Fixed Points, Nash Equilibria, and the Existential Theory of the Reals

Marcus Schaefer
School of Computing
DePaul University
243 South Wabash
Chicago, Illinois 60604, USA
mschaefer@cdm.depaul.edu

Daniel Štefankovič
Computer Science Department
University of Rochester
Rochester, NY 14627-0226
stefanko@cs.rochester.edu

Abstract

We introduce a new complexity class $\exists\mathbb{R}$ based on the existential theory of the reals, and show that it is robust from a computational point of view. The class $\exists\mathbb{R}$ captures the complexity of several well-known problems; we show that the complexity of decision variants of fixed-point problems, including Nash equilibria, are captured by this class, complementing work by Etessami and Yannakakis [4].

Keywords: Fixed points, Brouwer, existential theory of the real numbers, Nash equilibrium, computational complexity

1 Introduction

Many computational problems in geometry, graph drawing and other areas can be shown decidable using the (existential) theory of the real numbers, including the rectilinear crossing number, the Steinitz problem, and finding a Nash equilibrium; what is less often realized, though there are some exceptions, is that the existential theory of the reals captures the computational complexity of many of these problems precisely. In previous papers, we investigated some geometric problems related to graph drawing [16, 17]. In the current paper, we look at fixed point-problems and Nash equilibria, complementing work of Etessami and Yannakakis [4].

From an algebraic point of view, there are two ways to define the existential theory of the reals depending on whether we allow equality or not; for example, the rectilinear crossing number problem can be expressed as a system of strict inequalities, and, as a consequence, a drawing realizing

the rectilinear crossing number of a graph can be assumed to have vertices with rational coordinates (even if some of them may require exponential precision); similarly, intersection graph problems can typically be captured by strict inequalities (for example, the problems in [16], including segment intersection graphs). On the other hand, fixed-point problems need equality to be modeled in the existential theory of the reals, and so solution sets do not necessarily contain rational points: the fixed point of $f(x) = 2/x$ is $\sqrt{2}$. In Section 2 we prove the rather unexpected result that from a computational point of view, these two variants of the existential theory of the reals are the same, justifying the introduction of a single complexity class $\exists\mathbb{R}$. In Section 3 we then show that several fixed-point problems are complete for this class. Together with the results from earlier papers this already gives us a sizable collection of complete problems for $\exists\mathbb{R}$ from many different areas (see [12, 7, 9, 16, 10, 17], for example; a survey on the topic is in preparation [15]).

We assume that the reader is familiar with basic notions of computational complexity, including polynomial-time many-one reducibilities and complexity classes such as **NP**, and **PSPACE** [13, 18].

2 The Existential Theory of the Reals

The existential theory of the reals, ETR, is the set of true sentences of the form

$$(\exists x_1, \ldots, x_n)\ \varphi(x_1, \ldots, x_n),$$

where φ is a quantifier-free $(\vee, \wedge, \neg)$-Boolean formula over the signature $(0, 1, +, *, <, \leq, =)$ and the sentence is interpreted over the universe of real numbers.[1]

It was first shown by Tarski that this theory is decidable, but the running time of his decision procedure was not elementary (that is, bounded above by a tower of exponentials of fixed height); the first algorithm solving the problem in elementary (double-exponential) time was discovered by Collins in the 70s, using cylindrical algebraic decomposition. In the late 80s, Canny showed that the problem is solvable in **PSPACE**. For a detailed survey, see [11]; for experimental comparisons of running times, see [6].

We will find it useful to distinguish two special cases of ETR. Let INEQ be the subset of ETR, in which we do not allow $\vee, \neg$ and $=$, that is φ is a

[1] When writing formulas in the existential theory of the reals, we will freely use integers and rationals, since these can easily be eliminated without affecting the length of the formula substantially.

conjunction of atoms of the form $s < t$ and $s \leq t$ ($s = t$ can be expressed as $s \leq t \wedge t \leq s$ so not allowing equality is not a real restriction). Furthermore, let STRICT INEQ be the subset of INEQ, in which we do not allow $\leq$, that is, φ is a conjunction of strict inequalities $s < t$.

Following our first impulse as complexity theorists we use STRICT INEQ and INEQ to define complexity classes $\exists_< \mathbb{R}$ and $\exists_= \mathbb{R}$ as the downward closures of these problems under polynomial-time many-one reductions; with this definition $\exists_< \mathbb{R} \subseteq \exists_= \mathbb{R}$ and there seems to be evidence that these two classes are different: solutions to an INEQ-type problem can require irrational numbers, e.g. $x^2 = 2$, while solutions to STRICT INEQ can always be perturbed slightly to make them rational. These difference are of an algebraic nature and, in a slightly surprising twist of events, do not affect the computational complexity of these problems. It turns out that $\exists_< \mathbb{R} = \exists_= \mathbb{R}$ as we will see in Section 2.2. In other words, INEQ polynomial-time many-one reduces to STRICT INEQ.

Note that $\mathbf{NP} \subseteq \exists_< \mathbb{R}$, since we can express satisfiability of a Boolean formula in $\exists_< \mathbb{R}$. For example, $\varphi = (x \vee \overline{y} \vee z) \wedge (\overline{x} \vee y \vee z) \wedge (\overline{x} \vee \overline{y} \vee \overline{z})$ is equivalent to

$$\begin{aligned}(\exists x, y, z)[\ &(-\varepsilon < x < 2) \wedge (-\varepsilon < y < 2) \wedge (-\varepsilon < z < 2) \\ &\wedge (x(1-y)z) + ((1-x)yz) + ((1-x)(1-y)(1-z)) < \varepsilon],\end{aligned}$$

if we choose $\varepsilon = 1/8(1 + 4m) = 1/104$ where m is the number of clauses, so $m = 3$ in the example. If the formula is satisfiable, then we assign a variable the value 0 if it is true and 1 otherwise, so that the sum becomes 0 which is less than ε; in the example: $x = y = 0$ and $z = 1$ will do. For the reverse direction, assume that we have x, y, and z satisfying the real formula. Each term of the sum is at least $-\varepsilon \cdot 2^2 = -4\varepsilon$; so the whole sum is at least $-4m\varepsilon \geq -12/104$, but then for the whole sum to be less than $1/104$ every term must be less than $1/104 + 12/104 = 1/8$. Now each term is the product of three factors, so at least one factor must be at most $(1/8)^{1/3} = 1/2$. Let the corresponding variable be true if the factor is of the form x and false if it is of the form $1 - x$. This yields a satisfying assignment of the original Boolean formula φ.

