots, y_k \rangle \in B$$

(the quantifiers Q_i alternate, starting with $Q_1 = \exists$). The set A^n is dense in P_S iff the statement

$$Q_1 y_1 \in \{0,1\}^{p(n)} \cdots Q_k y_k \in \{0,1\}^{p(n)} F_n(y_1, \ldots, y_k) \tag{1}$$

is true. Here $F_n(y_1, \ldots, y_k)$ stands for: "$\{x \in \mathbb{R}^n;\ \langle x, y_1, \ldots, y_k \rangle \in B\}$ is dense in P_S". Since $B \in \mathrm{P}^0_{\mathbb{R}_{vs}}$, we know that $F_n(y_1, \ldots, y_k)$ can be decided in polynomial time by the generic point method. Therefore (1) shows that $L \in \Sigma^k$. $\square$

Note that in the case $\mathcal{C} = \Sigma^k$ it is really necessary to go to boolean quantification before applying the generic point method (think for instance of the set of points $x \in \mathbb{R}$ defined by the formula $\exists y\ x = y$).

4.1 Separation of Lower Levels

It was pointed out in [10] that the Twenty Questions problem might be a plausible witness to the separation $\mathrm{P}_{\mathbb{C}} \neq \mathrm{NP}_{\mathbb{C}}$. Formally, this problem is defined as follows:

$$\mathrm{TQ} = \bigcup_{n \in \mathbb{N}} \{(x_1, \ldots, x_n) \in \mathbb{R}^n,\ x_1 \in \{0, \ldots, 2^n - 1\}\}.$$

Here we show that Twenty Questions can be used to separate $\Sigma^1_{\mathbb{R}_{vs}}$ from $\Pi^1_{\mathbb{R}_{vs}}$.

Proposition 3 $\mathrm{TQ} \in \Sigma^1_{\mathbb{R}_{vs}} - \Pi^1_{\mathbb{R}_{vs}}$.

Proof. We recall from [10] that $\mathrm{TQ} \in \Sigma^1_{\mathbb{R}_{vs}}$ since the restriction of this problem to $\mathbb{R}^n$ is $\mathrm{TQ}_n = \{(x_1, \ldots, x_n),\ \exists u_1, \ldots, u_n \in \{0,1\}\ x_1 = \sum_i 2^{i-1} u_i\}$.

Assume by contradiction that $\mathrm{TQ} \in \Pi^1_{\mathbb{R}_{vs}}$: TQ_n is therefore the intersection of a finite number of sets A_i of the form $A_i = L_i \cap \mathbb{R}^n$ where $L_i \in \mathrm{P}^0_{\mathbb{R}_{vs}}$. Since TQ_n is not dense in $\mathbb{R}^n$, there exists an A_{i_0} which is also not dense in $\mathbb{R}^n$. The generic path method then implies that A_{i_0} is included in the union of $n^{O(1)}$ hyperplanes. This is impossible since TQ_n is made of 2^n distinct hyperplanes. □

Proposition 4 $(\Sigma^2_{\mathbb{R}_{vs}} \cap \Pi^2_{\mathbb{R}_{vs}}) - (\Sigma^1_{\mathbb{R}_{vs}} \cup \Pi^1_{\mathbb{R}_{vs}}) \neq \emptyset$.

Proof. Consider the following language:

$$\begin{aligned} L = \textstyle\bigcup_{n \in \mathbb{N}} \quad & \{(x_1, \ldots, x_n),\ \exists u_1, \ldots, u_n \in \{0,1\}\ \forall v_1, \ldots, v_n \in \{0,1\} \\ & x_1 = \textstyle\sum_i 2^{i-1} u_i \text{ and } x_2 \neq \sum_j 2^{j-1} v_j\}. \end{aligned}$$

