ELSEVIER

Theoretical Computer Science 219 (1999) 49–64

Theoretical Computer Science

www.elsevier.com/locate/tcs

Equality is a jump

Paolo Boldi [1], Sebastiano Vigna *,[1]

Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano, Via Comelico 39/41, I-20135 Milano MI, Italy

Abstract

We define a notion of *degree of unsolvability* for subsets of R^n (where R is a real closed Archimedean field) and prove that, in contrast to Type 2 computability, the presence of exact equality in the BSS model forces exactly one jump of the unsolvability degree of decidable sets.
© 1999 Elsevier Science B.V. All rights reserved.

1. Introduction

The problem of extending classical recursion theory to the non-discrete world of real numbers has given rise to two complementary approaches: following the tradition of Turing, one can extend the notion of Turing machine by allowing input and output tapes to contain (infinite) representations of real numbers; this approach is known as *Type 2 recursion theory* [14]. On the other hand, it is possible to consider the reals as basic atomic entities, on which exact computations and tests are permitted, as in the *BSS model* [2].

In this paper we want to push further the analysis of the relations between decidability in the Turing machine and in the BSS model started in [3]. We give a notion of *degree of a subset of* R^n, and use some of the results we obtained there in order to build a BSS decidable set whose degree is at least one jump more than that of the constants of the deciding machine. This is a precise degree-theoretic formalization of the slogan "equality[2] is at least one jump" [3], which prompted the conjecture that equality is *exactly* one jump. In [3] we stated a purely BSS-theoretic version of the conjecture for semi-decidable sets, which turned out to be false (see Theorem 12); in this paper, we state and prove the degree-theoretic version of the conjecture for BSS decidable sets, and to this purpose we give a series of intermediate results of

* Corresponding author. E-mail: vigna@dsi.unimi.it.

[1] The authors have been partially supported by the Esprit Working Group No. 8556 (NeuroCOLT).

[2] With "equality" we refer to the possibility for a BSS machine to perform *exact* equality tests, as opposed to Type 2 Turing machines, which can just perform approximate comparisons.

0304-3975/99/$ - see front matter © 1999 Elsevier Science B.V. All rights reserved.
PII: S0304-3975(98)00283-7

independent interest: in particular, we show that the nonempty halting set problem for BSS machines is *Turing* semi-decidable (relativized to the constants appearing in the BSS machines), and that the interior and exterior of a BSS decidable set are Type 2 semi-decidable, given the knowledge of a jump.

The results of this paper draw from four sources: model theory (in particular, Tarski's quantifier elimination for real closed fields and related results [9]), field extension theory (in order to perform exact computations on finitely generated extensions of $\mathbf{Q}$ [8]), topology (because of topological properties typical of Type 2 computability) and degree theory (the notion of jump will prove to be fundamental [11]). We shall also use some Gröbner bases techniques [1].

In order to make the paper (at least partially) self-contained, the first three sections are devoted to the introduction of the computing models and of the part of field extension and degree theory we are going to use. We have tried to "factor out" of the proofs the parts that are strictly algebraic, in such a way that the reader accustomed with the results we use can skip directly to the heart of the matter.

2. Field extension theory

In this section we gather some definitions and properties about fields of characteristic 0 that shall be used frequently in the sequel (in fact, all we shall say is true of any *perfect* field). The algebraic results quoted here can be found in [8, 13].

Field extensions. Let F be a subfield of E (i.e., E is an *extension* of F), and let $a \in E$. We say that a is *algebraic* over F if there exists a nonzero polynomial $p(x) \in F[x]$ such that $p(a) = 0$, *transcendental* otherwise; if every element of E is algebraic over F, we say that E is an *algebraic extension* of F.

Real closed fields. A field R is (*formally*) *real* if -1 is not a sum of squares. It is *real closed* if it is real but has no (proper) real algebraic extensions. A real closed field has unique ordering, the positive elements in this ordering being precisely the squares. Every ordered real field has a *real closure* (i.e., a maximal real ordered algebraic extension, which is of course real closed), unique up to isomorphism.

Finitely generated extensions. Let $F \subseteq E$ be an extension, and $\alpha_1, \dots, \alpha_r \in E$. The (finitely generated) extension $F \subseteq F(\alpha_1, \dots, \alpha_r)$ is the smallest subfield of E containing F and $\alpha_1, \dots, \alpha_r$.

The primitive element theorem. If $F \subseteq F(\beta_1, \dots, \beta_t)$ is an algebraic extension, then there is a $\beta \in F(\beta_1, \dots, \beta_t)$ such that $F(\beta_1, \dots, \beta_t) = F(\beta)$. In other words, every finitely generated algebraic extension can be thought of as being generated by just a single element, called *primitive*.

Algebraic extensions. Every algebraic extension $F \subseteq F(\alpha)$ induces a surjective homomorphism $F[x] \to F[\alpha]$ (if $F \subseteq E$ is an extension and $\alpha \in F$, then $F[\alpha]$ denotes the ring obtained by evaluating in α the polynomials of $F[x]$), given by the evaluation of x to α. The kernel of this homomorphism is an ideal, generated by an irreducible polynomial $p(x) \in F[x]$, which is separable (i.e., without multiple roots) and can be assumed to be monic without loss of generality, called the *minimum polynomial of* α. The important consequence is that[3] $F[x]/\langle p(x)\rangle \cong F[\alpha]$; moreover, $F[\alpha]$ is a field, and it is thus equal to $F(\alpha)$.

Transcendental extensions. Given a finitely generated extension $F \subseteq F(\alpha_1,\ldots,\alpha_s)$, if there is no nonzero polynomial in s variables and coefficients in F that vanishes when evaluated over $\alpha_1,\ldots,\alpha_s$ (i.e., the α_i's are *algebraically independent*), we have $F(\alpha_1,\ldots,\alpha_s) \cong F(x_1,\ldots,x_s)$, where the latter expression denotes the field of rational functions with s arguments and coefficients in F. Moreover, for every finitely generated extension $F \subseteq F(\alpha_1,\ldots,\alpha_r)$ we can assume without loss of generality that there is an $s \leqslant r$ such that $\alpha_1,\ldots,\alpha_s$ are algebraically independent and the extension $F(\alpha_1,\ldots,\alpha_s) \subseteq F(\alpha_1,\ldots,\alpha_r)$ is algebraic.