So, with respect to classical complexity classes, we can summarize our present knowledge of the existential theory of the reals by

$$\mathbf{NP} \subseteq \exists_< \mathbb{R} \subseteq \exists_= \mathbb{R} \subseteq \mathbf{PSPACE},$$

where the last implication is due to Canny's result [2].

2.1 Semi-Algebraic Sets of Bounded Complexity

Semi-algebraic sets are solution sets to systems of polynomial equalities and inequalities; more formally, a set $S \subseteq \mathbb{R}^n$ is *semi-algebraic* if there is a $(\vee, \wedge, \neg)$-Boolean quantifier-free formula over the signature $(0, 1, +, *, <, \leq, =)$ so that $S = \{x \in \mathbb{R}^n : \varphi(x)\}$. We use $|\varphi|$ to denote the *length* of φ, that is, the number of bits necessary to write down φ. The *(bit)-complexity of a semi-algebraic set* is the shortest length of any formula defining the set. We write $[n]$ as an abbreviation for the set $\{1, \ldots, n\}$.

Our goal in this section is to establish the following two lemmas showing some limitations on semi-algebraic sets of bounded complexity.

Lemma 2.1. *If a bounded semi-algebraic set has complexity at most L, then all its points have distance at most $2^{L^{cn^2}}$ from the origin for some absolute constant $c > 0$.*

The *distance* $d(A, B)$ between two sets $A, B \subseteq \mathbb{R}^n$ is defined as $d(A, B) = \inf\{d(a, b) : a \in A, b \in B\}$, where $d(a, b)$ is the Euclidean distance between two points.

Lemma 2.2. *Suppose two semi-algebraic sets have total complexity at most L. If the two sets have positive distance (for example, if they are compact and disjoint), then they have distance at least $1/2^{L^{cn^3}}$ for some absolute constant $c > 0$.*

The remainder of this section is devoted to the proofs of these two lemmas, based on two classical results from algebraic geometry and logic. The first, due to Grigor'ev and Vorobjov, is a fundamental result on polynomials. Recall that the *total degree* of a (multivariate) polynomial $f : \mathbb{R}^n \to \mathbb{R}$ is the maximum over the sum of the variable exponents in each monomial term occurring in f. E.g. $f(x, y, x) = 5x^7y^2 - 2x^3yz^6$ has total degree $3 + 1 + 6 = 10$.

Theorem 2.3 (Grigor'ev, Vorobjov [5][2])**.** *If $f_1, \ldots, f_k : \mathbb{R}^n \to \mathbb{R}$ are polynomials each of total degree at most d and coefficients of bit-length at most L; then every connected component of $\{x \in \mathbb{R}^n : f_1(x) \geq 0, \ldots, f_k(x) \geq 0\}$ contains a point of distance less than $2^{Ld^{cn}}$ from the origin, for some absolute constant $c > 0$.*

We adapt this result slightly for our purposes before proceeding.

[2]The theorem can also be found in [1, Theorem 13.15] though the statement contains a typo in the radius of the the ball.

Corollary 2.4. *A non-empty semi-algebraic set of complexity at most L contains a point of distance at most $2^{L^{cn}}$ from the origin, for some absolute constant $c > 0$.*

Proof. Suppose S is a non-empty semi-algebraic set given by a quantifier-free formula $\varphi(x)$ of length at most L, where x ranges over $\mathbb{R}^n$. We simplify φ as follows: we eliminate negation by pushing it to the atomic level, and then, if necessary, replacing inequalities; e.g. $\overline{s < t}$ becomes $s = t \vee s > t$. Hence, we can assume that φ is a monotone formula. Since S is non-empty, there is a $p \in \mathbb{R}^n$ for which $\varphi(p)$ is true. If φ contains a disjunction $\alpha \vee \beta$ and at least one of α or β, say α, is satisfied for $x = p$, we replace $\alpha \vee \beta$ with α. Otherwise, we replace it with $\perp$ (false). In either case, the resulting formula remains satisfiable (since it is true for $x = p$). In this way, we can eliminate all disjunctions from φ, so we are left with a conjunction of polynomial equalities and inequalities. Let $s_i > t_i$, $i \in [k]$, be the remaining strict inequalities. Replace them with $s_i \geq t_i$, $i \in [k]$ and $z^2\ \Pi_{i\in[k]}(s_i - t_i) = 1$ adding a single new variable z. Call the resulting formula φ'. Then $|\varphi'| \leq c'|\varphi|$ for some fixed $c' > 0$ and any x satisfying φ' also satisfies φ (by the process of construction). We can now apply Theorem 2.3 to φ' to conclude that there is a solution x for φ' and thus φ with $d(x, 0) < 2^{L^{c''n}}$ for some constant $c'' > 0$ (using the estimate $d \leq c'L$ on the degree of the polynomials). □

