On the Computing Power of $+,-,$ and $\times$

Marcello Mamino
Laboratoire d’Informatique de l’École Polytechnique (LIX)
Bâtiment Alan Turing
1 rue Honoré d’Estienne d’Orves, Campus de l’École Polytechnique, 91120 Palaiseau, France
Email: mamino@lix.polytechnique.fr

Abstract—Modify the Blum-Shub-Smale model of computation replacing the permitted computational primitives (the real field operations) with any finite set $B$ of real functions semialgebraic over the rationals. Consider the class of Boolean decision problems that can be solved in polynomial time in the new model by machines with no machine constants. How does this class depend on $B$? We prove that it is always contained in the class obtained for $B = \{ +, -, \times \}$. Moreover, if $B$ is a set of continuous semialgebraic functions containing $+$ and $-$, and such that arbitrarily small numbers can be computed using $B$, then we have the following dichotomy: either our class is $\emptyset$ or it coincides with the class obtained for $B = \{ +, -, \times \}$.

I. INTRODUCTION

In this work, we study the power of computation over the real numbers to decide classical Boolean problems. As opposed to discrete domains, there is currently no universally accepted natural point of view on computation over the reals, most of the existing models being roughly divided in two groups. On the one hand, if we regard a computation over the reals as a process of approximation to be carried out through discrete means, then we are into the tradition of computable analysis and the bit model. On the other hand, we can forgo some extent of realism, and consider theoretical machines capable of directly manipulating real numbers with unbounded precision: in this case, we are looking at models such as the Blum-Shub-Smale model and real random access machines. In the context of computational complexity, adopting the second point of view means, usually, to fix a finite basis of primitive operations that a machine can perform on real numbers, and fixing some prescribed, often unitary, cost for such operations: in short, a rigorous form of counting flops. In general, in these models, machines compute real functions of real inputs. There is, however, a trend to bring complexity in the Blum-Shub-Smale model back into contact with classical discrete complexity by studying the power of Boolean parts: the Boolean part of a complexity class over the reals is obtained by restricting the input and output of the corresponding machines to Boolean values (the idea dates back to [Goo94] and [Ko93], the reader may find more information in §2.2 of the book [BCSS98], which is also the reference for the Blum-Shub-Smale model, additional bibliography can be found in [ABKM09]). In this work, we will explore how the Boolean parts of real complexity classes change by varying the set of primitive operations. In particular, we are interested in machines performing various sets of semialgebraic operations at unit cost.

One point of criticism to the Blum-Shub-Smale model (raised, for example, in [Bra05], [BC06]) is that the only computable functions are piecewise polynomial. In short: why should $\sqrt{\pi}$ not be computable? Consider the Sum of Square Roots problem – compare two sums of square roots of positive integers – which is important in computational geometry due to ties with the Euclidean Travelling Salesman Problem (GJ76). This problem is trivially solvable in polynomial time by a real Turing machine with primitives $+,-,$ and $\sqrt{\pi}$ (we always assume to have equality and comparison tests), and, in fact, it can be solved in polynomial time also by the usual real Turing machine (i.e. with primitives for rational functions), but the result requires a clever argument (Iw92).

Are we witnessing a coincidental fact, or is there a deeper relation between the ad-hoc set of primitives $\{ +,-,\sqrt{\pi}\}$ and the one chosen by Blum, Shub, and Smale $\{ +,-,\sqrt{\pi}\}$? For instance, as long as we restrict our attention to discrete decision problems and, say, polynomial time, it is conceivable that adding $\sqrt{\pi}$ to the basic functions of the real Turing machine (or replacing $\times$ and $\div$ with $\sqrt{\pi}$) may not increase (or alter) its computational power—or, more precisely, the set of discrete decision problems that it can decide in polynomial time. It is a corollary of our results that the question we started with has, in fact, an answer: for Boolean problems, the real Turing machine doesn’t need the primitive $\sqrt{\pi}$, because it can simulate it.

The study of complexity over arbitrary structures has been initiated by Goode in [Goo94] and continued by many, see for instance Poizat’s book [Po95] (also, in the context of recursion theory, there has been previous work: see [Ers81], [EM92]). This line of research focused mainly on questions such as $P = NP$ inside different structures, or classes of structures; i.e. considering equivalences or separations relativized to various structures more or less in the same spirit as one relativizes to various oracles. Adding Boolean parts to the mix, we gain the ability to meaningfully compare complexity classes across structures. Several problems that are complete for the Boolean part $BP^0_2(P^0_2)$ of the class of problems solvable in polynomial time by real Turing machines without machine constants have been recently studied by Allender, Bürgisser, Kjeldgaard-Pedersen, and Miltersen [ABKM09]. One of the

Date: 201-1-2012
The author has received funding from the European Research Council under the European Community’s Seventh Framework Programme (FP7/2007-2013 Grant Agreement no. 257039).
BP(P^0_2)-complete problems identified in [ABKM09], called by them the Generic Task of Numerical Computation, is offered as a prototype for problems which are hard for numerical, as opposed to combinatorial, reasons; suggesting that the notion of BP(P^0_2)-hardness may have practical value, to prove intractability of numerical problems, much as NP-hardness is used for combinatorial problems. In fact, recent research adopts precisely this point of view to assess the complexity of fixed point problems [EY10], and of semidefinite programming [TV08]. We are therefore encouraged to investigate how the (analogue of) the class BP(P^0_2) changes when varying the computational basis, both as a means to evaluate how generic the GTNC really is, and as a way to build up a toolbox of problems hard or complete for BP(P^0_2).

The aim of this paper is to prove that the computing power of any finite set of real functions semialgebraic over Q – examples of which are the square root, a function computing the real and imaginary parts of the roots of a seventh degree polynomial given by its coefficients, or the euclidean distance of two ellipsoids in R^3 represented using, say, positive semidefinite matrices – does not exceed the computing power of +, −, and ×. We also prove, under reasonable technical hypotheses, that a basis of functions semialgebraic over Q either solves in polynomial time precisely the discrete problems in BP(P^0_2), or precisely P. For instance, to go back our little example, we can answer our own question very precisely: the discrete problems that the computational bases \{+, −, √x\}, \{+, −, x, ÷\}, and \{+, −, x, ÷, √x\} can solve in polynomial time are precisely the same.

We will, now, spend a few words on the technical setting of our results. Among the BP(P^0_2)-complete problems identified in [ABKM09] there is the problem PosSLP: to decide whether a given circuit with gates for 0, 1, +, −, × and no input gates represents a positive number. Clearly, the completeness of PosSLP for BP(P^0_2) can be generalized to any basis B and the corresponding polynomial time class. In other words, one can consider a Boolean (⊂ {0, 1}) language to be efficiently decidable using B, when it is decidable in polynomial time by a machine over the reals with basic operations B. Taking a different approach, one may say that a language is efficiently decidable using B if it is polynomial time Turing reducible to PosSLP(B) – i.e. PosSLP with gates in B. These two points of view are clearly equivalent mathematically, and, in fact, our work can be phrased in either or both settings. However, for the sake of clarity, we prefer to fix one and stick to it. So, even though it may seem a less direct approach, we choose the PosSLP point of view, both because it allows finer grained classifications – we will state some intermediate result for many-one instead of Turing reductions – and because, we believe, in total it makes the argument shorter.

For each finite set of real functions S semialgebraic over Q, we prove that PosSLP(S) is polynomial time Turing reducible to PosSLP—this is a direct generalization of a result in [ABKM09] proving BP(P^0_2) = BP(P_{algebraic}), however we obtain our result with different techniques, involving algebraic number theory and model theory. Then, under the additional hypothesis that all the functions in S are continuous, that + and − are in S, and that arbitrarily small numbers can be represented by circuits with gates in S, we obtain the following dichotomy for the computational complexity of PosSLP(S):

Either all the functions in S are piecewise linear, and in this case PosSLP(S) is in P, or not, and in this case PosSLP(S) is polynomial time equivalent to PosSLP (in the sense of Turing reductions).