Representing finitely generated extensions of $\mathbf{Q}$. By combining the above observations, for every extension $\mathbf{Q} \subseteq \mathbf{Q}(\alpha_1,\ldots,\alpha_r)$, with $\alpha_1,\ldots,\alpha_r \in R$, we have

$$\mathbf{Q}(\alpha_1,\ldots,\alpha_r) = \mathbf{Q}(\alpha_1,\ldots,\alpha_s)[\beta] \cong \mathbf{Q}(x_1,\ldots,x_s)[x]/\langle p(x)\rangle,$$

where $\alpha_1,\ldots,\alpha_s$ are again algebraically independent and β is a primitive element generating the algebraic part of the extension. Thus, every element of $\mathbf{Q}(\alpha_1,\ldots,\alpha_r)$, and in particular, every α_i, has a *coding* as an element of $\mathbf{Q}(x_1,\ldots,x_s)[x]/\langle p(x)\rangle$, given by this isomorphism (for instance, α_i, with $i \leqslant s$, is coded by x_i); moreover, all field operations of $\mathbf{Q}(\alpha_1,\ldots,\alpha_r)$ can be performed symbolically in $\mathbf{Q}(x_1,\ldots,x_s)[x]/\langle p(x)\rangle$, as well as equality tests, given that we know the codings of the elements involved (one just uses the standard polynomial operations modulo $p(x)$); of course, this is not true of order comparisons.

3. Computational models

All our computations are based on a field R, which is always intended to be Archimedean and real closed; for this reason, it will always be identified with an ordered subfield of the reals [13]. We choose the natural topology induced by the *rational open balls*

$$B_\varepsilon(\boldsymbol{r}) = \{\, \boldsymbol{x} \in R^n \mid d(\boldsymbol{r},\boldsymbol{x}) < \varepsilon\},$$

with $\boldsymbol{r} \in \mathbf{Q}^n$ and $0 < \varepsilon \in \mathbf{Q}$; $d(\cdot,\cdot)$ here represents the standard Euclidean metric. Note that if $R \subset \mathbf{R}$, this topology is totally disconnected.

3.1. The finite-dimensional BSS model

A *finite-dimensional BSS machine* M over R consists of three spaces: the input space $\bar{I} = R^l$, the output space $\bar{O} = R^m$ and the state space $\bar{S} = R^n$, together with a finite

[3] $F[x]/\langle p(x)\rangle$ denotes the ring of polynomials modulo $p(x)$, which turns out to be a field iff $p(x)$ is irreducible.

directed connected graph with node set $\bar{N}=\{1,2,\dots,N\}$ $(N>1)$ divided into four subsets: *input*, *computation*, *branch* and *output* nodes.

Node 1 is the only *input node*, having fan-in 0 and fan-out[4] 1; node N is the only *output node*, having fan-out 0. They are endowed with linear functions with integer coefficients (named $I(\cdot)$ and $O(\cdot)$), mapping, respectively, the input space to the state space and the state space to the output space. Any other node $q\in\{2,3,\dots,N-1\}$ can be of the following types:

(1) a *computation node*: in this case, q has fan-out 1 and there is a componentwise rational function $g_q:\bar{S}\to\bar{S}$ associated with it;

(2) a *branching node*: in this case, q has fan-out 2 and its two (distinguished) successors are $\beta^-(q)$ and $\beta^+(q)$; branching on $-$ or $+$ will depend upon whether or not the first coordinate of the state space is negative.[5]

The computation of M on $\boldsymbol{a}\in R^n$ starts from node 1, with state space set to $I(\boldsymbol{a})$, and proceeds as follows: at a computation node q we apply g_q to the state space and move to the unique next node; at a branching node we move to the "minus" or "plus" successor, depending on whether $x_1<0$ or not; we halt when we reach node N. The set of all inputs on which M halts is called the *halting set* of M, and it is denoted by Ω_M. A set which is the halting set of some BSS machine is called *semi-decidable*; if moreover its complement is also semi-decidable, we shall say that the set is *decidable*. If $\alpha_1,\dots,\alpha_r\in R$ are the coefficients of the polynomials appearing in the description of M (i.e., the *constants of* M) we let $E_M=\mathbf{Q}(\alpha_1,\dots,\alpha_r)\subseteq R$ be the *extension of* M. If $X\subseteq R^n$ is (semi-)decided by a machine with constants $\alpha_1,\dots,\alpha_r$, we shall simply say that X is *(semi-)decidable using* $\alpha_1,\dots,\alpha_r$.

3.2. Type 2 Turing machines

Since any Archimedean field is isomorphic to a subfield of the reals, its elements are approximable by converging sequences of rationals (by density of $\mathbf{Q}$), and its operations are approximable using rational approximations of the arguments.

In particular, without loss of generality, we can restrict our attention to sequences of dyadic numbers converging exponentially fast, or, again without loss of generality, to the *signed binary digit* representation. Such a representation is given by an infinite string $s\in\{\bar{1},0,1,.\}^\omega$ of the form

$$s=b_nb_{n-1}\cdots b_0.b_{-1}b_{-2}\cdots,$$

where we assume that $b_n\neq 0$ and that the part on the left of the dot does not start with $1\bar{1}$ or $\bar{1}1$. The number $\bar{s}$ represented by s is defined by

$$\bar{s}=\sum_{i=n}^{-\infty} b_i2^i,$$

[4]If q is a node with fan-out 1, then $\beta(q)$ denotes the "next" node in the graph after q.

[5]Note that usually a test with a polynomial is assumed, but the present restriction can be made without loss of generality [2].

where the symbol $\bar{1}$ has value -1 (of course, not all representations will correspond to elements of R unless $R = \mathbf{R}$). For several reasons [7, 6], this representation is particularly suitable for Turing machines, and will be used in order to represent elements of an Archimedean field R as infinite sequences of symbols (to be given as a generalized input to a Turing machine).