The second result we need is a quantifier elimination result. Tarski showed that semi-algebraic sets are closed under projection: if $S \subseteq \mathbb{R}^n$ is a semi-algebraic set defined by a quantifier-free formula φ, then $\{(x_2, \ldots, x_n) \in \mathbb{R}^{n-1} : (\exists x_1)\ \varphi(x_1, \ldots, x_n)\}$ is also semi-algebraic; that is, it equals $\{(x_2, \ldots, x_n) \in \mathbb{R}^{n-1} : \psi(x_2, \ldots, x_n)\}$ for some quantifier-free formula ψ which can be obtained from φ effectively (though it will be much larger). Indeed, more is true: $(\exists x_1)\ \varphi(x_1, \ldots, x_n) \equiv \psi(x_2, \ldots, x_n)$. The procedure of obtaining ψ from φ is known as *quantifier elimination*. There are much stronger results on quantifier elimination by now; we make use of an algorithm due to Basu, Pollack and Roy [1, Theorem 14.21]; we do not state this result in its full generality, and we weaken it by removing the dependence on the degree of the polynomials in φ.

Theorem 2.5 (Basu, Pollack, Roy [1])**.** *Given a formula*

$$\Phi(y_1, \ldots, y_\ell) = (\exists x_1, \ldots, x_k)\ \varphi(x_1, \ldots, x_k, y_1, \ldots, y_\ell),$$

we can find an equivalent quantifier-free formula $\psi(y_1, \ldots, y_\ell)$ of length $|\varphi|^{c(k\ell+1)}$ in time $|\varphi|^{c(k\ell+1)}$, for some fixed constant $c > 0$.

This theorem follows from Theorem 14.21 in [1] for a single block ($\omega = 1$). We use rather rough estimates: $s \leq |\varphi|$ and $d \leq |\varphi|$ which gives $I \leq |\varphi|^{O(k\ell)}$, $J_i \leq |\varphi|^{O((k))}$, and $N_{ij} \leq |\varphi|^{O((k))}$.

Combining the bounds of Grigor'ev and Vorobjov with the quantifier elimination result, we obtain proofs of Lemma 2.1 and Lemma 2.2.

Proof of Lemma 2.1. Suppose we have a bounded semi-algebraic set S given by a quantifier-free formula φ of length at most L in n free variables. Then the formula

$$y > 0 \wedge (\forall x \in \mathbb{R}^n)[\varphi(x) \rightarrow \sum_{i \in [n]} x_i^2 \leq y^2]$$

is true if and only if y is an upper bound on $d(x, 0)$ for every $x \in S$. Applying Theorem 2.5, we obtain an equivalent formula $\psi(y)$ of length at most L^{cn} for some fixed $c > 0$. Since S is bounded, there is some y so that $\psi(y)$ is true. Then by Corollary 2.4 there is some y for which $\psi(y)$ is true and $y < 2^{L^{c'n^2}}$. □

Proof of Lemma 2.2. Suppose we have two semi-algebraic sets given by quantifier-free formulas φ and ψ in n free variables with total length at most L. The distance u between the two sets is definable as follows:

$$\begin{aligned} u \geq 0 \wedge (\forall x \in \mathbb{R}^n)(\forall y \in \mathbb{R}^n)(\forall \varepsilon > 0)(\exists x' \in \mathbb{R}^n)(\exists y' \in \mathbb{R}^n) \\ [(\overline{\varphi(x)} \vee \overline{\psi(x)} \vee (x_1 - y_1)^2 + \cdots + (x_n - y_n)^2 \geq u^2) \\ \wedge \quad (\varphi(x') \wedge \varphi(y') \wedge (x'_1 - y'_1)^2 + \cdots + (x'_n - y'_n)^2 \leq u^2 + \varepsilon)]. \end{aligned}$$

Applying Theorem 2.5 twice, we obtain an equivalent formula $\tau(u)$ of length at most L^{cn^2} for some $c > 0$ that defines the distance between the two sets. Consider $S := \{u' \in \mathbb{R} : \tau(u) \wedge uu' = 1\}$. Apply Corollary 2.4 to conclude that there is a $u' \in S$ with $u' < 2^{L^{c'n^3}}$ for some constant $c' > 0$. But since $uu' = 1$ this implies $u \geq 1/2^{L^{cn^3}}$ completing the proof. □

2.2 $\exists_< \mathbb{R}$ and $\exists_= \mathbb{R}$

We have defined three variants of the existential theory of the reals, ETR, INEQ, and STRICT INEQ; in this section we will see several additional variants, and show that they are all computationally equivalent. In particular, it will follow that $\exists_< \mathbb{R} = \exists_= \mathbb{R}$, which is a bit of a surprise, since algebraically these two classes differ.

We first show that ETR and INEQ are equivalent; it seems easiest to use FEASIBLE which restricts INEQ to formulas not containing $<$ and $\leq$; in

other words, FEASIBLE asks whether a system of multivariate polynomials is *feasible*, that is, has a common root.

Given an instance of ETR as a quantifier-free Boolean formula φ, we can translate it into a formula φ' in conjunctive normal form (that is, a conjunction of disjunctions of literals) in polynomial time and increasing the length by at most a linear factor so that φ and φ' are equisatisfiable (that is, neither or both are satisfiable; this standard translation is due to Tseitin [19, 8]). We can also eliminate negations by absorbing them at the atomic level: e.g. $\overline{s < t}$ becomes $s \geq t$, and $\overline{s = t}$ turns into $s < t \vee t < s$. Now replace all inequalities of type $s \leq t$ by $s < t \vee s = t$; then replace inequalities of type $s < t$ by $(t - s) * x^2 = 1$, where x is a new variable, and finally turn equalities $s = t$ into $s - t = 0$. Now each clause of the formula is a disjunction of equalities: $s_1 = 0 \vee \cdots \vee s_k = 0$ which we can replace by a single equality $s_1 \cdots s_k = 0$; the resulting formula is a conjunction of equalities, and therefore a positive instance of FEASIBLE if and only if φ was a positive instance of ETR. Since FEASIBLE is a special case of INEQ, this in particular shows that INEQ and ETR are polynomial-time equivalent. We can replace the conjunction $t_1 = 0 \wedge \cdots \wedge t_\ell = 0$ by a single condition $t_1^2 + \cdots + t_{\ell^2} = 0$, showing that even testing whether a single multivariate polynomial has a root is as hard as ETR.