The two quantifier blocks in this definition can be swapped, whence $L \in \Sigma^2_{\mathbb{R}_{vs}} \cap \Pi^2_{\mathbb{R}_{vs}}$. Assume now that $L \in \Sigma^1_{\mathbb{R}_{vs}}$. The restriction $L_n = L \cap \mathbb{R}^n$ is therefore a finite union of sets A_i of the form $L_i \cap \mathbb{R}^n$ where $L_i \in \mathrm{P}^0_{\mathbb{R}_{vs}}$. One of these sets, say A_{i_0}, must be dense in the subspace $S = \{x \in \mathbb{R}^n, x_1 = 0\}$. The generic path method applied to L_{i_0} on this subspace shows that $A^c_{i_0} \cap S$ is included in the union of $n^{O(1)}$ hyperplanes. This is impossible since $L^c \cap S$ is the union of 2^n hyperplanes. A similar argument applied to $S' = \{x \in \mathbb{R}^n, x_2 = 1/2\}$ shows that $L \notin \Pi^1_{\mathbb{R}_{vs}}$. □

4.2 Transfer Theorems for the Polynomial Hierarchy

In this section we show that it will be considerably harder to separate the higher levels of $\mathrm{PH}_{\mathbb{R}_{vs}}$ than the lower levels. We begin with two lemmas. Lemma 6 is a remark on the structure of definable subsets of $\mathbb{R}^n$, and in Lemma 7 we build a generic $\Sigma^2_{\mathbb{R}_{vs}}$ formula deciding a definable set A with the help of the predicate $\dim S \cap A = \dim S$ (the variable S represents an affine subset of $\mathbb{R}^n$).

Lemma 6 *Any nonempty definable set $A \subseteq \mathbb{R}^n$ can be written as*

$$A = E_k \setminus (E_{k-1} \setminus (\ldots \setminus E_0))$$

where E_i is a finite union of affine subspaces, $E_{i-1} \subsetneq E_i$, $E_i = \overline{E_i \setminus E_{i-1}}$, and $k \leq \dim A$.

Proof. If $\dim A = 0$ the result is clearly true since A is a finite set of points. Assume by induction that the result is true for all definable sets of dimension at most $d-1$, and let A be a definable set of dimension d. The topological closure $\overline{A}$ of A is a finite union of affine subspaces. If $A = \overline{A}$ we set $k = 0$ and $E_k = A$. Otherwise, consider the definable set $A_1 = \overline{A} \setminus A$. Since $\dim A_1 \leq d-1$, for some $k \leq d$ one can write by induction hypothesis $A_1 = E_{k-1} \setminus (E_{k-2} \setminus (\cdots \setminus E_0))$ with E_i a finite union of affine subspaces, $E_{i-1} \subsetneq E_i$, $E_i = \overline{E_i \setminus E_{i-1}}$. Since $A = \overline{A} \setminus A_1$, we can take $E_k = \overline{A}$. $\square$

Lemma 7 *For any definable set $A \subseteq \mathbb{R}^n$ we have:*

$$\begin{aligned}
(x_1, \ldots, x_n) \in A \Leftrightarrow & \exists S_1 \forall S_2 \\
& x \in S_1 \wedge \dim A \cap S_1 = \dim S_1 \\
& \wedge(\dim A \cap S_1 \cap S_2 < \dim S_1 \cap S_2 \Rightarrow x \notin S_1 \cap S_2)
\end{aligned}$$

where S_1 et S_2 are affine subspaces of $\mathbb{R}^n$.

Proof. The result is clearly true if $A = \emptyset$. Otherwise, write $A = E_k \setminus (E_{k-1} \setminus (\ldots \setminus E_0))$ as in Lemma 6. Let $x \in A$ and $i_0 = \min\{i;\ i = k \bmod 2,\ x \in E_i\}$. Then $x \notin E_{i_0-1}$: if x belonged to E_{i_0-1}, since $x \in A$ there would exist $i < i_0$ such that $i = k \bmod 2$ and $x \in E_i$. This would be in contradiction with the minimality of i_0.

We first show the implication from left to right: let S_1 be a maximal affine subspace in E_{i_0} containing x. Since $x \in S_1$ and $x \notin E_{i_0-1}$ the strict inclusion $S_1 \cap E_{i_0-1} \subsetneq S_1$ holds. Hence $\dim S_1 \setminus E_{i_0-1} = \dim S_1$ and $\dim A \cap S_1 = \dim S_1$. At last, if $\dim A \cap S_1 \cap S_2 < \dim S_1 \cap S_2$, then $S_1 \cap S_2 \subseteq E_{i_0-1}$. Thus $x \notin S_1 \cap S_2$.