The tape of an ordinary Turing machine is nonblank only on a finite number of cells, at any computation stage. Thus, in order to allow elements of R to be taken into consideration, one slightly generalizes the notion of a machine. A (deterministic) Type 2 Turing machine [14] consists of

(1) a finite number of read-only one-way input tapes (possibly none), each containing at the start an *infinite* string belonging to $\{\bar{1}, 0, 1, .\}^{\omega}$ and representing an element of R;
(2) a finite number of conventional read-only one-way input tapes (possibly none), each containing at the start a *finite* string belonging to $\{0, 1\}^*$;
(3) a finite number of write-only one-way output tapes (possibly none), on which the machine is supposed to write representations of elements of R;
(4) some other work tapes, initially blank.

The finite control is defined as usual via a finite set of states and a transition function. The only differences with a standard Turing machine are the possibility of filling completely the input tapes, and of considering nonstopping machines as machines outputting elements of R. A set $X \subseteq R^n$ is (Type 2) Turing semi-decidable iff there is a Type 2 Turing machine M with n input tapes that stops iff the input tapes are filled with signed binary digit representations of the coordinates of an $\boldsymbol{a} \in X$. Note that the definition implies that the halting does not depend on the particular representations chosen. Moreover, since a halting machine reads only a finite portion of the input, there is always an accepted ball around each accepted point; thus, all Type 2 semi-decidable sets are *open*. A function $f : R^n \to R$ is (Type 2) Turing computable iff there is a Type 2 Turing machine M with n input tapes that never stops and writes a representation of $f(\boldsymbol{a})$ on its output tape whenever its input tapes are filled with a representation of $\boldsymbol{a}$.

All notions introduced in this section will be also used relativizing Turing machines to arbitrary oracles.

4. Degrees of real numbers and jumps

A set $A \subseteq \mathbf{N}$ is recursive in $B \subseteq \mathbf{N}$ iff there is an oracle Turing machine that decides membership to A using B as an oracle; this relation is a preorder on the subsets of $\mathbf{N}$, and the equivalence classes induced by this preorder are called *(Turing) degrees of unsolvability* [11]; they are of course a partially ordered set $\mathscr{D}$ (the order relation being denoted by "$\leqslant$"), which possesses finite suprema denoted by "$\vee$"; the bottom element (corresponding to decidable sets) is denoted by $\mathbf{0}$. We write $\operatorname{dg} A$ for the degree of a subset A of $\mathbf{N}$.

Now consider a set $A \subseteq \mathbf{N}$; let μ_A be the least positive integer included in A (1 if A does not contain any positive integer), and let σ_A be either 1 or -1, depending on whether $0 \in A$ or not. Define

$$\rho(A) = \sigma_A \cdot \left(\mu_A - 1 + \sum_{\mu_A < i \in A} 2^{\mu_A - i} \right).$$

It should be clear that, for any nondyadic real number α, there exists exactly one set in $\rho^{-1}(\alpha)$ (which is neither finite nor cofinite), and we define the degree of α, denoted by $\operatorname{dg} \alpha$, as the degree of unsolvability of $\rho^{-1}(\alpha)$ [12, 5, 4]; moreover, we let $\operatorname{dg} \alpha = \mathbf{0}$ for every dyadic rational α. When the distinction is irrelevant, we shall confuse real numbers, subsets of $\mathbf{N}$ and degrees, omitting the map ρ; this will happen particularly when using real numbers as oracles, or when specifying an arbitrary real number of given degree; note that, in particular, it is equivalent to think of a Turing machine as using oracles $\alpha_1, \ldots, \alpha_r$ or the single oracle $\alpha_1 \vee \cdots \vee \alpha_r$.

The last concept we need from degree theory is the notion of a *jump*. Given a degree $\boldsymbol{d} \in \mathscr{D}$, we can consider the set B that encodes the halting of the universal Turing machine relativized to (any set belonging to) $\boldsymbol{d}$; one defines $\boldsymbol{d}' = \operatorname{dg} B$, where $\boldsymbol{d}'$ is called the *jump of* $\boldsymbol{d}$. Note that one has $\boldsymbol{d}' > \boldsymbol{d}$ for all $\boldsymbol{d} \in \mathscr{D}$.

5. Emulating BSS machines

Armed with the mathematical tools described in the previous sections, we now approach our main problem; in particular, this section is devoted to a series of results that show how a Turing machine can, in a (precise) sense, partially emulate a BSS machine.

Recall that a *semi-algebraic set* $X \subseteq R^n$ is the set of points satisfying a finite boolean combination of disequations of the form $p(x_1, \ldots, x_n) \leqslant 0$, with $p(x_1, \ldots, x_n) \in R[x_1, \ldots, x_n]$. The first ingredient of our proofs is the following lemma, which essentially states that the numbers that are transcendental over the coefficients of the polynomials defining a semi-algebraic set are irrelevant in order to establish if the set is empty (or, more generally, if it intersects a ball). This will prove to be very useful in the following, as the halting sets of BSS machines are countable unions of semi-algebraic sets.

Lemma 1. *Let $X \subseteq R^n$ be a semi-algebraic set, and F be the real closure of the extension of $\mathbf{Q}$ finitely generated by the coefficients of the polynomials used in the definition of X. Then $X \cap F^n$ is dense in X.*

Proof. It suffices to prove that every rational open ball $B_\varepsilon(\boldsymbol{r})$ intersecting X also intersects $X \cap F^n$. Let $\alpha_1, \ldots, \alpha_r$ be the irrational coefficients of the polynomials defining X; there clearly exists a formula $\phi(y_1, \ldots, y_r, x_1, \ldots, x_n)$ in the first-order language of ordered fields [9] such that

$$\{\langle a_1, \ldots, a_n \rangle \in R^n \mid \phi(\alpha_1, \ldots, \alpha_r, a_1, \ldots, a_n)\} = X.$$

Consider now the formula

$$\psi(y_1,\dots,y_r)=\exists x_1,\dots,x_n[\langle x_1,\dots,x_n\rangle\in B_\varepsilon(\boldsymbol{r})\wedge\phi(y_1,\dots,y_r,x_1,\dots,x_n)].$$

By Tarski's theorem, the theory of real closed fields has quantifier elimination, so there is a quantifier-free formula $\xi(y_1,\dots,y_r)$ such that

$$\vdash\forall y_1,\dots,y_r[\psi(y_1,\dots,y_r)\Leftrightarrow\xi(y_1,\dots,y_r)].$$

But since $B_\varepsilon(\boldsymbol{r})\cap X\neq\emptyset$ iff $R\models\psi(\alpha_1,\dots,\alpha_r)$ iff $R\models\xi(\alpha_1,\dots,\alpha_r)$, and the truth value of a quantifier-free formula is preserved by restriction, we obtain $F\models\xi(\alpha_1,\dots,\alpha_r)$, whence $F\models\psi(\alpha_1,\dots,\alpha_r)$, i.e., $B_\varepsilon(\boldsymbol{r})\cap X\cap F^n\neq\emptyset$. □

The second ingredient is an algebraic observation: since the extension of a BSS machine M is finitely generated, if we know the minimum polynomial of the extension E_M and the codings of the constants of M we can emulate its behaviour with a Turing machine on any coded input (i.e., on any n-tuple of elements of E_M), given that we provide the binary expansion of the constants as oracles. The process can be pushed further, in fact up to the real closure of E_M.