The problem FIXED asks whether a set of polynomials $f_i : \mathbb{R}^n \to \mathbb{R}$, $i \in [n]$, has a fixed point, that is $x \in \mathbb{R}^n$ so that $(f_1(x), \ldots, f_n(x)) = x$. Obviously, FIXED is a special case of INEQ, so to show it equivalent to the other problems it is enough to reduce FEASIBLE to it. So let g be an n-variate polynomial (we saw above that this special case is hard for FEASIBLE already), and let $f_i(x) = g(x) + x_i$. Then g has a root if and only if $(f_i)_{i \in [n]}$ has a fixed point.

None of the previous reductions are original, though we are not aware of them having been stated in this particular form in the literature before.

Theorem 2.6. *The following problems are polynomial-time equivalent:* ETR*,* FIXED*,* INEQ*,* FEASIBLE*,* STRICT INEQ*.*

Proof. The only missing part is showing that FEASIBLE can be reduced to STRICT INEQ. So suppose we are given an n-variate polynomial g and ask whether there is an $x \in \mathbb{R}^n$ so that $g(x) = 0$. Let L be the bit-complexity of g. By Theorem 2.3 we know that if there is a solution, that solution has distance at most $R = 2^{L^{cn}}$ from the origin (bounding d by L). Consider the two semi-algebraic sets $\{(z, x) \in \mathbb{R}^{n+1} : g(x) = z, \sum_{i \in [n]} x_i^2 \leq R^2\}$ and $\{(z, x) \in \mathbb{R}^{n+1} : z = 0, \sum_{i \in [n]} x_i^2 \leq R^2\}$. If these two sets do not intersect,

they have positive distance (both being compact), and, by Lemma 2.2, that distance is at least $1/2^{L^{cn^3}}$. Hence, $g = 0$ is equivalent to the system

$$-\delta < g, g < \delta, \delta < 1/2^{L^{cn^3}}$$

of strict inequalities being solvable. The last inequality can be replaced by a sequence of at most $cn^3 \log L$ inequalities (using repeated squaring), so we have shown that FEASIBLE can be reduced to STRICT INEQ. Note that this reduction does not (and cannot) maintain the realization space of the system (the set of solutions). □

As a corollary we obtain:

Corollary 2.7. $\exists_< \mathbb{R} = \exists_= \mathbb{R}$.

The corollary allows us to simplify our notation and call our new complexity class simply $\exists\mathbb{R}$. Our computational world now looks as follows:

$$\mathbf{NP} \subseteq \exists\mathbb{R} \subseteq \mathbf{PSPACE}.$$

3 Fixed Points and the Nash Equilibrium

How hard is it to find a fixed-point of a continuous function from a convex, compact set to itself? A big difference to the problem FIXED we discussed earlier is that such functions always have a fixed point by the Brouwer Fixed-Point Theorem. There are several versions of this problem that can be asked (see the detailed discussion by Etessami and Yannakakis [4]). We start with the decision version of the problem and discuss variants and the Nash Equilibrium problem later.

We consider functions computed by straight-line programs. A *straight-line program (SLP)* is a sequence of assignments of the form $S_i := c, c \in \{-1, -1/2, 0, 1/2, 1\}$, $S_i := x_j, i \in [\ell]$ or $S_i := S_j \circ S_k$, where $1 \leq j, k < i \leq \ell$ and $\circ \in \{+, -, *\}$; ℓ is the *length* of the program.[3] We can think of the straight-line program as a succinct way of describing a multivariate polynomial in variables $x_j, j \in [n]$. A *straight-line program for a function*

[3]We could allow arbitrary assignments $S_i := c$, where $c \in \mathbb{Q}$ or $c \in [-1, 1] \cap QN$, the following results would still be true if we redefine length in this case to include the number of bits needed to write down any rational constants used. We decided to stick with the more restrictive model in which only $\{-1, -1/2, 0, 1/2, 1\}$ are allowed as constants. We will see presently that this is not a real restriction as far as fixed point computations are concerned: allowing division does not yield any additional computational power.

$f = (f_i)_{i\in[m]} : U \to V$, where $U \subseteq \mathbb{R}^n$, $V \subseteq \mathbb{R}^m$, is a straight-line program in which the first n assignments are $S_i := x_i$, and the last m assignments calculate $f_i(x_1, \ldots, x_n)$ so that $S_{\ell-m+i} = f_i(x_1, \ldots, x_n)$, for $i \in [m]$. In an *extended straight-line program (ESLP)* we also allow operations /, max, min, and $\sqrt[k]{\ }$. (The definition in this case implies that for inputs in U no division by zero or even roots of negative numbers occur.)

Let $B_n(x, r)$ be the closed ball around $x \in \mathbb{R}^n$ of radius r in the ℓ^∞-metric; in other words, $B_n(x, r)$ is an n-dimensional box; for example, $B_n(0, 0.5)$ is the unit cube centered at the origin. We define the following computational version of the Brouwer fixed-point problem:

BROUWER
Given: An SLP for a function $f : B_n(0, 1) \to B_n(0, 1)$, $x \in \mathbb{Q}^n$, $r \in \mathbb{Q}$.
Question: Does f have a fixed-point in $B_n(x, r)$?