Finally, as a possible indication for future research, we would like to raise the question of machine constants (which are just 0-ary primitives) and other sets of primitives not semialgebraic over Q, most importantly those that are commonly met in practice: for instance, the typical pocket calculator functions sin(x), log(x), e^x, &c. Taking Boolean parts, one can compare the relative strength of any two computational models: is it possible to show equivalence or separation results involving transcendental functions?

II. Preliminaries & Notations

We will consider circuits whose gates operate on real numbers (real circuits, for short). Our circuits will have any number of input gates and precisely one output gate, hence, for us, circuits compute multivariate real functions. If a circuit has no input gates, we will call it a closed circuit: closed circuits represent a well defined real value. We measure the size of a circuit as the number of its gates. We call basis a finite set of real functions, which we intend to use as gates. Given a basis B, a B-circuit is a circuit with gates belonging to B, and V(B) is the set of the values of all closed B-circuits. We will consistently employ the same symbol to denote a circuit and the function it represents. We will identify algebraic formulæ with tree-like circuits. The notation ||·|| denotes the circuit size, while ||·|| is the absolute value.

Broadly speaking, we are interested in the efficient evaluation of the sign of closed circuits in some basis B, by use of an oracle for the evaluation of the sign of closed circuits in some other basis B'. In general, we will employ the technique, common in computational geometry, of combining approximate evaluation with explicit zero bounds: see [LPY05] for a survey.

Definition II.1 (zero bound). Let C be a class of closed real circuits. We say that \( Z : C \to \mathbb{R}_{>0} \) is a zero bound for C if, for all \( c \in C \), either \( c \) evaluates to zero (\( c = 0 \)), or \( Z(c) < |c| \).

It is clear that, given a zero bound \( Z \) for \( C \), we can decide the sign of a circuit \( c \in C \) by looking at an approximation \( c' \) of \( c \) up to an additive error bounded by \( Z(c)/2 \). In fact, if \(|c'| \leq Z(c)/2\), then \( c = 0 \), otherwise \( c' \) and \( c \) have the same sign. Both directions of our argument will follow this general recipe. We will now summarize a few facts about semialgebraic sets and Weil heights, that we need in order to provide the ingredients.

A subset of \( \mathbb{R}^n \) is semialgebraic over a subring \( A \) of \( \mathbb{R} \) if it can be described by a finite Boolean combination of subsets of \( \mathbb{R}^n \) defined by polynomial equalities \( P_i(x) = 0 \) or inequalities \( Q_j(x) > 0 \), with \( P_i, Q_j \in A[x] \). A function
is said to be semialgebraic over $A$ if its graph, as a set, is semialgebraic over $A$. As a general reference for the reader, we suggest the book of Van Den Dries [vdD98]. In this paper, we are mainly interested in functions and sets semialgebraic over $\mathbb{Q}$. We recall the central property of semialgebraic sets.

**Fact II.2** (Tarski-Seidenberg). Let $\phi$ be a first-order formula in the field language $(0,1,+,−,\times)$, then there is a quantifier free formula $\psi$ in the same language such that

$$\mathbb{R} \models \phi \equiv \psi$$

In other words, a set is first-order definable over $A$ in the real field, if and only if it is semialgebraic over $A$. This statement has a number of well known consequences on the geometry of semialgebraic sets: for instance semialgebraic sets have finitely many connected components, and semialgebraic functions are almost everywhere infinitely differentiable. As an example, it is convenient to state the following corollary (see [vdD98, Chapter 2(3.7)]), which will constitute the hinge of our zero bound.

**Fact II.3.** If $g: \mathbb{R} \to \mathbb{R}$ is semialgebraic (over $\mathbb{R}$), then there are $d \in \mathbb{N}$ and $M > 0$ such that $|g(x)| \leq x^d$ for all $x > M$.

Finally, we summarize a few facts that we need about the notion of absolute Weil height, which was introduced by André Weil in the context of Diophantine geometry. For our purpose, absolute heights are real numbers associated to points in $\mathbb{P}^n(\mathbb{Q}^{ab})$. The absolute height $H(p)$ of $p \in \mathbb{P}^n(\mathbb{Q}^{ab})$ is a positive real number meant to represent a notion of size of $p$. For instance, if $p$ is in $\mathbb{P}^n(\mathbb{Q})$, then its absolute height can be determined as follows: take a tuple $q = \{n+1\}$ coprime integers representing $p$, then $H(p) = \max_q q$. For the general definition, which is too technical for this introduction, we refer the reader to [vdD98 Chapter 3]. We will summarize below the facts that we need. The absolute height $H(x)$ of an algebraic number $x$ is defined as the height of $(1, x) \in \mathbb{P}^1(\mathbb{Q}^{ab})$. The following facts will be used to bound the result of algebraic computations.

**Fact II.4** ([Wal00 Property 3.3]). Let $a$ and $b$ be algebraic numbers, then

$$H(ab) \leq H(a)H(b) \quad H(a \pm b) \leq 2H(a)H(b)$$

**Fact II.5.** Let $p(x) = a_0 + a_1x + \cdots + a_dx^d$ be a polynomial with algebraic coefficients. Let $\alpha$ be a root of $p$. Then $H(\alpha) \leq 2^d \prod_{k=1}^{d} H(a_i)$

**Proof.** Follows from [SIL86 Chapter VIII Theorem 5.9] observing that $H([a_0 \ldots a_d]) \leq \prod_{k=1}^{d} H(a_i)$. □

**Fact II.6.** Let $\alpha \neq 0$ be an algebraic number of degree $d$. Then

$$H(\alpha)^{-d} \leq |\alpha| \leq H(\alpha)^d$$

**Proof.** Immediate from the definition. □

### III. Statement of the Results

Now we state our main results. A brief discussion of the hypothesis of Theorem III.3 as well as a third result that may be of interest in certain cases, can be found in Section VII. The next three sections will be devoted to proving Theorem III.2 and Theorem III.3.

**Definition III.1.** Let $B$ be a finite set of functions $\mathbb{R}^* \to \mathbb{R}$. The decision problem $\text{PosSLP}(B)$ is defined as follows. **Input:** a closed $B$-circuit $c$ **Output:** YES if $c > 0$, NO otherwise Keeping the same notation as [ABKM09], we will denote $\text{PosSLP}(\mathbb{0}, 1, +, −, \times)$ simply as $\text{PosSLP}$.

**Theorem III.2.** Let $B$ be a finite set of real functions semialgebraic over $\mathbb{Q}$. Then $\text{PosSLP}(B)$ is polynomial time Turing reducible to $\text{PosSLP}$.

**Theorem III.3.** Let $B \supset \{+, −\}$ be a finite set of continuous functions, semialgebraic over $\mathbb{Q}$, such that $V(B)$ is dense. The following dichotomy holds: either $\text{PosSLP}(B)$ is in $\text{P}$, if all the functions in $B$ are piecewise linear; or, if not, $\text{PosSLP}$ and $\text{PosSLP}(B)$ are mutually polynomial time Turing reducible.

### IV. Circuits with Gates for Polynomial Roots

In this section we study circuits in the basis $\text{PosSLP}$ with the convention that $\mathbb{Q}$ denotes the function mapping a tuple $(a_0 \ldots a_d)$ to the largest real root of the polynomial $\sum_i a_ix^i \in \mathbb{R}[x]$. Let $B$ be a finite set of continuous functions $\mathbb{R}^* \to \mathbb{R}$, such that $V(B)$ is dense. The following dichotomy holds: either $\text{PosSLP}(B)$ is in $\text{P}$, if all the functions in $B$ are piecewise linear; or, if not, $\text{PosSLP}$ and $\text{PosSLP}(B)$ are mutually polynomial time Turing reducible.

Let $\text{PosSLP}(B_1) \leq_{\text{P}} \text{PosSLP}(B_2) \leq_{\text{P}} \text{PosSLP}(B_3) \leq_{\text{P}} \cdots$ where $\leq_{\text{m}}$ and $\equiv_{\text{m}}$ denote polynomial time many-one reducibility and mutual many-one reducibility—the first equivalence is standard: keep rationals as pairs numerator-denominator. Our goal is to prove the following statement.