Lemma 2. *Given a BSS machine M using $\alpha_1,\dots,\alpha_r\in R$ there is a (classical) Turing machine M' with oracle $\alpha_1\vee\cdots\vee\alpha_r$ that can emulate M on any element $\boldsymbol{a}\in R^n$ whose coordinates are algebraic over E_M. The input to M' is given by an n-tuple of pairs $\langle q(x),(a,b)\rangle$, where $q(x)$ is an irreducible monic polynomial of $\mathbf{Q}(x_1,\dots,x_s)[x]$ and the rational interval (a,b) contains a unique real root of $\bar q(x)\in\mathbf{Q}(\alpha_1,\dots,\alpha_s)[x]$, $\bar q(x)$ being obtained by the evaluation $x_i\mapsto\alpha_i$.*

Proof. First of all, denoting by $\boldsymbol{a}\in R^n$ the (unique) vector specified by the input pairs, the primitive element theorem tells us that there is (using the notation of Section 2) a $\gamma\in R$ such that

$$F=\mathbf{Q}(\alpha_1,\dots,\alpha_r)[a_1,\dots,a_n]=\mathbf{Q}(\alpha_1,\dots,\alpha_s)[\beta][a_1,\dots,a_n]=\mathbf{Q}(\alpha_1,\dots,\alpha_s)[\gamma];$$

in order to be able to perform computations in F, by means of Gröbner base algorithms [1] M' obtains from $p(x)$ (i.e., the minimum polynomial appearing in the coding of E_M, which is hardwired in M') and from the polynomials $q_1(x),\dots,q_n(x)$ coefficients $c_1,\dots,c_n\in\mathbf{Q}(x_1,\dots,x_s)$ such that

$$\gamma=\beta+\bar c_1a_1+\cdots+\bar c_na_n.$$

(again, overlining denotes the evaluation $x_i\mapsto\alpha_i$ for $1\leqslant i\leqslant s$), a polynomial $m(x)\in\mathbf{Q}(x_1,\dots,x_s)[x]$ such that $\bar m(x)$ is the minimum polynomial of γ and also polynomials $p_\beta(x)$, $p_{a_1}(x),\dots,p_{a_n}(x)\in\mathbf{Q}(x_1,\dots,x_s)[x]$ such that $\bar p_\beta(\gamma)=\beta$ and $\bar p_{a_i}(\gamma)=a_i$. In other words, M' recodes all constants of M (by substituting x with $p_\beta(x)$) and all inputs in the form given by the isomorphism

$$F=\mathbf{Q}(\alpha_1,\dots,\alpha_s)[\gamma]\cong\mathbf{Q}(x_1,\dots,x_s)[x]/\langle m(x)\rangle.$$

As remarked in Section 2, M' can now use this coding to perform exact symbolic computations and equality tests in F; moreover, whenever M' wants to establish non-negativity of an element, after deciding equality with 0 by a symbolic check it can approximate its evaluation in F, using the oracle and (Type 2) root extraction (see, e.g., [6]) in order to obtain the digits of $\alpha_1,\dots,\alpha_r$, β and, finally, γ (polynomials are continuous and Type 2 computable). Since the "less-than" relation is Type 2 decidable provided that the inputs are not equal, the sign check always terminates. But this is exactly all M' needs to emulate the behaviour of M. □

The previous lemma allows us to emulate a BSS machine M on an arbitrary tuple of elements of R algebraic over E_M; to use Lemma 1, however, we need to be able to cover the entire real closure of E_M, and this can be done by emulating M on *all* such tuples, an idea which is the core of the next lemma.

Lemma 3. *Let* $\alpha_1,\dots,\alpha_s\in R$ *be algebraically independent, so* $\mathbf{Q}(x_1,\dots,x_s)\cong\mathbf{Q}(\alpha_1,\dots,\alpha_s)$. *Then there is a (classical) Turing machine M with oracle* $\alpha_1\vee\cdots\vee\alpha_s$ *enumerating a list of all elements of R algebraic over* $\mathbf{Q}(\alpha_1,\dots,\alpha_s)$ *represented as pairs* $\langle q(x),(a,b)\rangle$, *where* $q(x)$ *is an irreducible monic polynomial of* $\mathbf{Q}(x_1,\dots,x_s)[x]$ *and the rational interval* (a,b) *contains a unique real root of* $\bar{q}(x)\in\mathbf{Q}(\alpha_1,\dots,\alpha_s)[x]$.

Proof. We can easily enumerate all monic polynomials in $\mathbf{Q}(x_1,\dots,x_s)[x]$ and eliminate all instances of reducible polynomials (this is easy – by the Gaussian Lemma [1], in order to do so we just need to be able to factor in $\mathbf{Z}[x_1,\dots,x_s]$). Then we can use standard approximation techniques to produce a finite list of intervals, each containing exactly one root: for instance, we can firstly count the real roots (e.g., using Tarski's theorem), and then use the known rational lower bound δ on the distance between roots (obtained from the discriminant of the polynomial) to exhaustively search a list of intervals of the form $[k\delta,(k+1)\delta]$ containing all roots. For more sophisticated methods, see [1]. □

We can finally apply the previous lemmata; our first result is apparently not related to our main problem, but it will prove to be useful, and it is of independent interest:

Theorem 4. *Let* $\alpha_1,\dots,\alpha_r\in R$. *There is a Turing machine with oracle* $\alpha_1\vee\cdots\vee\alpha_r$ *that receives in input a BSS machine*[6] *M using* $\alpha_1,\dots,\alpha_r$ *and stops iff* $\Omega_M\neq\emptyset$.