We write F_f for the set of fixed points of a function f in its domain. The following lemma shows that ESLPs have no edge on SLPS with respect to capturing sets of fixed points: for each ESLP there is an SLP that has essentially the same set of fixed points. We write $\mathbf{1}$ for the vector consisting of all ones (with appropriate dimension).

Lemma 3.1. *If $f : B_n(0, 1) \to B_n(0, 1)$ is a function given by an ESLP, then we can construct in polynomial time an SLP for a function $g : B_{n'}(0, 1) \to B_{n'}(0, 1)$, $n' \geq n$, so that $F_f \cup \{\mathbf{1}\} = \pi_n(F_g)$, where $\pi_n : \mathbb{R}^{n'} \to \mathbb{R}^n$ projects a vector on its first n coordinates. Moreover, we can ensure that $F_g \subseteq (F_f \times B_{n'-2n}(0, 1/2) \times F_f) \cup \{\mathbf{1}\}$.*

Remark 3.2. (*i*) There is nothing special about adding $\mathbf{1}$ as a fixed point when going from f to g, the construction we used could be adapted to add any point in $[-1, 1]^{n'}$ describable by an SLP. So one way to eliminate that point is by making this added fixed-point a fixed-point of f, however that would require the ability to find some (any) fixed-point of f and that we will probably not be able to do in polynomial time: Papadimitriou showed that this problem is **PPAD**-complete [14]. It seems possible that there is a different construction which obviates the need to add an extra point as fixed-point.

(*ii*) As it is, Lemma 3.1 tells us that finding an arbitrary fixed point of a function $f : B_n(0, 1) \to B_n(0, 1)$ specified by an ESLP can be reduced to finding at least two (arbitrary) fixed points of a function $g : B_{n'}(0, 1) \to B_{n'}(0, 1)$ specified by an SLP.

Proof of Lemma 3.1. Let $(S_i)_{i\in[\ell]}$ be the ESLP computing f. The first n instructions have the form $S_i := x_i$, $i \in [n]$, and the last n variables, $S_{\ell-n+1}, \ldots, S_\ell$ contain the outputs. We can assume that divisions are of the form $S_i := 1/S_k$; moreover, since $\max\{x, y\} = (x + y + |x - y|)/2$, $\min\{x, y\} = x + y - \max\{x, y\}$ and $|x| = \sqrt{x^2}$, we can assume that the ESLP does not contain max or min. Finally, we replace any instruction $S_i := \sqrt[k]{S_j}$ for even k with two instructions: $A := \sqrt[2k]{S_j}$, $S_i := A * A$; here A is a new variable we insert just before S_i (and which is only used to calculate S_i). This modified ESLP will calculate S_i (and thus the rest of the program) correctly as the positive kth root of S_j, independently of whether $\sqrt[2k]{S_j}$ returns the positive or negative $2k$th root of S_j.

We show how to create an SLP computing g as in the statement of the lemma. We can consider each S_i as a function $S_i(x_1, \ldots, x_n)$ from $\mathbb{R}^n$ to $\mathbb{R}$; we know that every S_i is well-defined (no divisions by zero or even roots of negative arguments), but while the S_i calculating the output are restricted to values in $[-1, 1]$, the intermediate values can be large; however, since the input set is compact, there is an M so that $S_i(x_1, \ldots, x_n) \leq M/2$ for all $i \in [\ell]$ and $(x_1, \ldots, x_n) \in [-1, 1]^n$. Consider $\mathcal{S} := \{(S_1(x), \ldots, S_\ell(x)) : x \in [-1, 1]^n\}$. Note that $\mathcal{S}$ is a semi-algebraic set (all instructions can be rewritten as polynomial (in)equalities, e.g. $S_i := \sqrt[2k]{S_j}$ becomes $S_i \geq 0 \wedge S_i^{2k} - S_j = 0$ and $S_i := \sqrt[2k+1]{S_j}$ becomes $S_i^{2k+1} - S_j = 0$); hence, by Lemma 2.1, we can choose $M = 2^{\ell^{\lceil cn^2 \rceil}}$ (for some $c > 0$).[4] By adding at most $O(\lceil cn^2 \rceil \log \ell)$ instructions to the SLP for g, we can assume that we have a variable containing the value $1/M$ (use rational constant $1/2$ as a base for repeated squaring). This SLP will also give us $1/4$ and $1/16$, so we are allowed to use $1/M$, $1/4$ and $1/16$ in an SLP (since they can be expressed without using division); we will write $(1/M)$, $(1/4)$ and $(1/16)$ to make it clear that these expressions refer to a variable containing the respective value.

The new SLP will have $n + \ell + 1$ new input variables $y_0, y_1, \ldots, y_\ell$, and $z_1, \ldots, z_n$. For a fixed point $g(x, y, z) = (x, y, z)$ we will ensure that $y_i = S_i(x)/M$ for $i \in [\ell]$ and $z_i = S_{\ell-n+i}$ for $i \in [n]$ (unless $y_0 = 0$), so the y_i simulate the calculations of the straight-line program and the z_i the output.

Based on the instructions in the ESLP for f we add compound instruc-

[4] A similar bound could also be proved directly by arguing that the range of every S_i is an interval.

tions to the SLP for g as follows:

$$\begin{aligned}
S_i := c &\rightarrow N_i := 1 - q(y_i - c * (1/M)) \\
S_i := x_j &\rightarrow N_i := 1 - q(y_i - x_j * (1/M)) \\
S_i := S_j + S_k &\rightarrow N_i := 1 - q(y_i - (y_j + y_k)) \\
S_i := S_j * S_k &\rightarrow N_i := 1 - q(y_i * (1/M) - (y_j * y_k)) \\
S_i := 1/S_j &\rightarrow N_i := 1 - q(y_i * y_j - 1 * (1/M)^2) \\
S_i := \sqrt[k]{S_j} &\rightarrow N_i := 1 - q(y_i^k - y_j * (1/M)^{k-1}),
\end{aligned}$$

where $q(x) = x^2 * (1/16)$. Each of the compound instructions for the N_i can be replaced by a small number of SLP instructions.[5] Finally, we need to create $n+\ell+1$ output instructions. For simplicity, let us call the variables containing outputs of the SLP X_i, Z_i, $i \in [n]$ and Y_i, $i \in [0{:}\ell]$. For every $S_{\ell-n+i}$, the output variable equated with x_i in the original fixed point problem $f(x) = x$, we add the following instructions: $O_i = 1 - q(y_{\ell-n+i} - z_i * (1/M))$, $i \in [n]$.