Conversely, assume now that x satisfies the formula for $S_1 = S$. Since $A \cap S$ is definable, by Lemma 6 we can write $A \cap S = E_k \setminus (E_{k-1} \setminus (\ldots \setminus E_0))$. Here $E_k = \overline{A \cap S} = S$ (the second equality follows from $\dim A \cap S = S$). E_{k-1} is a finite union of affine subspaces. For any subspace S_2 in this union we have $\dim A \cap S \cap S_2 < \dim S \cap S_2$, therefore $x \notin S \cap S_2$. This shows that $x \notin E_{k-1}$, hence $x \in A \cap S$. $\square$

Remark 3 *If the definable set A in the above lemma is a boolean combination of hyperplanes with coefficients in some subset $\mathcal{D} \subseteq \mathbb{R}$, then we can quantify only on affine subspaces defined by systems of affine equations with coefficients in $\mathcal{D}$.*

We can now state and prove our transfer theorems for $\mathbb{R}_{vs}$. Note that there is a two level shift in Theorem 4 and Theorem 5.

Theorem 4 $\mathrm{P} = \mathrm{PSPACE} \Rightarrow \mathrm{PAR}^0_{\mathbb{R}_{vs}} = (\Sigma^2_{\mathbb{R}_{vs}})^0 \cap (\Pi^2_{\mathbb{R}_{vs}})^0$.

Proof. Let us assume that P = PSPACE, and let $L \in \mathrm{PAR}^0_{\mathbb{R}_{vs}}$ be decided by a family of P-uniform circuits with depth $t(n)$. It is enough to show that $\mathrm{PAR}^0_{\mathbb{R}_{vs}} = (\Sigma^2_{\mathbb{R}_{vs}})^0$ since $\mathrm{PAR}^0_{\mathbb{R}_{vs}}$ is closed under complement. By Lemma 7, L is decided by the following $(\Sigma^2_{\mathbb{R}_{vs}})^0$ formula:

$$
\begin{aligned}
&(x_1, \ldots, x_n) \in L \Leftrightarrow \exists \mathcal{S}_1 \forall \mathcal{S}_2 \\
&\quad x \in P_{\mathcal{S}_1} \wedge \dim L^n \cap P_{\mathcal{S}_1} = \dim P_{\mathcal{S}_1} \\
&\quad \wedge(\dim L^n \cap P_{\mathcal{S}_1 \cup \mathcal{S}_2} < \dim P_{\mathcal{S}_1 \cup \mathcal{S}_2} \Rightarrow x \notin P_{\mathcal{S}_1 \cup \mathcal{S}_2})
\end{aligned}
$$

where $L^n = L \cap \mathbb{R}^n$, $\mathcal{S}_1$ and $\mathcal{S}_2$ are systems of at most n affine equations with coefficients in $\{-2^{t(n)}, \ldots, 2^{t(n)}\}$ (Remark 3), and $P_{\mathcal{S}}$ is the subspace of $\mathbb{R}^n$ defined by $\mathcal{S}$. By Lemma 5 the condition $\dim L_n \cap P_{\mathcal{S}} = \dim P_{\mathcal{S}}$ can be checked in PSPACE, and therefore in P by hypothesis. □

Theorem 5 *For all $k \geq 0$:*

$$\mathrm{PH} = \Sigma^k \Rightarrow \mathrm{PH}^0_{\mathbb{R}_{vs}} = (\Sigma^{k+2}_{\mathbb{R}_{vs}})^0.$$