Proof. If $\Omega_M\neq\emptyset$, then there is an $\boldsymbol{a}\in R^n$ that follows an acceptance path. Thus, $\boldsymbol{a}$ lies in the semi-algebraic set defined by conjunction of the disequations tested along the path. By Lemma 1, we obtain that a tuple of elements $\tilde{\boldsymbol{a}}$ of the real closure of E_M follows the same path. Thus, a Turing machine dovetailing the enumeration of all n-tuples of elements of R algebraic over E_M (represented as in Lemma 3) with

[6] Note that, of course, the machine M must be described in symbolic form (i.e., by suitably naming the constants).

the emulation of M given by Lemma 2 will certainly stop as soon as it emulates M on $\tilde{\boldsymbol{a}}$. □

By combining two BSS machines in such a way to halt on the intersection of their halting sets, we obtain

Corollary 5. *Let $\alpha_1,\ldots,\alpha_r \in R$. There is a Turing machine M with oracle $\alpha_1 \vee \cdots \vee \alpha_r$ that receives in input pairs $\langle M_1, M_2\rangle$ of BSS machines using $\alpha_1,\ldots,\alpha_r$ and stops iff $\Omega_{M_1} \cap \Omega_{M_2} \neq \emptyset$.*

But, more interestingly, since rational (open or closed) balls are BSS decidable, we also obtain the following

Corollary 6. *Let $\alpha_1,\ldots,\alpha_r \in R$. There is a Turing machine with oracle $\alpha_1 \vee \cdots \vee \alpha_r$ that receives in input a BSS machine M using $\alpha_1,\ldots,\alpha_r$ and a rational ball, and stops iff the rational ball meets Ω_M.*

Note that we cannot hope to semi-decide (relatively to $\alpha_1 \vee \cdots \vee \alpha_r$) *any* of the remaining related questions, i.e., whether (i) $\Omega_M = \emptyset$, (ii) $\Omega_M = R$, (iii) $\Omega_M \neq R$, whether a rational ball is entirely (iv) contained in Ω_M or (v) Ω_M^c, or whether a rational ball (vi) intersects Ω_M^c. This can be easily seen by suitably coding subsets of $\mathbf{N}$ in R: a positive answer for any of the first three questions would imply semi-decidability of the corresponding problems on $\mathbf{N}$ for classical Turing machines (by building a BSS machine that suitably filters out, in case (i), or includes all, in cases (ii) and (iii), noninteger elements of R and then emulates a given Turing machine), while the set of balls of radius $\frac{1}{4}$ centered around the naturals belonging to $(\alpha_1 \vee \cdots \vee \alpha_r)'$ shows that (v) and (vi) are not semi-decidable, either. A proof of (iv), however, needs some more tools, and it is postponed to Section 6.

We remark also that a consequence of the previous corollary is that there is a Turing machine *with oracle* $(\alpha_1 \vee \cdots \vee \alpha_r)'$ that *decides* whether a rational ball is contained in Ω_M^c. This observation suggests our main theorem.

Theorem 7. *Given a set $X \subseteq R^n$ BSS decidable using $\alpha_1,\ldots,\alpha_r \in R$, there is a Turing machine M with oracle $(\alpha_1 \vee \cdots \vee \alpha_r)'$ that decides whether a rational ball is contained in X. In particular, the interior and the exterior[7] of X are Type 2 semi-decidable with oracle $(\alpha_1 \vee \cdots \vee \alpha_r)'$.*

Proof. The machine M just uses the oracle to decide whether the machine of Corollary 6 (applied to the BSS machine semi-deciding X^c) will stop. Then, we can use M to decide whether an element of a sequence of increasingly smaller rational open balls around the input is entirely accepted, thus Type 2 semi-deciding the interior of X. For the exterior, one just applies the first part to X^c. □

[7] We recall that the set $(R^n \backslash X)^\circ = R^n \backslash \bar{X}$ is called the *exterior of X*.

The previous theorem shows the interplay of topological, algebraic and logical aspects of the relation between BSS and Type 2 decidability; in the next section we shall see that the jump is actually necessary, and that moreover the theorem cannot be extended to BSS semi-decidable sets. We also obtain the following corollary.

Corollary 8. *Given an open set $X \subseteq R^n$ BSS decidable using $\alpha_1, \ldots, \alpha_r \in R$, there is a Type 2 Turing machine M with oracle $(\alpha_1 \vee \cdots \vee \alpha_r)'$ that semi-decides X.*

Note that in case X and $R^n \setminus X$ are both open (this of course cannot happen on the real numbers, except for trivial cases) Theorem 7 states that BSS decidable sets become Type 2 decidable sets, modulo the knowledge of an additional jump. Moreover, the first property claimed in Theorem 7 is in fact *strictly stronger* than Type 2 semi-decidability.

6. Equality is a jump

The first step in order to make the title of this section precise is noting that we can further extend the notion of degree to subsets of R^n; more precisely, we shall associate to each subset of R^n a set of degrees that, in a sense, represents the "hardness" of enumerating a sequence of open balls covering the set itself. Let $\mathscr{B}$ denote the set of rational open balls of R^n; clearly, $\mathscr{B}$ may be identified with a subset of $\mathbf{N}$, using a suitable coding, and in the sequel we shall often use this coding without mention. For each $X \subseteq R^n$, we let $\mathscr{B}(X)$ be the set of rational open balls included in X, i.e.,

$$\mathscr{B}(X) = \mathscr{B} \cap 2^X.$$

The union of all such balls covers the interior of X, but this might as well be true also of other (proper) subsets of $\mathscr{B}(X)$. For this reason, we define

$$\mathscr{C}(X) = \left\{ S \subseteq \mathscr{B}(X) \;\middle|\; \bigcup_{B \in S} B = X^\circ \right\};$$

each element of $\mathscr{C}(X)$ is a set of rational open balls that cover the interior of X: note that $\mathscr{C}(\cdot)$ is injective on open sets. Now, we define the (*extended*) *degree of X* as

$$\operatorname{dg} X = \{\boldsymbol{d} \in \mathscr{D} \mid \text{there is an } S \in \mathscr{C}(X) \text{ which is r.e. in } \boldsymbol{d}\} \subseteq \mathscr{D},$$

i.e., as the set of degrees in which some covering of X° is recursively enumerable. It is worth noticing that

Proposition 9. *The following holds for each $X \subseteq R^n$:*
(1) $\operatorname{dg} X$ *is upward closed;*
(2) $\operatorname{dg} X = \operatorname{dg} X^\circ$.