Finally, we add instructions $X_i = z_i$, $i \in [n]$, $Y_0 = 1 - (1 - y_0) * \Pi_{i\in[\ell]} N_i * O_i$, and $Y_i = p(y_0) * y_i + 1 - p(y_0)$ and $Z_i = p(y_0) * z_i + 1 - p(y_0)$ where $p(x) = 1 - (x - (1/4))^2 * (1/4)$.

This completes the SLP computing g. We need to show that g is a function from $[-1,1]^{2n+\ell+1}$ to $[-1,1]^{2n+\ell+1}$. This is obvious for X_i since $z_i \in [-1,1]$. $N_i, O_i \in [0,1]$ by the choice of q: the terms to which q is applied all lie in $[-3,3]$ (since $y_i, x_j, z_k \in [-1,1]$), so by applying q, we obtain numbers in $[0,1]$. It follows that $Y_0 \in [-1,1]$. Finally, $Y_i = p(y_0) * y_i + 1 - p(y_0) \leq p(y_0) + 1 - p(y_0) = 1$, and $Y_i \geq 1 - 2p(y_0) \geq 1 - 2 = -1$, and the same analysis shows that $Z_i \in [-1,1]$.

Fix (x,y,z) with $g(x,y,z) = (x,y,z)$. If $y_0 = 1$, then $p(y_0) < 1$. Since $y_i = Y_i = p(y_0) * y_i + 1 - p(y_0)$, we have $y_i(1 - p(y_0)) = 1 - p(y_0)$ and thus $y_i = 1$ for $i \in [\ell]$; by the same argument $z_i = 1$ for $i \in [n]$, and thus $x_i = 1$ for $i \in [n]$. If, on the other hand, $y_0 \neq 1$, then because of $y_0 = Y_0 = 1 - (1 - y_0) * \Pi_{i\in[\ell]} N_i * O_i$ we have $1 - y_0 = (1 - y_0) * \Pi_{i\in[\ell]} N_i * O_i$ and thus $1 = \Pi_{i\in[\ell]} N_i * O_i$ which implies $N_i = 1$ for all $i \in [\ell]$ and $O_i = 1$ for all $i \in [n]$. We distinguish cases; note that $q(x) = 0$ implies that $x = 0$ by choice of q.

[5] For example, $N_i := 1 - q(y_i - c * (1/M))$ can be calculated as $A_1 := -c$ ($c \in \{-1, -1/2, 0, 1/2, 1\}$), $A_2 := A_1 * (1/M)$, $A_3 = y_i + A_2$, $A_4 := A_3 * A_3$, $A_5 := A_4 * (1/4)$, $A_6 := -1$, $A_7 := A_6 * A_5$, $A_8 := 1$, $A_9 := A_8 + A_7$. The calculation of $N_i := 1 - q(y_i^k - y_j * (1/M)^{k-1})$ can be performed using $O(\log k)$ many instructions (using repeated squaring).

- $S_i := c$: since $N_i = 1$ we get that $q(y_i - c * (1/M)) = 0$ and thus $y_i = c/M$,
- $S_i := x_j$: since $N_i = 1$ we get that $q(y_i - x_j/M) = 0$ and thus $y_i = x_j/M$, or, in other words, $M * y_i = x_j$,
- $S_i := S_j + S_k$; we get $y_i = y_j + y_k$,
- $S_i := S_j * S_k$ we get $y_i/M = (y_j * y_k)$, so $M * y_i = (M * y_j) * (M * y_k)$,
- $S_i := 1/S_j$; we get $y_i * y_j = 1/M^2$, so $M * y_i = 1/(M * y_j)$,
- $S_i := \sqrt[k]{S_j}$; we get $y_i^k = y_j/M^{k-1}$, so $(M * y_i)^k = M * y_j$ (recall that if k is even we have ensured that $S_j \geq 0$).

This is enough to show inductively that $y_i = S_i/M$ in the first five cases and $|y_i| = |S_i|/M$ in the last case (which is sufficient as we saw earlier). A similar argument about the O_i shows that $y_{\ell-n+i} = z_i/M$, so $z_i = M * y_{\ell-n+i} = S_{\ell-n+i}$; now, since $x_i = X_i = z_i$, this shows that the fixed point of g, if projected on its first n coordinates $(x_1, \ldots, x_n)$ is a fixed point of f. Also note that since we chose M so that $S_i \leq M/2$ for all $i \in [\ell]$, we know that $y_i \in [-1/2, 1/2]$ for a fixed point of g. In summary, we have shown that $F_g \subseteq F_f \times B_{n'-2n}(0, 1/2) \times F_f \cup \{\mathbf{1}\}$. In the reverse direction, it is easy to see that for x with $f(x) = x$, we can let $y_0 = 1/4$, $y_i = S_i/M$ and $z_i = x_i$ to get a fixed point $(x, y, z) \in [-1, 1]^{2n+\ell+1}$ of g; so $F_f \cup \{\mathbf{1}\} = \pi_n(F_g)$.

This concludes the proof of the lemma. □

With Lemma 3.1 it is now easy to show that BROUWER is $\exists\mathbb{R}$-hard.