**Proposition IV.1.** For any fixed $d$, the decision problem $\text{PosSLP}(B_d)$ is polynomial time Turing reducible to $\text{PosSLP}$.
It is convenient to isolate a number of intermediate steps. First we will prove a zero bound for $B_d$-circuits in Lemma [IV.2]. Then, we will prove, in Lemma [IV.3], that, for a subclass of $B_d$-circuits which we call regular, there is an effective bound connecting the error of an approximate evaluating procedure at each gate, with the error accumulated at the end of the evaluation. Third, we will show how to convert $B_d$-circuits into regular $B_d$-circuits effectively in Lemma [IV.6]. Finally we will prove Proposition [IV.1] giving an evaluation procedure for regular $B_d$-circuits based on Newton’s method.

**Lemma IV.2.** For any fixed $d$ there is a constant $C_d$ such that for any closed $B_d$-circuit $c \neq 0$ we have

$$2^{-2^e}a^{1/|d|} < |c| < 2^{2^e}a^{1/|d|}$$

Proof. Using Fact [I.4] and Fact [I.5] we get a constant $K_d$ such that

$$H(c) < 2^{2^e}d^{1/|d|}$$

where $H(c)$ denotes the absolute Weil height of the algebraic number represented by $c$. The lemma follows immediately from Fact [I.5] observing that the degree of $c$ is bounded by a single exponential in $\|c\|$.

**Definition IV.3.** Let $c$ be a closed $B_d$-circuit. We say that a gate $g$ of $c$ is regular if

1. $g$ is 0 or 1
2. $g$ is $+, -, \times$, and its inputs are regular gates
3. $g$ is $\text{ch}(x, n, z, p)$ with $x$ regular and $n$ regular
4. $g$ is $\text{ch}(x, n, z, p)$ with $x$ regular and $p$ regular
5. $g$ is $\text{r}(a_0 \ldots a_8)$, its inputs $a_0 \ldots a_8$ are regular, the largest real root $\zeta$ of $p(x) = \sum a_i x^i$ exists, and the first $\delta$ derivatives of $p$ do not vanish at $\zeta$.

We say that $c$ is regular if its output gate is.

Recall that a function $f : X \to \mathbb{R}$ is called upper semicontinuous at $a \in X$ if for each $k > f(a)$ there is a neighbourhood $V$ of $a$ in $X$ such that $k > f(x)$ for each $x \in V$. Also, $f$ is called upper semicontinuous if it is upper semicontinuous at every $a \in X$.

During the proof of the next lemma, we will need the following fact, which follows almost immediately from the definition—see also [Bou66] IV §6.2 Theorem 3.

**Fact IV.4.** Let $X$ be a non-empty compact topological space and $f : X \to \mathbb{R}$ be upper semicontinuous. Then there is a $a \in X$ such that

$$\sup_{x \in X} f(x) = \max_{x \in X} f(x) = f(a)$$

**Lemma IV.5.** For any fixed $d$ there is a constant $E_d$ such that, for any regular $B_d$-circuit $c$ the following holds. Evaluate $c$ with infinite precision, and perturb the procedure adding at each $\tau_3$ gate an error which is smaller in absolute value than

$$e^{2^{-2^e}d^{1/|d|}}$$

for $e \in [0, 1]$ at regular gates, and unconstrained at non-regular gates. Then the error accumulated on the result is less than

$$e^{2^{-2^e}d^{1/|d|}}$$

Proof. We study the loss of precision incurred by our perturbed evaluation procedure at each gate.

The proof is stated in the language of o-minimality—see [vdD98] for a reference. Observe that the gates in $B_d$ are semialgebraic over $\mathbb{Q}$, or, in other words, first-order definable in $(\mathbb{R}, +, \times)$ by the Tarski-Seidenberg quantifier elimination theorem. In the following, definable will mean first-order definable in $(\mathbb{R}, +, \times)$.

By Lemma [IV.2] and induction on the size of $c$, the choices performed by regular $c$ gates are unaffected by the accumulated errors. Therefore, by Definition [IV.3] the errors occurred at non-regular gates have no influence on the value of $c$, therefore we can simply ignore the effect of these errors. Now we concentrate on regular $r$ gates. Clearly, all these gates compute the largest real root of a polynomial of degree $\leq d$, which, by the regularity, must be a single root.

Define the set $R \subset \mathbb{R}^{d+1}$ as the set of those tuples $x$ such that the largest real root of the polynomial represented by $x$ exists and is a single root. Consider a tuple $x_0 = R$ representing a polynomial with largest real root $r = r_d(x_0)$. Clearly, there is a positive $A$ such that $r_d = A$-Lipschitz on a neighbourhood of $x_0$ of radius $1/A$ with respect to the $\infty$-norm, i.e. for all tuples $\epsilon_1, \epsilon_2, x_0 \in \mathbb{R}^{d+1}$ with $|\epsilon_1|_\infty, |\epsilon_2|_\infty < 1/A$ we have

$$|r_d(x_0 + \epsilon_1) - r_d(x_0 + \epsilon_2)| \leq A|\epsilon_1 - \epsilon_2|_\infty$$

and, in particular, $R$ is an open set. Now, by [vdD98] Chapter 6 Proposition 1.2], the structure $(\mathbb{R}, +, \times)$ has definable Skolem functions, this means that there is a definable function $\alpha$ such that for all $x_0 \in R$ the choice $A = \alpha(x_0)$ works. Finally we observe that $\alpha$ can be replaced by

$$\alpha'(x) \equiv \inf_{|x - y|_\infty < \frac{1}{2\alpha(y)}} 2\alpha(y)$$

which is well defined, definable, and upper semicontinuous. In fact, if $|\epsilon_1|_\infty, |\epsilon_2|_\infty < 1/\alpha'(x)$, then for all $y$ such that $|x - y|_\infty < 1/(2\alpha(y))$ and $|\epsilon_1|_\infty, |\epsilon_2|_\infty < 1/(2\alpha(y))$ we have

$$|r_d(x + \epsilon_1) - r_d(x + \epsilon_2)| \leq 2\alpha(y)|\epsilon_1 - \epsilon_2|_\infty$$

and taking the inf on $2\alpha(y)$

$$|r_d(x + \epsilon_1) - r_d(x + \epsilon_2)| \leq \alpha'(x)|\epsilon_1 - \epsilon_2|_\infty$$

Well definedness of $\alpha'$ follows from the fact that the choice $y = x$ always verifies the condition $|x - y|_\infty < 1/(2\alpha(y))$. Definability is immediate. Upper semicontinuity follows observing that if $\alpha'(x) < k$, then there is $y$ such that $|x - y|_\infty < 1/(2\alpha(y))$ and $2\alpha(y) < k$, hence $\alpha'(x') < 2\alpha(y) < k$ for all $x'$ in a neighbourhood of $x$ of radius $1/(2\alpha(y)) - |x - y|_\infty$.