Proof. Consider a problem $L \in \mathrm{PH}^0_{\mathbb{R}_{vs}}$: we have $L \in (\Sigma^q_{\mathbb{R}_{vs}})^0$ for some $q \geq 0$. As in the proof of the previous theorem, we use the Σ^2 formula from Lemma 7. Since $\mathrm{PH}^0_{\mathbb{R}_{vs}} \subseteq \mathrm{PAR}^0_{\mathbb{R}_{vs}}$, by Remark 3 we can still quantify on systems of equations with coefficients in $\{-2^{p(n)}, \ldots, 2^{p(n)}\}$, for some polynomial p. By Lemma 5 the condition $\dim P_S \cap L_n = \dim P_S$ can be checked in Σ^q, and thus in Σ^k by hypothesis. Putting the resulting formula in prenex form shows that $L \in (\Sigma^{k+2}_{\mathbb{R}_{vs}})^0$. □

Our final transfer theorem is based on a slightly different technique.

Theorem 6 $\mathrm{P} = \mathrm{NP} \Rightarrow (\Sigma^1_{\mathbb{R}_{vs}})^0 \cap (\Pi^1_{\mathbb{R}_{vs}})^0 = \mathrm{P}^0_{\mathbb{R}_{vs}}$.

Proof. Consider a problem $L \in (\Sigma^1_{\mathbb{R}_{vs}})^0 \cap (\Pi^1_{\mathbb{R}_{vs}})^0$. Since $L \in (\Sigma^1_{\mathbb{R}_{vs}})^0$, there exists $A \in \mathrm{P}^0_{\mathbb{R}_{vs}}$ and a polynomial p such that

$$L = \{x \in \mathbb{R}^\infty;\ \exists u \in \{0,1\}^{p(|x|)}\ \langle x, u\rangle \in A\}.$$

There is also a problem $B \in \mathrm{P}^0_{\mathbb{R}_{vs}}$ and a polynomial q such that

$$L = \{x \in \mathbb{R}^\infty;\ \forall v \in \{0,1\}^{q(|x|)}\ \langle x, v\rangle \in B\}$$

since $L \in (\Pi^1_{\mathbb{R}_{vs}})^0$. We write $A^n_i = \{x \in \mathbb{R}^n; \langle x, i\rangle \in A\}$ and $B^n_j = \{x \in \mathbb{R}^n; \langle x, j\rangle \in B\}$. With these notations, we have $L^n = \bigcup_{i \in \{0,1\}^{p(n)}} A^n_i = \bigcap_{j \in \{0,1\}^{q(n)}} B^n_j$ where $L^n = L \cap \mathbb{R}^n$. Both languages A and B are decidable in polynomial time; let t be a polynomial such that $t(n)$ is an upper bound on the computation time required to:

(i) decide A on inputs of the form $\langle x, u\rangle$ with $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^{p(n)}$;

(ii) decide B on inputs of the form $\langle x, v\rangle$ with $x \in \mathbb{R}^n$ and $v \in \mathbb{R}^{q(n)}$.

Let us consider the two cases: $\dim L^n < n$ or $\dim L^n = n$. In the first case, one of the B_i^n, say $B_{i_0}^n$, must be such that $\dim B_{i_0}^n < n$ since L^n is the (finite) intersection of the B_i^n. As $L^n \subseteq B_{i_0}^n$, by the generic point argument L^n is included in the union of $t(n)$ affine subspaces h_k of dimension $n-1$. In the case $\dim L^n = n$, there must exist $A_{i_0}^n$ such that $\dim A_{i_0}^n = n$. Now $(L^n)^c = \bigcap_i (A_i^n)^c \subseteq (A_{i_0}^n)^c$. Therefore the complement of L^n is included in the union of $t(n)$ affine subspaces h_k of dimension $n-1$. Moreover, the restriction of L^n to a hyperplane h_k (or even any hyperplane) has the same structure.