Theorem 3.3. *Deciding* BROUWER *is $\exists\mathbb{R}$-complete, even for $x = 0$ and $r = 0.5$.*

The theorem remains true for any other appropriate choice of x and r. For fixed dimension, e.g. $n = 1$ or $n = 2$, BROUWER can be decided in $\mathbf{P}$ using quantifier elimination for the fixed number of quantifiers.

Proof. The problem is easily seen to lie in $\exists\mathbb{R}$. We saw earlier that deciding whether a collection of n-variate polynomials $f = (f_i)_{i\in[n]} : \mathbb{R}^n \to \mathbb{R}^n$ has a fixed point is hard for $\exists\mathbb{R}$; since these polynomials are given explicitly, it is easy to construct an SLP S computing f. By Theorem 2.3 if f has a fixed point, there has to be a fixed point at distance less than $R/2 = \lceil 2^{\ell^{cn+1}} \rceil$ from 0, where ℓ is the length of S, and c is a fixed constant. Now f maps $B_n(0, R)$ to $B_n(0, R')$, where $R' = \lceil R^{2^\ell} \rceil \leq \lceil 2^{2^{c'\ell n}} \rceil$ (each coordinate can be

at most squared in each of the at most ℓ steps of the computation; c' only depends on c, so it is a fixed constant). Let g be the continuous map that is the identity on $B_n(0, R/2)$ and bijectively maps $B_n(0, R') - B_n(0, R/2)$ to $B_n(0, R) - B_n(0, R/2)$ defined component-wise by:

$$g_i(x) = \begin{cases} x_i & \text{if } x_i \in B_1(0, R/2) \\ \operatorname{sgn}(x_i)\frac{R}{2}\left(\frac{|x_i|-(R/2)}{R'-R/2}+1\right) & \text{if } x_i \in B_1(0, R') - B_1(0, R/2) \end{cases}$$

for $i \in [n]$, where $\operatorname{sgn}(x)$ is the sign function. Then $g \circ f$ maps $B_n(0, R)$ to $B_n(0, R)$; moreover, any fixed point of f in $B_n(0, R/2)$ is still a fixed point of $g \circ f$ in $B_n(0, R/2)$ and vice versa. Finally, let h be a scaling by R, that is h is a continuous bijection between $B_n(0, R)$ and $B_n(0, 1)$. Thus $h \circ g \circ f \circ h^{-1} : B_n(0, 1) \to B_n(0, 1)$ has a fixed point in $B_n(0, 0.5)$ if and only if f has a fixed point (in $\mathbb{R}^n$).

Now, there will not, in general, be an SLP computing $h \circ g \circ f \circ h^{-1}$ since such an SLP would require division and case distinction; however, it is easy to see that there is an ESLP: this is clear for h and most of g; the only interesting question is how to perform the case distinction, but that can be done using max and min:

$$\begin{aligned} g_i(x) = \max(0, \min(x_i, R/2) + \frac{R}{2}\max(0, \frac{x_i - R/2}{R' - R/2})) \\ - \max(0, \min(-x_i, R/2) + \frac{R}{2}\max(0, \frac{-x_i - R/2}{R' - R/2})). \end{aligned}$$

Finally, ESLPs are closed under composition of functions, so we can conclude that there is an ESLP for $h \circ g \circ f \circ h^{-1}$ and thus, by Lemma 3.1 an SLP for a function $f' : B_{n'}(0, 1) \to B_{n'}(0, 1)$ so that $F_{h\circ g\circ f\circ h^{-1}} \cup \{\mathbf{1}\} = \pi_n(F_{f'})$. Moreover, the lemma allows us to conclude that $F_{f'} \subseteq F_{h\circ g\circ f\circ h^{-1}} \times B_{n'-2n}(0, 1/2) \times F_{h\circ g\circ f\circ h^{-1}} \cup \{\mathbf{1}\}$. Now f has a fixed point if and only if f' has a fixed point in $B_{n'}(0, 1/2)$. □

Etessami and Yannakakis show how to reduce the fixed-point problem to the Nash equilibrium problem (which we will not define here); the core result that makes their reduction possible is summarized in the following lemma:

Lemma 3.4 (Etessami, Yannakakis [4, Theorem 4.3])**.** *Given a function $f : B_n(0, 1) \to B_n(0, 1)$ specified by an SLP, one can construct in polynomial time a 3-player game G and compute an integer N so that*

- *if x^* is a fixed-point of f then there is a Nash equilibrium $z = (z_1, z_2, z_3)$ of Γ so that $z_1[1:n] = x^*/N$,*
- *if $z = (z_1, z_2, z_3)$ is a Nash equilibrium of Γ, then $x^* = Nz_1[1:n]$ is a fixed point of f.*

By Theorem 3.3, BROUWER is $\exists\mathbb{R}$-hard, and Lemma 3.4 shows that there is a reduction from BROUWER to the Nash equilibrium problem, so the following corollary is immediate now.

Corollary 3.5. *Deciding whether a 3-player game Γ has a Nash equilibrium in $B_n(x, r)$ for $x \in \mathbb{Q}^n$, $r \in \mathbb{Q}$ is $\exists\mathbb{R}$-complete even for $x = 0$ and $r = 0.5$.*

Remark 3.6. Datta showed the universality of 3-player totally mixed Nash equilibria [3]; algebraically this is a stronger result, since it shows that arbitrary semi-algebraic sets can be encoded as Nash equilibria; however, the reduction is not polynomial time, since some players in her game use $\Omega(d^n)$ pure strategies, where d is the highest power of any variable in the polynomial equations encoding the semi-algebraic sets, see [3, Theorem 2].