Now we turn our attention to the set $R(N)$ of all elements of $R$ whose coordinates can be represented by $B_d$-circuits of
size at most \(N\). Our aim is to construct a uniform family of compact definable sets \(R'(t)\) such that

\[
R(N) \subset R' \left( 2^{2^{g(N)}} \right) \subset R
\]

for some constant \(K\). By [PD01, Theorem 2.4.1] there are finitely many polynomials over the integers \(p_{i,j}\) such that \(R\) can be written in the following form

\[
R = \bigcup_{i} \bigcap_{j} \{ x \in \mathbb{R}^{d+1} \mid 0 < p_{i,j}(x) \}
\]

Let \(S\) be the maximum size of the circuits representing the polynomials \(p_{i,j}\). Fix \(K\) in such a way that \(C_{d}((d+1)N + S) \leq KN\) for all positive integer \(N\). By Lemma [IV.2]

\[
R(N) \subset \left\{ x \in \mathbb{R}^{d+1} \mid 2^{-2^{KN}} \leq \max_{i,j} p_{i,j}(x) \right\} \subset R
\]

The middle set is closed, and, again by lemma [IV.2], we can intersect it with a box of radius \(2^{2^{KN}}\). Hence we have our set

\[
R'(t) \equiv \left\{ x \in \mathbb{R}^{d+1} \mid (|x|_{0} \leq t) \land \left( \frac{1}{t} \leq \max_{i,j} p_{i,j}(x) \right) \right\}
\]

Finally we consider the definable function

\[
\beta(t) \equiv \max_{x \in R'(t)} \alpha'(x)
\]

\(R'(t)\) is compact and \(\alpha'(x)\) is upper semicontinuous, therefore, by Fact [IV.4], the function \(\beta\) is well defined. Since the structure \((\mathbb{R}, +, \times)\) is polynomially bounded – Fact [II.3] – there are positive integers \(m\) and \(n\) such that \(\beta(t) < mt^n\) for all \(t \geq 1\). Let

\[
\gamma(N) = m2^{n2^{KN}}
\]

Considering circuits of size bounded by \(N\), we know that, by construction, if we perturb the inputs of a regular \(r_{d}\) gate by an additive error bounded by \(e < 1/\beta(2^{2^{KN}})\), then the resulting perturbation on the output is bounded by \(e\beta(2^{2^{KN}})\). Hence we get the following bound:

(*) for circuits of size bounded by \(N\), if the inputs of a regular \(r_{d}\) gate are perturbed by at most \(e < 1/\gamma(N)\), then the resulting perturbation on the output is bounded by \(\gamma(N)\).

The statement (*) holds for \(r_{d}\) gates, holds a fortiori for \(r_{2d}\) gates with \(\delta' < \delta\), and it can easily be proven, as well, for \(+, -\), and \(\times\): in fact, in this case it is a direct consequence of Lemma [IV.2]. Now, choose \(E_{d}\) in such a way that

\[
2^{2E_{d}X} \geq (2\gamma(X))^X
\]

for all positive integer \(X\). We need to prove that as long as the error added on each \(r_{d}\) gate of a circuit \(c\) is bounded by

\[
e^{2^{-2E_{d}X}} < e^{2\gamma(||c||)}^{-1} \|c\|
\]

the accumulated error on the evaluation of \(c\) is bounded by

\[
e \frac{e}{2\gamma(||c||)} < e^{2^{-2E_{d}X}}
\]

This can be done observing that, by induction, the accumulated error on the output of a gate at depth \(i\) in \(c\) is bounded by \(e(2\gamma(||c||))^{-i-1}\).

**Lemma IV.6.** There is a polynomial time procedure that given a closed \(B_{d}\)-circuit \(c\) generates a regular \(B_{d}\)-circuit \(\tilde{c}\) such that \(\tilde{c} = c\).

**Proof.** Our procedure consists of two steps: first we regularize the \(r_{d}\) gates, and then we deal with the choice gates. We say that a gate \(g\) is quasi-regular if

1. \(g = 0\) or 1
2. \(g = +, -\), or \(\times\) and its inputs are quasi-regular gates
3. \(g = \text{ch}(x, n, z, p)\) with \(x\) quasi-regular < 0 and \(n\) quasi-regular
4. \(g = \text{ch}(x, n, z, p)\) with \(x\) quasi-regular = 0 and \(z\) quasi-regular
5. \(g = \text{ch}(x, n, z, p)\) with \(x\) quasi-regular > 0 and \(p\) quasi-regular
6. \(g = r_{0}(a_0 \ldots a_d)\), its inputs \(a_0 \ldots a_d\) are quasi-regular, the largest real root \(\zeta\) of \(p(x) \equiv \sum_{i} a_i x^i\) exists, and the first derivative of \(p\) do not vanish at \(\zeta\).

A quasi-regular circuit is one whose output gate is quasi-regular. In other words, being quasi-regular is like being regular, except that the first arguments of choice gates may be zero.

For the first step, our procedure generates a quasi-regular \(B_{d}\)-circuit \(c'\) such that \(c' = c\). It suffices to prove that we can replace \(r_{d}(a_0 \ldots a_d)\) with a suitable circuit \(s_{d}(a_0 \ldots a_d)\) in such a way that the output of \(s_{d}\) is quasi-regular whenever its inputs are quasi-regular. For the case \(d = 0\), we define \(s_{0}(x) = 0\). For positive \(d\), in order to construct \(s_{d}\), we produce a more general family of circuits \(s_{d,i}\) such that \(s_{d,i}(a_0 \ldots a_d)\) evaluates to the \(i\)-th largest real root of \(p\) (counted with multiplicity), or to 0 if said root does not exist. Clearly the choice \(s_{d} = s_{d,1}\) works. By induction we will show how to build \(s_{d,i}\) using all \(s_{d',i'}\) for \(d' < d\), or for \(d' = d\) and \(i' < i\).

If \(i = 1\), we use choice gates to test whether the degree of \(p\) is less than \(d\), or, using Tarski-Seidenberg, whether the largest root of \(p\) either does not exist or it is a root of some derivative of \(p\). In each of these cases, we choose to use the appropriate \(s_{d',i'}\) or the constant 0. Otherwise, we use \(r_{d}\). If \(i > 1\), we use Tarski-Seidenberg again, to guard against the case in which the required root does not exist, and if it exists we use \(s_{d-1,i-1}\) applied to the coefficients of \(p(x)/(x-s_{d,1}(a_0 \ldots a_d))\) which can be computed by polynomial division (hence using +, -, and \(\times\), because the denominator is a monic polynomial).

Now we have a quasi-regular circuit \(c'\) of size bounded by a multiple (depending on \(d\)) of the size of \(c\). By Lemma [IV.2] for each gate \(g\) of \(c'\), either \(g = 0\), or \(|g| > 2^{-2^{d}2^{d+1}}\). Hence

\[
\text{ch}(g, n, z, p) = \left\{ \begin{array}{ll} 2^{2^{d+1}} & g = 1, n, 0, \text{ch}(2^{2^{d+1}}g - 1, z, 0, p) \end{array} \right.
\]

where, clearly, the first arguments of the \(\text{ch}\) gates on the right hand side can never be zero. Therefore, performing the substitution above on all gates of \(c'\) gives us a regular circuit \(\tilde{c}\). Finally, we observe that our substitution does entail at most a linear increase in size from \(c'\) to \(\tilde{c}\). In fact, \(2^{2n}\)
can be computed by a circuit of size linear in \( n \) by iterated squaring.

**Observation IV.7.** Fix a base \( B = \{0, 1, +, -, \times, \text{ch}, \ldots\} \).
Let \( a < b \) be integers, and let \( f_t : [2^k, 2^k] \to \mathbb{R} \) be a family of continuous monotonic functions parameterized by \( t \in S \subset \mathbb{R}^n \). Assume that the family \( f_t \) is represented as a \( B \)-circuit—i.e., there is a \( B \)-circuit \( f \) that takes inputs \( t \) and \( x \) and computes \( f_t(x) \). Moreover, assume that \( f_t(2^k) f(2^k) < 0 \) for all \( t \in T \). Then there is a \( B \)-circuit \( g \) representing a function \( : \mathbb{R}^n \to \mathbb{R} \) mapping any \( t \in T \) to a power of 2 such that

\[
f_t\left( g(t) \right) f_t\left( 2g(t) \right) \leq 0
\]

and \( \|g\| \leq p(|a| + |b| + \|f\|) \) for some fixed polynomial \( p \).

**Proof.** The observation says that we can find the order of magnitude of the solution \( x \) of the equation \( f_t(x) = 0 \) uniformly in \( t \) through a circuit of size polynomial in \( |a| + |b| + \|f\| \). It is easy to devise a bisecting procedure that finds the binary digits of an integer \( e_t \) in such a way that

\[
f_t(2^e) f_t(2^{e+1}) \leq 0
\]

and to turn such procedure into a \( B \)-circuit of the required size.