The upward closed subsets of $\mathscr{D}$, ordered by reverse inclusion, form a poset of their own, and $\mathscr{D}$ may be embedded in it by mapping each degree to the cone above it:

$$\boldsymbol{d} \mapsto \{\boldsymbol{e} \in \mathscr{D} \mid \boldsymbol{d} \leqslant \boldsymbol{e}\}.$$

For this reason, in the following we shall use $\leqslant$ to denote the (reverse inclusion) order on upward closed subsets of $\mathscr{D}$, and identify each degree with the corresponding cone.

Now, by slightly modifying the construction of a (BSS semi-decidable, Type 2 non-semi-decidable) set proposed by Vasco Brattka (and used in [3] in order to prove the existence of BSS non-locally-time-bounded computations) we build a set that proves the first half of our main statement:

Theorem 10. *For every $\alpha_1,\dots,\alpha_r \in R$, there is a (regular) open set $X \subseteq R$, decidable by a BSS machine using $\alpha_1,\dots,\alpha_r$, such that*

$$(\alpha_1 \vee \cdots \vee \alpha_r)' \leqslant \mathrm{dg}\, X.$$

Proof. Consider a function $f : \mathbf{N} \to \mathbf{N}$ (recursive in $\alpha_1 \vee \cdots \vee \alpha_r$) enumerating the halting set relativized to $\alpha_1 \vee \cdots \vee \alpha_r$ (i.e., the set of Cantor-coded pairs $\langle j, n\rangle$ such that the jth Turing machine using $A \in \mathrm{dg}^{-1}(\alpha_1 \vee \cdots \vee \alpha_r)$ as oracle halts on input n). We first want to code A in the set X, and we do this by using suitable open balls in the negative part:

$$X_1 = \bigcup_{i \in A} B_{1/4}(-2i-1) \cup \bigcup_{i \notin A} B_{1/4}(-2i-2).$$

On the positive side, we build the set as follows: for each natural i, we add larger and larger open balls approximating i from below as long as we do not find i in the enumeration produced by f. If $i \in f(\mathbf{N})$, we complete the process by adding a small ball to the immediate left of i, and leave an entire closed subinterval of $(i-1, i)$ out of X; conversely, if $i \notin f(\mathbf{N})$ the interval will be entirely covered by X. More precisely, define $X_2 \subseteq R$ as follows:[8]

$$X_2 = \bigcup_{n \in \mathbf{N}} \left[\bigcup_{i \notin f([n])} \left(i-1, i - \frac{1}{2+n} \right) \cup \bigcup_{i \in f([n])} \left(i - \frac{1}{5 + \min f^{-1}(i)}, i \right) \right] \cup \mathbf{N},$$

and let $X = X_1 \cup X_2$. The set X is decided by the following BSS machine:

BSS machine $M(x : R)$;
  /* Decides the set X. */
var i, n : integer;
begin
  if $x \leqslant -1$ **return**($x \in X_1$);
  if $x \in \mathbf{N}$ **return**(1);
  $i \leftarrow \lceil x \rceil$;
  $n \leftarrow 0$;

[8] Here and in the sequel we use the standard notation $[n]$ for the set $\{0, 1, \dots, n-1\}$.

forever
  if $x \in \left(i-1, i-\dfrac{1}{2+n}\right)$ **return**(1);
  if $f(n)=i$ **exit**;
  $n \leftarrow n+1$
loop
if $x \in \left(i-\dfrac{1}{5+n}, i\right)$ **return**(1);
return(0)
end

Note that we used membership to X_1 as a subroutine, as it is trivial to build a BSS machine that decides X_1 using $\alpha_1, \ldots, \alpha_r$; a detailed correctness proof for M has been given in [3].

To prove the statement, we have to show that, for every $S \in \mathscr{C}(X)$ and every degree $\boldsymbol{d} \in \mathscr{D}$, if S is r.e. in $\boldsymbol{d}$ then $(\alpha_1 \vee \cdots \vee \alpha_r)' \leqslant \boldsymbol{d}$. Suppose that we have an S and a $\boldsymbol{d}$ with the previous property; first notice that we can decide membership to A using the machine with oracle $\boldsymbol{d}$ enumerating S (for each $i \in \mathbf{N}$, either $-2i-1$ or $-2i-2$ belong to X, and the former happens iff $i \in A$). If we want to decide whether $i \in (\alpha_1 \vee \cdots \vee \alpha_r)'$ (i.e., whether $i \in f(\mathbf{N})$) or not, we operate as follows: we first enumerate S until we find an open ball B including i. Then, we compute $f(n)$ for increasing values of n (this can be done using the fact that membership to A is decidable); eventually, either we shall find an n such that $f(n)=i$, or the ball $(i-1, i-1/(2+n))$ will intersect B (and thus $(i-1, i) \subseteq X$, which implies $i \notin f(\mathbf{N})$). □

Thus, there exist BSS decidable open sets whose degree is strictly greater than the degree of the constants used by the machine deciding them. As a matter of fact, now we prove that the above inequality really boils down to an equality, thus showing that the degree of a BSS decidable set is at most a jump above the constants, and that equality can actually be reached; in other words, the possibility for a BSS machine to perform exact equality tests imposes a jump in the degree of unsolvability (of the constants of the machine), or, in a slogan, "equality is a jump".

Theorem 11. *Let $X \subseteq R^n$ be a set BSS decidable using $\alpha_1, \ldots, \alpha_r$. Then*

$$\operatorname{dg} X \leqslant (\alpha_1 \vee \cdots \vee \alpha_r)'$$

and there are sets for which equality holds.