We now explain how L can be decided in $\mathrm{P}^0_{\mathbb{R}_{vs}}$ under the hypothesis $\mathrm{P} = \mathrm{NP}$. Let x be an input in $\mathbb{R}^n$. At the first step, we use the generic path method to check whether $\dim L^n < n$ or $\dim L^n = n$ (this is an NP problem). First case: $\dim L^n < n$. Then we guess $j \in \{0,1\}^{q(n)}$ such that $\dim B_j^n < n$. Using again the generic path method, we compute at most $t(n)$ hyperplanes whose union contains L^n (this algorithm is in P). Then we check whether x lies on one of these hyperplanes: if it doesn't, we reject; otherwise, we go to the next step. The case $\dim L^n = n$ is similar. We guess $i \in \{0,1\}^{p(n)}$ such that $\dim A_i^n = n$. The generic path method yields at most $t(n)$ hyperplanes whose union contains $(L^n)^c$. It remains to check if x belongs to one of the hyperplanes. If it doesn't we accept, otherwise we go to the next step.

At the beginning of the k-th step, we have computed a system of equations (with integer coefficients) defining an affine subspace of dimension $n-k+1$ on which x lies. We also know that the restriction of L^n to this space has the same structure as in the first step: L^n or its complement, restricted to this space, is included in the union of at most $t(n)$ affine subspaces of dimension $n-k$. Using a boolean NP oracle we can determine these hyperplanes and then we either accept x, reject x, or go to step $k+1$. This algorithm terminates after at most n steps. $\square$

Remark 4 *The three transfer theorems of this section can be extended to the case of machines with parameters. For example, let us show that* $\mathrm{PH} = \Sigma^k \Rightarrow \mathrm{PH}_{\mathbb{R}_{vs}} = \Sigma^{k+2}_{\mathbb{R}_{vs}}$. *For any problem* $L \in \mathrm{PH}_{\mathbb{R}_{vs}}$ *there exist parameters* $\alpha_1, \ldots, \alpha_p$ *and a problem* $L' \in \mathrm{PH}^0_{\mathbb{R}_{vs}}$ *such that* $(x_1, \ldots, x_n) \in L$ *iff* $(x_1, \ldots, x_n, \alpha_1, \ldots, \alpha_p) \in L'$. *By Theorem 5,* $L' \in (\Sigma^{k+2}_{\mathbb{R}_{vs}})^0$ *under the assumption* $\mathrm{PH} = \Sigma^k$. *This implies that* $L \in \Sigma^{k+2}_{\mathbb{R}_{vs}}$.

References

[1] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation*. Springer-Verlag, 1998.

[2] F. Cucker and P. Koiran. Computing over the reals with addition and order: Higher complexity classes. *Journal of Complexity*, 11:358–376, 1995.

[3] H. Fournier and P. Koiran. Are lower bounds easier over the reals ? In *Proc. 30th ACM Symposium on Theory of Computing*, pages 507–513, 1998.

[4] P. Koiran. Computing over the reals with addition and order. *Theoretical Computer Science*, 133(1):35–48, 1994.

[5] K. Meer. A note on a $P \neq NP$ result for a restricted class of real machines. *Journal of Complexity*, 8:451–453, 1992.

[6] F. Meyer auf der Heide. A polynomial linear search algorithm for the n-dimensional knapsack problem. *Journal of the ACM*, 31(3):668–676, 1984.

[7] F. Meyer auf der Heide. Fast algorithms for n-dimensional restrictions of hard problems. *Journal of the ACM*, 35(3):740–747, 1988.

[8] B. Poizat. *Les Petits Cailloux*. Nur Al-Mantiq Wal-Ma'rifah **3**. Aléas, Lyon, 1995.

[9] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, New-York, 1986.

[10] M. Shub and S. Smale. On the intractability of Hilbert's Nullstellensatz and an algebraic version of "P=NP". *Duke Mathematical Journal*, 81(1):47–54, 1996.

[11] S. Smale. On the P=NP problem over the complex numbers. Lecture given at the MSRI workshop on Complexity of Continuous and Algebraic Mathematics, November 1998. Lecture on video at www.msri.org.

[12] L. G. Valiant. Completeness classes in algebra. In *Proc. 11th ACM Symposium on Theory of Computing*, pages 249–261, 1979.

[13] L. G. Valiant. Reducibility by algebraic projections. In *Logic and Algorithmic (an International Symposium held in honour of Ernst Specker)*, pages 365–380. Monographie n^o 30 de L'Enseignement Mathématique, 1982.