Robustness of the Discrete Real Polynomial Hierarchy

Bachelor's Thesis of

Tim Janik Junginger

At the Department of Informatics
Institute of Theoretical Informatics (ITI)

Reviewer:	Torsten Ueckerdt
Second reviewer:	Thomas Bläsius
Advisors:	Paul Jungeblut

08.05.2023 – 08.09.2023

Karlsruher Institut für Technologie
Fakultät für Informatik
Postfach 6980
76128 Karlsruhe

I hereby declare that this document has been composed by myself and describes my own work, unless otherwise acknowledged in the text. I also declare that I have read and observed the *Satzung zur Sicherung guter wissenschaftlicher Praxis am Karlsruher Institut für Technologie.*

Karlsruhe, 08.09.2023

. . . .

(Tim Janik Junginger)

Abstract

We look at the (discrete) real polynomial hierarchy, an analogue to the polynomial hierarchy, but over the theory of reals. In addition to the standard two quantifiers $\exists$ and $\forall$, we expand the hierarchy with three exotic quantifiers: $\exists^*$, $\forall^*$ and H. There are two definitions of this expansion, respectively defined over the standard discrete Turing model and the so called Blum-Shub-Smale model where exact calculations with real numbers are possible. For a restriction on the latter (the constant-free Boolean part) these definitions coincide in many cases. We provide evidence and an intermediary result for their complete equivalence.

We conjecture that the polynomial hierarchy is *robust* under these exotic quantifiers, meaning that no new complexity classes arise form their introduction. We prove robustness for the first and second level of the hierarchy and provide first results for all higher levels.

Zusammenfassung

Wir betrachten die (diskrete) reelle polynomielle Hierarchie, welche ein Analogon zur polynomiellen Hierarchie, aber über den reellen Zahlen, ist. Zusätzlich zu den normalen Quantoren $\exists$ und $\forall$, erweitern wie die Hierarchie um drei exotische Quantoren: $\exists^*$, $\forall^*$ und H. Für diese Erweiterung gibt es zwei verschiedene Definitionen, die einerseits das normale Turing-Berechnungsmodell und andererseits das sogenannte Blum-Shub-Smale Modell nutzen. Unter einer Einschränkung (dem sogenannten constant-free Boolean Part) des letzteren, stimmen diese teilweise überein. Wir geben Gründe und ein Zwischenergebnis dafür an, dass diese tatsächlich vollständig übereinstimmen.

Außerdem stellen wir die Vermutung auf, dass die reelle polynomielle Hierarchie *robust* unter den exotischen Quantoren ist, was bedeutet, dass keine neuen Komplexitätsklassen durch deren Einführung entstehen. Wir beweisen die Robustheit für die erste und zweite Ebene der Hierarchie und stellen Ergebnisse für alle höheren Ebenen vor.

Contents

1 Introduction

1.1 Motivation

The complexity classes P and NP have already been studied extensively. One tool in this study have been reductions. We focus only on polynomial time many-to-one reductions in this thesis. We say that a problem Π_1 can be reduced, in polynomial time, to another problem Π_2 if we can transform any instance of Π_1, in polynomial time, to an equivalent instance of Π_2. This implies (among other things) that if Π_2 can be solved in polynomial time, then so too can Π_1.

With these reductions one can define the notions of *hardness* and *completeness*. A problem Π is considered hard for a complexity class C (C-hard) if all problems $\Pi' \in$ C can be reduced to it. Π is then C-complete if it is also contained in C itself.

We can even use these concepts to define the complexity class NP in a non-standard way: Consider the problem SAT as deciding whether a Boolean formula $\phi(A)$ over a set of Boolean variables A is satisfiable, i.e. whether the sentence $\exists a : \phi(a)$ is true. We can now define NP as the complexity class for which SAT is complete. This is equivalent to the standard definition of NP because of the well-known Cook-Levin theorem which shows that SAT is indeed NP-complete.

We can, however, extend this approach to obtain further complexity classes. Consider, for example, the problem Tautology as deciding whether a Boolean formula $\phi(A)$ is true for all variable assignments, i.e. whether the sentence $\forall a : \phi(a)$ is true. We can (again in a non-standard way) define coNP as the complexity class for which Tautology is complete. We can even introduce additional quantifiers to the left of a formula to obtain even more complexity classes. In fact, by doing so, we obtain the *polynomial hierarchy* (which is often instead defined over oracle machines). This hierarchy has also been studied extensively.

And again defining a hierarchy over complete problems can be applied to different problems as well. For example, by quantifying a formula over the Peano arithmetic, one obtains the *arithmetical hierarchy* and by quantifying a formula over the second-order arithmetic, one obtains the *analytical hierarchy*. While these hierarchies have at least found some recognition (all have wikipedia articles [Wikd] [Wikb] [Wika]) another hierarchy has not found that level of recognition:

1.1.1 Definitions

(Discrete) Real Polynomial Hierarchy The *(discrete)*[1] *real polynomial hierarchy* is defined by quantifying a formula in the first-order theory of the reals (see Section 2.1 for an introduction of this theory).