Proof. For the first part, we just have to show that there is an $S \in \mathscr{C}(X)$ which is r.e. in $(\alpha_1 \vee \cdots \vee \alpha_r)'$: but the whole set $\mathscr{B}(X)$ satisfies the last statement, by applying Theorem 7 (which actually tells us that $\mathscr{B}(X)$ is even *recursive* in $(\alpha_1 \vee \cdots \vee \alpha_r)'$); the second part is Theorem 10. □

As suggested by one of the anonymous referees, if we drop the openness and decidability hypotheses and just require X to be BSS semi-decidable we can prove even more.[9]

Theorem 12. *For every $\alpha_1, \ldots, \alpha_r \in R$, there is a (regular) closed set $Y \subseteq R$, semi-decidable by a BSS machine using $\alpha_1, \ldots, \alpha_r$, such that*

$$(\alpha_1 \vee \cdots \vee \alpha_r)'' \leqslant \mathrm{dg}\, Y.$$

Proof. Since $(\alpha_1 \vee \cdots \vee \alpha_r)''$ belongs to $\Sigma_2^{0, \alpha_1 \vee \cdots \vee \alpha_r}$ (i.e., to the second level of the arithmetical hierarchy relativized to $\alpha_1 \vee \cdots \vee \alpha_r$ – see [10]), there is a set $A \subseteq \mathbf{N}^2$ which is recursively enumerable with oracle $\alpha_1 \vee \cdots \vee \alpha_r$ such that $n \notin (\alpha_1 \vee \cdots \vee \alpha_r)''$ iff for all k we have $\langle n, k\rangle \in A$. Let $f : \mathbf{N} \to \mathbf{N}^2$ be a function recursive in $\alpha_1 \vee \cdots \vee \alpha_r$ enumerating A; then, the set

$$Z = \mathbf{N} \cup \bigcup_{m \in \mathbf{N}} \bigcup_{\{n\} \times [k] \subseteq f([m])} \left[n - 1, n - \frac{1}{2+k} \right]$$

is clearly a regular closed BSS semi-decidable subset of R, but i belongs to Z° iff $i \notin (\alpha_1 \vee \cdots \vee \alpha_r)''$: in other words, $(\alpha_1 \vee \cdots \vee \alpha_r)''$ is co-recursively enumerable in every degree $\boldsymbol{d} \geqslant \mathrm{dg}\, Z$ (the proof is identical to that of Theorem 10).

Consider now the set X of Theorem 10, and note that $\mathrm{dg}\, \bar{X} = \mathrm{dg}(\bar{X})^\circ = \mathrm{dg}\, X \geqslant (\alpha_1 \vee \cdots \vee \alpha_r)'$ by regularity of X and by Proposition 9. The set $\bar{X}$ (which is still BSS decidable – just close all intervals involved in its definition) can be easily mapped into the interval $(-4, -2)$ by the homeomorphism $x \mapsto x/(1 + |x|) - 3$, which is BSS computable and preserves and reflects degrees of sets (being a composition of moduli and rational operations, its restriction to the rationals can be computed by a Turing machine). The set Y obtained by joining Z, the image of $\bar{X}$ and the points -2 and -4 is thus regular closed and BSS semi-decidable.

We remark that $\mathrm{dg}\, Y = \mathrm{dg}\, X \vee \mathrm{dg}\, Z$. Indeed, every enumeration of balls covering Y° can be recursively turned into an enumeration for X (or Z) simply by discarding those balls lying at the right (left, respectively) of -2, and possibly using the inverse homeomorphism $y \mapsto |y+3|/(1 - |y+3|)$. The other inequality is proved analogously.

Now, for every degree $\boldsymbol{d} \geqslant \mathrm{dg}\, Y$ we have $\boldsymbol{d} \geqslant \mathrm{dg}\, X \geqslant (\alpha_1 \vee \cdots \vee \alpha_r)'$ by Theorem 10, so $(\alpha_1 \vee \cdots \vee \alpha_r)''$ is recursively enumerable in $\boldsymbol{d}$ (since it is recursively enumerable in $(\alpha_1 \vee \cdots \vee \alpha_r)'$). Moreover, $\boldsymbol{d} \geqslant \mathrm{dg}\, Z$ and thus $(\alpha_1 \vee \cdots \vee \alpha_r)''$ is also co-recursively enumerable in $\boldsymbol{d}$, hence the thesis. □

The "packing technique" used in the previous proof (two subsets of R were joined without overlapping by homeomorphically mapping one of them into a rational interval) can be used in a very general way so as to join a finite number of subsets, obtaining a new set whose degree is the join of the original degrees. The interesting point is

[9] In fact, Theorem 12 also shows that Conjecture 1 of [3] is false, by using Theorem 13 and Corollary 4 of [3].

that the homeomorphism is both BSS and Type 2 computable, and its trace on the rationals is recursive. Thus, not only topological, but also computational properties are preserved.

The previous results should be considered as against the following one:

Theorem 13. *If the interior of* $X \subseteq R^n$ *is Type* 2 *semi-decidable using an oracle* (*with degree*) $\boldsymbol{d}$, *then*

$$\operatorname{dg} X \leqslant \boldsymbol{d}.$$

Proof. Just relativize the standard proof of the fact that a Type 2 semi-decidable set can be expressed as the union of a recursively enumerable set of rational open balls. In other words, dovetail the emulation of the machine semi-deciding X on every rational and, on halting, output the corresponding open ball. □

Finally, the set X of Theorem 10 can be used to give the impossibility proof anticipated in the previous section:

Theorem 14. *Given* $\alpha_1, \ldots, \alpha_r \in R$, *there is no Turing machine with oracle* $\alpha_1 \vee \cdots \vee \alpha_r$ *that receives in input a BSS machine* M *using* $\alpha_1, \ldots, \alpha_r$ *and a rational ball, and semi-decides whether the ball is included in* Ω_M.