**Proof of Proposition IV.7.** For \( d = 1 \) the result follows immediately observing that \( r_d(x, y) = -x/y \). Hence, it suffices to produce a polynomial time algorithm that, for \( d > 1 \), given a \( B_d \)-circuit \( c \), computes a \( B_{d-1} \)-circuit \( \tilde{c} \) such that \( \tilde{c} \) is positive if and only if \( c \) is positive. First we use Lemma IV.6 to get a regular circuit \( \tilde{c} = c \). Our plan, now, is to approximate the \( r_d \) gates of \( \tilde{c} \) with suitable \( B_{d-1} \)-circuits, in such a way that the errors introduced do not alter the sign of any gate in \( \tilde{c} \), and ultimately of \( c \) itself. We will henceforth produce a new \( B_{d-1} \)-circuit \( \tilde{c} \) obtained from \( c \) through the replacement of \( r_d \) gates by \( B_{d-1} \)-circuits based on the Newton’s approximation method.

Let \( N_{d}^{1}(x_0, a_0 \ldots a_d) \) be a \( B_{d-1} \)-circuit representing one iteration of Newton’s method applied to the polynomial \( p(x) \approx \sum_i a_i x^i \), i.e.

\[
N_{d}^{1}(x_0, a_0 \ldots a_d) = x_0 - \sum_{i} a_i x_0^i
\]

For every positive integer \( k \), build a \( B_{d-1} \)-circuit \( N_{d}^{k}(x_0, a_0 \ldots a_d) \) representing the \( k \)-th iterate of Newton’s method applied to starting from \( x_0 \) through the iteration

\[
N_{d}^{k+1}(x_0, a_0 \ldots a_d) = N_{d}^{k}\left( N_{d}^{1}(x_0, a_0 \ldots a_d), a_0 \ldots a_d \right)
\]

it is clear that, for fixed \( d \), the size of \( N_{d}^{k} \) is linear in \( k \). Let \( \zeta \) be the largest real root of \( p \). We say that \( x_0 \) is an approximate root (cf. BCS98 Chapter 8 § 1) of \( p \) if for all positive \( k \)

\[
|N_{d}^{k}(x_0, a_0 \ldots a_d) - \zeta| \leq |x_0 - \zeta| 2^{-2^{k-1}}
\]

We will produce a \( B_{d-1} \)-circuit \( A_d(a_0 \ldots a_d) \) evaluating to a suitable approximate root of \( p \). Assuming that we have \( A_d \), we build \( \tilde{c} \) substituting each \( r_d(a_0 \ldots a_d) \) with \( \tilde{r}_d(a_0 \ldots a_d) \equiv N_{d}(E_{d}+C_{d+1})(|\|+K) \left( A_d(a_0 \ldots a_d), a_0 \ldots a_d \right) \) for a suitable constant positive integer \( K \) that we will describe later in the proof. At each \( r_d \) gate, our substitution introduces an error bounded by

\[
2^{-2^d(\ell+1)+K}
\]

Therefore, irrespective of the value of \( K \geq 1 \), this substitution verifies the hypothesis of Lemma [IV.3] with \( \epsilon = 0.5 \), and, by Lemma IV.2 we have that \( \tilde{c} - 2^{-2^{d}2^{\ell+1}} \) is positive if and only if \( c \) is positive, hence the statement.

The last remaining step is the construction of \( A_d \). The circuit \( A_d \) that we are going to build will depend on the size of \( c \), i.e. it is going to work only in our particular situation, and not necessarily for any value of its inputs \( a_0 \ldots a_d \). To follow the proof, it is convenient to consider the construction of \( \tilde{c} \) as a stepwise process in which \( r_d \) gates are replaced in evaluation order. At a given step we are going to replace \( r_d(a_0 \ldots a_d) \) with \( \tilde{r}_d(a_0 \ldots a_d) \), where \( a_0 \ldots a_d \) are \( B_{d-1} \)-circuits obtained from \( a_0 \ldots a_d \) during the preceding steps of the construction, which, by induction, we can assume to yield the required precision at each \( r_d \) gate. If our particular \( r_d \) gate is not regular, then there is nothing to prove; therefore we can assume it to be regular. Let \( \tilde{p} \) denote \( \sum_i a_i x^i \). The derivatives of \( p \) in \( \zeta \) can be expressed as \( B_{d} \)-circuits with inputs \( a_0 \ldots a_d \) of size bounded by some constant \( \bar{K} \)—let \( r_d(a_0 \ldots a_d) \) denote the circuit for the \( i \)-th derivative of \( p \) at \( \zeta \). Therefore, for each \( i \leq d = \deg(p) \), Lemma IV.2 gives us

\[
2^{-2^d 2^{\ell+1+k}} < |\tilde{p}(i)(\zeta)|
\]

On the other hand, choosing \( K > k \), we see that the derivatives of \( \tilde{p} \) at its largest real root \( \zeta \) approximate the derivatives of \( p \) at \( \zeta \) to an absolute error smaller than

\[
\frac{1}{2} 2^{-2^d 2^{\ell+1+k}}
\]

in fact, this bound follows applying Lemma IV.5 to the circuits obtained plugging \( a_0 \ldots a_d \) into \( r_d \). Hence we have that, for each \( i \leq d \)

\[
\frac{1}{2} 2^{-2^d 2^{\ell+1+k}} < \tilde{p}(i)(\zeta)
\]

and, in particular, none of the first \( d \) derivatives of \( \tilde{p} \) vanishes at \( \zeta \).

By Tarski-Seidenberg, we can test the signs of the derivatives of \( \tilde{p} \) in \( \zeta \) through a fixed Boolean combination of polynomial conditions on the coefficients \( a_0 \ldots a_d \), and, in turn, we can realize this Boolean combination as a switching network of \( \text{ch} \) gates. Therefore we can build a circuit designed to decide its course of action based on the signs of the derivatives of \( \tilde{p} \) at \( \zeta \). From now on, we assume that the first and second derivatives of \( \tilde{p} \) in \( \zeta \) are positive: the reader can easily work out the three other cases. We claim that, if we can find \( a < b < c \) such that the following conditions hold

1. \( c - b \leq (b - a)/2 \)
2. \( b \leq \zeta < c \)
3. the first derivative of \( \tilde{p} \) is positive on \([a, c]\)
4. there is \( e \) such that \( 2^e \leq \tilde{p}(x) \leq 2^{e+1} \) for all \( x \in [a, c] \)
then \( c \) is an approximate root of \( p \). In fact, let \( z_k = N^d_d(c, \tilde{a}_0 \ldots \tilde{a}_d) \). It is well known (for instance [Aik89, Formula 2.2.2]) that
\[
  z_{k+1} - \tilde{c} = \frac{\tilde{p}''(\tilde{c})}{2\tilde{p}'(z_k)} (z_k - \tilde{c})^2
\]
for some \( \tilde{c} \in [\tilde{c}, z_k] \). Observing that
\[
  \tilde{p}''(\tilde{c}) \leq 2^{e+1}, \quad \tilde{p}'(z_k) \geq (z_k - a) 2^e
\]
we get
\[
  \frac{z_{k+1} - \tilde{c}}{b-a} \leq \left( \frac{z_k - \tilde{c}}{b-a} \right)^2
\]
and since \( z_k - \tilde{c} \leq c - b \leq (b-a)/2 \) we have the claim.

We will now look for a suitable choice of \( a, b, \) and \( c \). First we use \( r_{d-1} \) gates to write the roots of the first, second, and third derivatives of \( \tilde{p} \). We call \( S \) the set of all these values plus \( -1 \) and \( 1 \), which are an upper and lower bound for \( \tilde{c} \). Now, choosing gate sets, we can use Observation \([V,7]\) to the composition \( p \circ \tilde{p}''^{-1} \) and get a new interval \([a, d] \subset [s, t] \) such that \( \tilde{c} \in [a, d] \) and there is an \( e \) such that
\[
  2^e \leq \tilde{p}''(x) \leq 2^{e+1}
\]
for all \( x \in [a, d] \).