Etessami and Yannakakis also given a reduction from BROUWER (with max) to the exchange equilibrium problem (see [4, Proposition 4.4] for details); starting with our more restrictive version of BROUWER, we can conclude that the problem remains hard if the ESLP is restricted further.

Corollary 3.7. *The exchange equilibrium problem with an excess demand function given by an ESLP is $\exists\mathbb{R}$-complete. Indeed, this remains true if the ESLP is restricted to division (that is, no roots,* max *or* min *operations are allowed).*

Etessami and Yannakakis [4] studied in depth the search versions of fixed-point problems and Nash equilibriums: Suppose we are given a function $f : B_n(0,1) \to B_n(0,1)$ via an ESLP. How hard is it to find some (any) fixed point of f? This, of course, is a problem over real numbers, but one can turn it into a discrete problem as follows. For any input $r \in \mathbb{Q}^n$ we are allowed to ask questions of the type $x\Theta r$, where Θ is one of $\{\leq, \geq, <, >\}$. If $x\Theta r$ for all fixed points x of f the answer has to be "yes", if $x\Theta r$ is false for all fixed points of f the answer has to be "no"; otherwise, the answer can be either "yes" or "no". Etessami and Yannakakis call this the *decision problem* (in their terminology, BROUWER is an existence problem, not a decision problem). The class of all such fixed-point decision problems is called $\mathbf{FIXP_d}$. $\mathbf{FIXP_d}$ is rather robust, for example it is not affected by

changes of domain. Etessami and Yannakakis [4, Theorem 4.7] also show that $\mathbf{FIXP_d}$ remains the same if ESLPs are restricted to $\{+, *, \max\}$.[6]

Moreover, the class has natural complete problems, including, among several others, the decision versions of BROUWER and the Nash equilibrium problem for 3 players. Clearly, $\mathbf{FIXP_d} \subseteq \exists_{=}\mathbb{R}$, and Etessami and Yannakakis show that PosSLP reduces to any $\mathbf{FIXP_d}$-complete problem; currently this is the best known lower bound on $\mathbf{FIXP_d}$. Etessami and Yannakakis point out that it is unlikely that any $\mathbf{FIXP_d}$-problem is **NP**-hard, since in that case it is also **coNP**-hard, which would imply $\mathbf{coNP} \subseteq \exists_{=}\mathbb{R}$, which is not impossible, but seems counterintuitive. Similarly, $\mathbf{FIXP_d} = \exists_{=}\mathbb{R}$ would imply that $\exists_{=}\mathbb{R}$ is closed under complement, which again appears unlikely.

References

[1] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in real algebraic geometry*, volume 10 of *Algorithms and Computation in Mathematics*. Springer-Verlag, Berlin, second edition, 2006.

[2] John Canny. Some algebraic and geometric computations in pspace. In *STOC '88: Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 460–469, New York, NY, USA, 1988. ACM.

[3] Ruchira S. Datta. Universality of Nash equilibria. *Math. Oper. Res.*, 28(3):424–432, 2003.

[4] Kousha Etessami and Mihalis Yannakakis. On the complexity of Nash equilibria and other fixed points. *SIAM J. Comput.*, 39(6):2531–2597, 2010.

[5] D. Yu. Grigor'ev and N. N. Vorobjov. Solving systems of polynomial inequalities in subexponential time. *J. Symb. Comput.*, 5(1-2):37–64, 1988.

[6] Hoon Hong. Comparison of several decision algorithms for the existential theory of the reals. Technical Report 91-41, RISC-Linz, Johannes Kepler University, Linz, Austria, 1991.

[7] Jan Kratochvíl and Jiří Matoušek. Intersection graphs of segments. *J. Combin. Theory Ser. B*, 62(2):289–315, 1994.

[6] The proof uses Nash equilibria, compare Lemma 3.1 which gets rid of max as well, but adds a fixed point.

[8] Daniel Kroening and Ofer Strichman. *Decision procedures*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2008. An algorithmic point of view, With a foreword by Randal E. Bryant.

[9] Jan Kynčl. Simple realizability of complete abstract topological graphs in p. *Discrete & Computational Geometry*, 45:383–399, 2011.

[10] C.J.H. McDiarmid and T. Müller. The number of bits needed to represent a unit disk graph, 2010.

[11] Bhubaneswar Mishra. Computational real algebraic geometry. In *Handbook of discrete and computational geometry*, CRC Press Ser. Discrete Math. Appl., pages 537–556. CRC, Boca Raton, FL, 1997.

[12] N. E. Mnëv. The universality theorems on the classification problem of configuration varieties and convex polytopes varieties. In *Topology and geometry—Rohlin Seminar*, volume 1346 of *Lecture Notes in Math.*, pages 527–543. Springer, Berlin, 1988.

[13] Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley Publishing Company, Reading, MA, 1994.

[14] Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. System Sci.*, 48(3):498–532, 1994. 31st Annual Symposium on Foundations of Computer Science (FOCS) (St. Louis, MO, 1990).

[15] Marcus Schaefer. The real logic of drawing graphs. Unpublished Manuscript.

[16] Marcus Schaefer. Complexity of some geometric and topological problems. In David Eppstein and Emden R. Gansner, editors, *Graph Drawing*, volume 5849 of *Lecture Notes in Computer Science*, pages 334–344. Springer, 2009.

[17] Marcus Schaefer. Realizability of graphs and linkages. Invited to volume on talks from *Conference on Geometric Graph Theory*, Lausanne, 2010, 2011.

[18] Michael Sipser. *Introduction to the Theory of Computation*. Course Technology, 2nd edition, 2005.

[19] G. S. Tseitin. On the complexity of derivation in propositional logic. In Graham Wrightson Jörg Siekmann, editor, *Automation of Reasoning: Classical Papers on Computational Logic 1967–1970*, volume 2, pages 466–483. Springer, 2009.