Proof. Simply note that by giving as input to the Turing machine a BSS machine semi-deciding the set X of Theorem 10 and the interval $(i - \frac{3}{4}, i)$ or its closure we could decide whether i belongs to $(\alpha_1 \vee \cdots \vee \alpha_r)'$. □

7. Conclusions and open problems

Several remarks are in order. First of all, we want to stress that our existence theorems are by their very nature *nonconstructive*. The polynomial $p(x)$ used in Lemma 2 depends uniquely on the constants of the BSS machine, but of course there is no reasonable notion of "constructivity" that allows one to derive $p(x)$ from *arbitrary* constants. More formally, there is no way of turning the Turing machine of Theorem 4 into a Type 2 Turing machine accepting $\alpha_1, \ldots, \alpha_r$ as additional inputs – this would allow one to build a Type 2 Turing machine semi-deciding the closed set $[0, \infty)$ by providing as input the (input-independent) BSS machine halting exactly when its only parameter is nonnegative. However, there are also sensible ways of expressing constants that make $p(x)$ computable (for instance, this happens when the constants are algebraic and given through their minimum polynomials and through rational intervals, as done for the inputs in the statement of Lemma 2).

The reader can certainly notice that Corollary 8 does not intuitively appear to be "the best possible". In particular, one would like to weaken the hypothesis on the set X to BSS semi-decidability. However, the interplay between the topological and recursive properties of X and X^c make this goal very hard; indeed, we have not been able to prove or disprove the following statement:

every open set BSS semi-decidable using $\alpha_1,\dots,\alpha_r$ is Type 2 semi-decidable with oracle $(\alpha_1 \vee \cdots \vee \alpha_r)'$.

In fact, a (more informative, yet less appealing) title for this paper could have been *Equality is a Jump on BSS Decidable Sets*; should the above conjecture be true, it could be turned into *Equality is a Jump on Open BSS Semi-Decidable Sets*. It is however clear that the *topological* gap forced by equality is incomparable with no matter what notion of degree, the latter being a computational, rather than topological, invariant.

Another interesting open question concerns the nature of extended degrees: when does $\mathrm{dg}X$ happen to be a *cone*? In other words, when can we associate to X a *Turing degree*, instead of an upward closed subset of $\mathscr{D}$?

As it stands, the question mixes inextricably topological and degree-theoretical properties; however, in the particular case of the reals, *one can factor out the topological part*. This is due to the fact that the degree of a set $X \subseteq \mathbf{R}^n$ is actually the set of degrees in which *the set of all closed balls contained in* X° is recursively enumerable. This equivalence can be easily shown on one side by compactness, and on the other by the fact that $\mathbf{R}$ is T_3. Hence, at least in this case, the question reduces to the following purely degree-theoretical problem:

given a set $A \subseteq \mathbf{N}$, does the set of degrees in which A is r.e. always contain a minimum?

This problem, as far as we know, is currently open.

We conclude with a counterexample showing that, in general, this "factorization" cannot be performed for an arbitrary subfield $R \subset \mathbf{R}$. Assume R is not *Turing closed* [4], i.e., there is a Type 2 Turing machine that accepts inputs $\alpha_1,\dots,\alpha_r \in R$ and outputs a real number $\alpha \notin R$ (this happens iff the set $\{\mathrm{dg}\,x \mid x \in R\}$ is not an ideal of $\mathscr{D}$). Hence, assuming without loss of generality that $\alpha \in (0,1)$, there is a Turing machine with oracle $\alpha_1 \vee \cdots \vee \alpha_r$ that enumerates rationals $0 < l_0 < l_1 < \cdots < \alpha$ ($\alpha < \cdots < u_1 < u_0 < 1$) approximating α from below (above, respectively). Let f and X_1 be as in the proof of Theorem 10; then, the open set

$$Y = \bigcup_{n\in\mathbf{N}} \bigcup_{i \notin f([n])} (i, i + l_n) \cup \bigcup_{i,n\in\mathbf{N}} (i - 1 + u_n, i + l_0) \cup X_1$$

satisfies $\mathrm{dg}\,Y \leqslant \alpha_1 \vee \cdots \vee \alpha_r$, but it has "closed balls degree" greater than or equal to $(\alpha_1 \vee \cdots \vee \alpha_r)'$, since the question $i \notin (\alpha_1 \vee \cdots \vee \alpha_r)'$ reduces to $[i, i+1] \subseteq Y$, and the question $i \in (\alpha_1 \vee \cdots \vee \alpha_r)'$ can be answered by emulation using X_1.

Acknowledgements

Several discussions with Giorgio Donnini and Davide Ferrario helped to streamline the mathematics of this paper. We are moreover indebted to the participants of the

Workshop on *Computability and Complexity in Analysis*, held at Schloß Dagstuhl in April 1997, for several useful inputs. One of the anonymous referees helped with many useful comments; among them, he suggested the construction of the set Z in the proof of Theorem 12.

References

[1] T. Becker, V. Weispfenning, Gröbner Bases, Graduate Texts in Mathematics, vol. 141, Springer, Berlin, 1993, in cooperation with Heinz Kredel.
[2] L. Blum, M. Shub, S. Smale, On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines, Bull. Amer. Math. Soc. (N.S.) 21 (1989) 1–46.
[3] P. Boldi, S. Vigna, δ-uniform BSS machines, J. Complexity 14 (1998) 234–256.
[4] P. Boldi, S. Vigna, The Turing closure of an Archimedean field, Theoret. Comput. Sci., to appear.
[5] K.-I. Ko, Reducibilities on real numbers, Theoret. Comput. Sci. 31 (1–2) (1984) 101–123.
[6] K.-I. Ko, Complexity Theory of Real Functions, Progress in Theoretical Computer Science, Birkhäuser, Basel, 1991.
[7] C. Kreitz, K. Weihrauch, Theory of representations, Theoret. Comput. Sci. 38 (1985) 35–53.
[8] S. Lang, Algebra, Addison-Wesley, Reading, MA, 1971.
[9] D. Marker, M. Messmer, Anand Pillay, Model Theory of Fields, of Lecture Notes in Logic, vol. 5, Springer, Berlin, 1996.
[10] P. Odifreddi, Classical Recursion Theory, vol. 125, of Studies in Logic and the Foundations of Mathematics, North-Holland, Amsterdam, 1989.
[11] J.R. Shoenfield, Degrees of Unsolvability, North-Holland, Amsterdam, 1971.
[12] R.I. Soare, Recursion theory and Dedekind cuts, Trans. Amer. Math. Soc. 140 (1969) 271–294,
[13] B.L. van der Waerden, Algebra, 8th ed., Springer, Berlin, 1971.
[14] K. Weihrauch, Type 2 Recursion Theory, Theoret. Comput. Sci. 38 (1) (1985) 17–33.