Now we turn our attention to the order of magnitude of \( \tilde{c} - a \). First we use a choice gate to test whether \( \tilde{c} \) happens to be within the required precision from \( a \). If this is the case we just output \( a + 2^{2^e d} \epsilon e )^{e+K} \). If this is not the case, then
\[
  2^{-2^2 d} a (\epsilon |\epsilon| + H) \leq \tilde{c} - a \leq 1 + 2^2 d a |\epsilon|
\]
i.e. the order of magnitude of \( \tilde{c} - a \) is bounded in a range of size linear in \( |\epsilon| \). Again by Observation \([V,7]\) we find \( b \) and \( c \) such that \( c - b \leq (b-a)/2 \). This concludes the proof. \( \square \)

**V. Reducing \( \times \) to an Arbitrary Semialgebraic Function**

In this section we address the opposite problem of Section \([V]\)
amely we want to recover \( \times \) starting from a regular not piecewise linear function \( f: \mathbb{R} \to \mathbb{R} \). The argument is comparatively technically easier. The idea is to observe that the product can be simulated using linear operations and the square function. In turn, the square can be approximated, in some sense, zooming in on a point on the graph of \( f \), because we can expect that, under strong magnification, \( f \) should be practically indistinguishable from its second order approximation. The zero bound, in this direction, is trivial, because the input circuit computes an integer value. Technical obstacles lie in the fact that we can use just + and −, as opposed to all linear functions, and in balancing the quality of our approximation with the size of the resulting circuit.

**Lemma V.1.** Let \( p \in \mathbb{R}[x] \) be a polynomial of degree \( d > 1 \), and let \( \alpha \) be a positive real. Then PosSLP is polynomial time many-one reducible to PosSLP(0, \( \alpha \), +, −, \( \times \)).

**Proof.** We will produce a polynomial time procedure that, given a closed circuit \( c \) with gates in the basis \( B \equiv \{0, 1, +, −, \times \} \), generates a closed circuit \( \tilde{c} \) with gates in \( B' \equiv \{0, \alpha, +, −, \times \} \), in such a way that \( c \) is positive iff \( \tilde{c} \) is positive.

First we prove that without loss of generality we can assume \( p(x) = ax^2 \) for some real coefficient \( a > 0 \). Let \( q \) be the polynomial
\[
  q(x) = \Delta d^{-2} [p_{i+1}] (x)
\]
where \( \Delta d^{-2} \) is the \((d-2)\)-th iterate of the first difference operator
\[
  \Delta_1 [f] (x) = f (x+1) - f (x)
\]
Clearly \( q \) can be implemented with a fixed number, depending on \( d \), of \( B' \)-gates, and the degree of \( q \) is 2. Now, the polynomial
\[
  q(2x) - 2q(x) + q(0)
\]
has the required form, except at most for the sign of \( a \), which is easily corrected.

Assuming \( p(x) = ax^2 \), we observe that
\[
  2axy = p(x+y) - p(x) - p(y)
\]
therefore we can replace \( B' \) with
\[
  \{0, 1, +, −, (x, y) \mapsto 2axy\}
\]
Now we modify \( c \) in such a way that there is only one 1 gate and all the paths from any given gate to the output node have the same length, this can be accomplished by addition of no more than \( 2|\epsilon|^2 \) dummy gates arranged in \( 0 + 0 + \cdots + \) subcircuits. Then, for every \( xy \) gate in \( c \) we put a corresponding \( 2axy \) gate in \( \tilde{c} \). For every \( x \pm y \) gate we put a \( 2a1(x \pm y) \) subcircuit. And for zeroes and ones we put zeroes and \( a \) respectively. Let \( b \) denote the depth of the only 1 gate in the modified \( c \) circuit—we can clearly assume that this is the deepest gate. Then the value of a gate at depth \( i \) of the modified \( c \) circuit is \( (2a)^{i-h}/\alpha \) times the value of the corresponding object in the new circuit. Therefore \( \tilde{c} \) and \( c \) have the same sign. \( \square \)

**Lemma V.2.** Let \( B \) be the basis \( \{0, +, −, f, \ldots \} \) where \( f \) denotes a unary function and the dots − − − indicate additional functions. Assume that for some \( \alpha, \beta > 0 \) we have
\[
  |f(x) - ax^2| \leq |x|^3
\]
for all \( x \in [-\beta, \beta] \), and assume that \( V(B) \) is dense in \( \mathbb{R} \). Then PosSLP is polynomial time many-one reducible to PosSLP(\( B \)).

**Proof.** By Lemma \([V,1]\) suffices to show a polynomial time procedure that, given a circuit \( c \) with gates in the basis
$B' \equiv (0, 1, +, -, \cdot, ^2)$, generates a $B$-circuit $\tilde{c}$ in such a way that $\epsilon$ is positive iff $\tilde{c}$ is positive.

Without loss of generality we may assume $2\beta \leq \alpha$. First we describe a procedure that, given positive numbers $\epsilon \leq \alpha$ and $u$, with $u$ represented as a $B$-circuit, and given a $B'$-circuit $n$, attempts to produce a $B$-circuit $a(n, u, \epsilon)$ such that

$$|a(n, u, \epsilon) - nu| < \frac{\epsilon u}{\alpha \cdot \alpha} \quad (*)$$

the intuition is that $a(n, u, \epsilon)$ represents an $\alpha/\epsilon$-approximation of $n$ scaled down by a factor of $u$. This procedure may either succeed in its goal or fail explicitly.

The circuit $a(n, u, \epsilon)$ is defined inductively on the structure of $n$

$$a(m^2, f(u), \epsilon) = f\left(a\left(m, u, \frac{\epsilon - 4(|m| + 1)^3 u}{4|m| + 2}\right)\right)$$

fail if $\epsilon \leq 4(|m| + 1)^3u$ or $\beta < (|m| + 1)u$.

$$a(m_1 \pm m_2, u, \epsilon) = a\left(m_1, u, \frac{\epsilon}{2}\right) \pm a\left(m_2, u, \frac{\epsilon}{2}\right)$$

$$a(1, u, \epsilon) = u$$

$$a(0, u, \epsilon) = 0$$

the procedure fails in any other case. It is easy to check by direct computation that property $(*)$ is preserved—for the first case, the computation goes as follows. Assume $|(|m| + 1)u \leq \beta$ and remember that $\epsilon \leq \alpha$ and $m$ is an integer. Let

$$\epsilon = \frac{\epsilon - 4(|m| + 1)^3 u}{4|m| + 2}$$

then clearly $0 < \epsilon \leq \alpha$, and we have

$$|f(a(m, u, \epsilon)) - m^2 f(u)| \leq \left| f(a(m, u, \epsilon)) - \alpha (a(m, u, \epsilon))^2 \right| + \alpha (a(m, u, \epsilon))^2 - 2\alpha m^2 u^2 + \frac{\epsilon u^2}{\alpha}$$

hence property $(*)$ is preserved.

Now we find $u_n$ such that $a(n, u_n, \alpha)$ does not fail. To this aim, we use the density hypothesis to pick a $B$-circuit $k$ such that

$$0 < k \leq \min\left(\frac{1}{3\alpha \cdot \beta}, \frac{\alpha}{\alpha}\right)$$

Observe that $0 < f'(k) \leq \beta < 2^{-i}$ for all $i$. We claim that $u_n \equiv f^{3|n|+3}(k)$ does the job. In fact, it suffices to show by induction that, at each step $a(n', u', \epsilon')$ of the recursion, the following conditions are met

$$u' \leq \beta^2 - 2^{3|n|+3-d}$$

$$\epsilon' \geq \frac{\alpha^2 - d^2}{a^{3|n|+3}}$$

where $d$ is the depth at which the recursion step occurs.

For the special case of computing a circuit representation of $a(n) \equiv a(n, u_n, \alpha)$, we argue that a variant of our procedure can be carried out in polynomial time. First, we already know that the procedure will not fail, hence we can omit to maintain the value of $\epsilon$, since this quantity is used uniquely to check for failure. Then, notice that the choice of $\epsilon$ does not depend on the circuit $n$ (but just on the basis $B$), therefore we can simply pick a valid $\epsilon$ and hard-code it into the procedure.

Hence, since the only values of $\epsilon$ that we encounter during the performance of the procedure are of the form $f^{i+3|n|+3}(k)$ with $0 \leq i < ||n||$, we can construct $B$-circuits to represent these values in time polynomial in $||n||$. Now, stipulate that every time we need to compute $a(n', u', \epsilon)$ for a subcircuit $n'$ of $n$ and a $u'$ in our list, we check if the same computation has already been performed, and if so we simply link to the already constructed subcircuit. Since there are $||n||$ possible subcircuits $n'$, and $u'$ ranges over a set of $||n||$ different values, our modified procedure makes at most $||n||^2$ recursive calls.

Finally, property $(*)$ yields

$$|a(2e - 1) - (2e - 1)| < 1$$

and, observing that $2e - 1$ necessarily represents an odd integer, this implies that $\tilde{c} \equiv a(2e - 1)$ is positive if and only if $\epsilon$ is positive.

VI. PROOF OF THE MAIN RESULTS

Proof of Theorem III.2 By Tarski-Seidenberg, there is a degree $d$ such that each function in $B$ can be written as a $B_d$-circuit, where $B_d$ denotes the language defined at the beginning of Section IV. Hence PosSLP($B_d$) is reducible to PosSLP($B_d$), which, in turn, is reducible to PosSLP by Proposition IV.7.

Proof of Theorem III.3 First assume that there is an integer $g$ such that $g\mid n$ is dense, and we can pick a point $x \in O \cap V(B)$ such that the Hessian matrix $H(g)(x)$ of $g$ at $x$ does not vanish. Now we pick an integer vector $v \in \mathbb{Z}^n$ such that $v^T H(g)(x_0)v$ does not vanish. Clearly $f: \mathbb{R} \to \mathbb{R}$

$$t \mapsto 2g(x + tv) - g(x + 2tv) - g(x)$$

can be represented by a $B$-circuit, and it is of class $C^2$ at $0$. It can be verified by direct computation that $f(0) = f'(0) = 0$ and $f''(0) \neq 0$. Hence either $f$ or $-f$ verifies the hypothesis of Lemma V.2. Therefore we have the first case.

Now assume that all functions in $B$ are piecewise linear, albeit possibly with algebraic coefficients. We will show how to evaluate $B$-circuits in polynomial time. Fix a number field $K$ in which all the coefficients of all the linear pieces of functions in $B$ reside. Fix $e_1, \ldots, e_n \in K$ that generate $K$ as a vector space over $\mathbb{Q}$. Clearly $V(B) \subset K$, hence, in the
evaluation of $B$-circuits, we can restrict the domain of our computation to $K$. We represent each element $x$ of $K$ using the unique vector $\sigma(x) \in \mathbb{Q}^n$ such that $x = \sum \sigma(x)_i e_i$. Now, for each function $f \in B$ we need to know how to compute $\sigma(f(x_1, \ldots, x_m))$ given $\sigma(x_1), \ldots, \sigma(x_m)$. Let $g(x_1, \ldots, x_m)$ be one of the linear pieces that constitute $f$, then $\sigma \circ g \circ \sigma^{-1}$ is a linear map from $(\mathbb{Q}^n)^m$ to $\mathbb{Q}^n$, therefore we decide to compute the linear pieces constituting $f$ simply by matrix multiplication in the rationals, kept as pairs of coprime integers, which in turn are kept in binary. To choose among the pieces, by Tarski-Seidenberg, suffices to evaluate a fixed (depending on $f$) set of rational polynomial conditions on the coefficients of $\sigma(x_1), \ldots, \sigma(x_m)$, which we precompute for the coefficients of functions semialgebraic over $\mathbb{Q}$, again by simple rational arithmetic. Finally, to decide $x > 0$ given $\sigma(x)$, we employ similarly Tarski-Seidenberg to translate this condition into a Boolean combination of polynomial conditions on $\sigma(x_1), \ldots, \sigma(x_n)$, and we evaluate it using rational arithmetic. Summarizing, we precompute the coefficients for the finite number of rational linear functions and polynomials that we will need, this data depends only on $B$, which is fixed. Then we carry out the evaluation as described above. It is easy to check that the algorithm works in polynomial time. This concludes the proof of the second case.

VII. ADDENDA

In this last section we collect a few additional results, observations, questions.

First we would like to discuss the density and continuity assumptions in Theorem III.3. If all the functions in $B$ happen to be constants or unary functions, we can dispense with the aforementioned assumptions because of the following fact, which is a consequence of [WIL04 Corollary 2.2] and Fact III.3

**Fact VII.1.** Let $f : \mathbb{R} \to \mathbb{R}$ be a semialgebraic function. Assume that $f$ maps integers to integers. Then there is $N \in \mathbb{N}$ such that $f|_{(N, +\infty)}$ coincides with a polynomial.

**Theorem VII.2.** Let $B = \{+, -, f_1, \ldots, f_n\}$ be a finite set of functions semialgebraic over $\mathbb{Q}$. Assume that $f_1, \ldots, f_n$ are either constants or unary functions. Then the following dichotomy holds: PosSLP$(B)$ is in $P$ if all the functions in $B$ are piecewise linear when restricted to $V(B)$, otherwise PosSLP and PosSLP$(B)$ are mutually polynomial time Turing reducible.

**Proof.** Along the lines of the proofs of Theorem III.3. The only new case to examine is when $V(B)$ is discrete and there is a function $f_i$ which restricted to $V(B)$ is not piecewise linear. Since $V(B)$ must be an additive subgroup of $\mathbb{R}$, because $+ \in B$, we can assume that $V(B)$ is precisely $\mathbb{Z}$. By Fact VII.1, we have that $f_i$ must coincide with a non-affine polynomial on the integers in a half line. Our goal, now, is to construct, using $f_i$, a new function $g : \mathbb{R} \to \mathbb{R}$, represented as a $B$-circuit, which coincides with a non-affine polynomial on all of $\mathbb{Z}$. Given that, $g$ verifies the hypothesis of Lemma VII.1 and we are done.

The function $g$ is constructed as follows. By Fact VII.1, $f_i$ coincides with a polynomial $p$ on $[N, +\infty)$ and another polynomial $q$ on $]-\infty, -N]$. Let $d$ be the maximum of $\deg(p)$ and $\deg(q)$. Consider

$$h_1 \equiv \Delta_{i=2}^i[f_i], \quad h_2(x) \equiv h_1(2x) - 2h_1(x)$$

As in the proof of Lemma VII.1 it is easy to see that there are $a, b, a', b'$, with at least one of $a$ and $a'$ not null, such that

$$h_2(x) = ax^2 + b \quad \text{for } x \in [-N, -N]$$

$$h_2(x) = a'x^2 + b' \quad \text{for } x \in [N, +\infty]$$

where $N = M + d - 2$. If $a + a' \neq 0$, then

$$h_3(x) \equiv h_2(x) + h_2(-x)$$

coincides with the second degree polynomial

$$(a + a')x^2 + (b + b')$$

for $|x| \geq N$. Therefore, if $a + a' \neq 0$, we can simply choose $g(x) = h_3(N(2x + 1))$. It remains to be considered the case $a = -a'$. In this case we observe that, for $|x| \geq N + 1$

$$h_2(x + 1) - h_2(x - 1) = 4a|x|$$

Therefore we can replace $h_2$ in the above argument with

$$h_2(x) \equiv h_2(h_2(x + 1) - h_2(x - 1))$$

which coincides with $4a^2x^2 + b'\text{or-not } |x|$ sufficiently large.

Unfortunately, we do not have an analogue of Observation VII.1 for multivariate functions. Moreover, the following example shows that, failing either the continuity or the density hypothesis, we can not employ the technique of reducing to a unary function and invoke Lemma VII.1 or VII.2. In particular PosSLP$(B)$ may be equivalent to PosSLP even though all the unary functions represented by $B$-circuits are either constant on a cofinite set, or the identity function.

**Example VII.3.** Let

$$B = \{c_1 \ldots c_n, +, -, x\}$$

$$B' = \{c_1 \ldots c_n, c_1^2 \ldots c_n^2, g+, g-, h+, h-, g\times, h\times\}$$

where for $f : \mathbb{R}^2 \to \mathbb{R}$ we define

$$g_f : \mathbb{R}^4 \to \mathbb{R}$$

$$(x, y, z, t) \mapsto \begin{cases} f(x, y) & \text{if } z = x^2 \text{ and } t = y^2 \\ 0 & \text{otherwise} \end{cases}$$

$$h_f : \mathbb{R}^4 \to \mathbb{R}$$

$$(x, y, z, t) \mapsto \begin{cases} (f(x, y))^2 & \text{if } z = x^2 \text{ and } t = y^2 \\ 0 & \text{otherwise} \end{cases}$$

Clearly PosSLP$(B')$ is polynomial time many-one reducible to PosSLP$(B)$. And the converse is also true. In fact, we can simulate any $B$-circuit $c$ through a $B'$-circuit that keeps for each gate $g$ of $c$ a pair of gates, $g_1$ and $g_2$, computing $c$ and $c^2$ respectively.
On the other hand, for any function \( f : \mathbb{R}^4 \rightarrow \mathbb{R} \) in \( B' \), and for any choice of four linear functions \( a_1 \ldots a_4 : \mathbb{R} \rightarrow \mathbb{R} \), we see that \( f(a_1(x) \ldots a_4(x)) \), as a function of \( x \), is constant on a cofinite set. It follows that any unary function represented by a \( B' \)-circuit, unless it is the function \( x \mapsto x \), must be constant outside of a finite set.

In opposition to continuity and density, we are unsatisfied by the hypothesis that \( B \) contains \( +, - \) in Theorem 3.3 and would like to see it weakened or eliminated. This hypothesis comes directly from Lemma 2 where we need \( +, - \) to manipulate the graph of \( f \). It is conceivable that, as we can simulate \( \times \) killing the constant and first degree terms of a second order approximation of \( f \), we may be able to simulate some linear function killing the constant and second degree terms, at least if \( f \) is generic enough.

The initial motivation of the present work has been an ongoing attempt by the author and Manuel Bodirsky to investigate constraint satisfaction problems over the reals, continuing the work initiated by [BJO12]. For technical reasons, due to the convexity requirement proven in [BJO12], it would be desirable to assess the computational complexity of the problem of comparing two \( B \)-circuits (as opposed to one \( B \)-circuit and 0) in a basis \( B \) not containing the \( - \) function. For the case of the basis \( \{0, 1, +, -, x\} \), it is an observation that the comparison of circuits on the basis \( \{0, 1, +, x\} \) is polynomial time equivalent to PosSLP. Nevertheless, we can not say whether the same holds in a more general situation.

Acknowledgement

The author is indebted to Manuel Bodirsky for pointing out the direction that led to the present work, for numerous discussions and suggestions, and for commenting on preliminary versions. We would also like to express our gratitude to Eleonora Bardelli for interesting discussions and comments, and to Vincenzo Mantova for suggesting references on Weil heights.

References

[ABKM09] E. Allender, P. Bürgisser, J. Kjeldgaard-Pedersen, and P. B. Miltersen, “On the complexity of numerical analysis,” SIAM J. Comput., vol. 38, no. 5, pp. 1987–2006, 2008/09. [Online]. Available: http://dx.doi.org/10.1137/S0097539702466199

[Akt89] K. E. Atkinson, *An Introduction to Numerical Analysis*, 2nd ed. New York: John Wiley & Sons Inc., 1989.

[BC06] M. Braverman and S. Cook, “Computing over the reals: foundations for scientific computing,” Notices Amer. Math. Soc., vol. 53, no. 3, pp. 318–329, 2006.

[BCSS98] L. Blum, F. Cucker, M. Shub, and S. Smale, *Complexity and Real Computation*. New York: Springer-Verlag, 1998, with a foreword by Richard M. Karp.

[BJO12] M. Bodirsky, P. Jonsson, and T. v. Oertzen, “Essential convexity and complexity of semi-algebraic constraints,” Log. Methods Comput. Sci., vol. 8, no. 4, pp. 4:5, 25, 2012. [Online]. Available: http://dx.doi.org/10.2168/LMCS-8(4:5)2012

[Bour66] N. Bourbaki, *Elements of Mathematics. General Topology. Part 1*. Paris: Hermann, 1966.

[Bra05] M. Braverman, “On the complexity of real functions,” in *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, 2005, pp. 155–164.

[Ers79] A. P. Ershov, “Abstract computability on algebraic structures,” in *Algorithms in Modern Mathematics and Computer Science* (Urgench, 1979), ser. Lecture Notes in Comput. Sci. Berlin: Springer, 1981, vol. 120, pp. 137–200.

[EY10] K. Etesami and M. Yannakakis, “On the complexity of Nash equilibria and other fixed points,” SIAM J. Comput., vol. 39, no. 6, pp. 2531–2597, 2010. [Online]. Available: http://dx.doi.org/10.1137/080720826

[FM09] H. Friedman and R. Mansfield, “Algorithmic procedures,” Trans. Amer. Math. Soc., vol. 332, no. 1, pp. 297–312, 1992. [Online]. Available: http://dx.doi.org/10.1090/S0002-9947-1992-1137613-5

[GJ76] M. R. Garey, R. L. Graham, and D. S. Johnson, “Some NP-complete geometric problems,” in *Eighth Annual ACM Symposium on Theory of Computing* (Hershey, Pa., 1976). Assoc. Comput. Mach., New York, 1976, pp. 10–22.

[LPY05] C. Li, S. Pion, and C. K. Yap, “Recent progress in exact geometric computation,” J. Log. Algebra. Program., vol. 64, no. 1, pp. 85–111, 2005. [Online]. Available: http://dx.doi.org/10.1016/j.jalgebra.2004.07.006

[PD01] A. Prestel and C. N. Delzell, *Positive Polynomials*, ser. Springer Monographs in Mathematics. Berlin: Springer-Verlag, 2001, from Hilbert’s 17th problem to real algebra.

[Pou95] B. Poznat, *Les petits cailloux* [Light of Logic and Knowledge], 3. Lyon: Aléas, 1995, une approche modèle-théorique de l’algorithmie. [A model-theoretic approach to algorithms].

[Skl86] J. H. Silverman, *The Arithmetic of Elliptic Curves*, ser. Graduate Texts in Mathematics. New York: Springer-Verlag, 1986, vol. 106.

[Tiw92] P. Tiwari, “A problem that is easier to solve on the unit-cost algebraic RAM,” J. Complexity, vol. 8, no. 4, pp. 393–397, 1992. [Online]. Available: http://dx.doi.org/10.1016/0885-064X(92)90003-T

[TV08] S. P. Tarasov and M. N. Vyalyi, “Semidefinite programming and arithmetic circuit evaluation,” Discrete Appl. Math., vol. 156, no. 11, pp. 2070–2078, 2008. [Online]. Available: http://dx.doi.org/10.1016/j.dam.2007.04.023

[vdD98] L. van den Dries, *Tame Topology and O-Minimal Structures*, ser. London Mathematical Society Lecture Note Series. Cambridge: Cambridge University Press, 1998, vol. 248. [Online]. Available: http://dx.doi.org/10.1017/CBO9780511525919

[Wal00] M. Waldschmidt, *Diophantine Approximation on Linear Algebraic Groups*, ser. Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]. Berlin: Springer-Verlag, 2000, vol. 326, transcendence properties of the exponential function in several variables.

[Wil04] A. J. Wilkie, “Diophantine properties of sets definable in o-minimal structures,” J. Symbolic Logic, vol. 69, no. 3, pp. 851–861, 2004. [Online]. Available: http://dx.doi.org/10.2178/jsl/1096901771