[1] We generally mean the *discrete* real polynomial hierarchy and drop *discrete* only out of convenience. There is, however, reason to distinguish two different variants of this hierarchy. But because the same concept underlies both hierarchies, we do not distinguish between them here and delay such discussions until Section 4.1.

$$\vdots$$

$$\boxed{\exists\forall\exists\mathbb{R} \quad \leftarrow \text{co} \rightarrow \quad \forall\exists\forall\mathbb{R}}$$

$$\cup\!\!|$$

$$\boxed{\exists\forall\mathbb{R} \quad \leftarrow \text{co} \rightarrow \quad \forall\exists\mathbb{R}}$$

$$\cup\!\!|$$

$$\boxed{\exists\mathbb{R} \quad \leftarrow \text{co} \rightarrow \quad \forall\mathbb{R}}$$

$$\cup\!\!|$$

$$\boxed{\emptyset}$$

Figure 1.1: A first sketch of the real polynomial hierarchy

Definition 1.1: *We define the problem* STANDARD*($Q_1 \ldots Q_\omega$) as deciding whether for a given quantifier-free formula ϕ the following sentence is true:*

$$Q_1 x_1 \ldots Q_\omega x_\omega : \phi(x_1, x_2, \ldots, x_\omega)$$

with $Q_i \in \{\exists, \forall\}$.

This definition even extends to the case with no quantifiers, i.e. STANDARD($\emptyset$) in which case the problem is simply deciding whether a given quantifier-free sentence ϕ is true.

With these problems we can define the discrete real polynomial hierarchy which is the union of complexity classes of the form:

Definition 1.2: *The complexity class* $Q_1 \ldots Q_\omega\mathbb{R}$ *is the class for which* *STANDARD*($Q_1, \ldots, Q_\omega$) *is complete and* $Q_i \in \{\exists, \forall\}$.

The $\mathbb{R}$ here signifies that this hierarchy is over the first-order theory of the *reals*.

One can quite easily see that there are levels to this hierarchy. So is for example the complexity class $\exists\forall\mathbb{R}$ contained in $\exists\forall\exists\mathbb{R}$ but also in $\forall\exists\forall\mathbb{R}$. It should also be intuitive that $\forall\exists\forall\mathbb{R}$ is the complement of $\exists\forall\exists\mathbb{R}$. Taking these together we can form a first visual understanding of this hierarchy shown in Figure 1.1.

Additionally, the real polynomial hierarchy seems to be the only such hierarchy for which new *exotic quantifiers* have been introduced and studied. We look at three such quantifiers: $\exists^*$, $\forall^*$, H. These can be understood roughly as "exists an open ball", "for almost all" and "for all small enough", but we define them formally in Section 2.4.

We can extend our definition of the real polynomial hierarchy by these quantifiers to obtain standard problems and complexity classes for them:

Definition 1.3: *We define the problem* STANDARD($\mathrm{Q}_1 \ldots \mathrm{Q}_\omega$) *as deciding whether for a given quantifier-free formula ϕ the following sentence is true:*

$$\mathrm{Q}_1 x_1 \ldots \mathrm{Q}_\omega x_\omega : \phi(x_1, x_2, \ldots, x_\omega)$$

with $\mathrm{Q}_i \in \{\exists, \forall, \exists^*, \forall^*, \mathrm{H}\}$.

Definition 1.4: *The complexity class* $\mathrm{Q}_1 \ldots \mathrm{Q}_\omega \mathbb{R}$ *is the class for which* STANDARD($\mathrm{Q}_1, \ldots, \mathrm{Q}_\omega$) *is complete and* $\mathrm{Q}_i \in \{\exists, \forall, \exists^*, \forall^*, \mathrm{H}\}$.

Now the question arises where these new "exotic" complexity classes, such as $\exists^*\forall\mathbb{R}$ or $\mathrm{H}\forall^*\mathrm{H}\mathbb{R}$, lie in our image of the real polynomial hierarchy.

1.1.2 Goal

This thesis aims to collect knowledge about the real polynomial hierarchy and show its *robustness* (at least up to the second level). Here, robustness is the computational equivalence of the exotic quantifiers to their non-exotic counterparts *within* this hierarchy. This makes the "exotic" complexity classes equivalent to their non-exotic counterparts allowing us to easily introduce them into our image of the hierarchy (from Figure 1.1 to Figure 4.2).

1.2 Related Work

Sontag shows that restricting the real polynomial hierarchy to only additions reduces it to the normal polynomial hierarchy [Son85]. It seems therefore highly unlikely that it is possible to eliminate multiplication in the real polynomial hierarchy.

Later Koiran provides an efficient elimination algorithm for the $\exists^*$ quantifier [Koi99].

Bürgisser and Cucker provide, to my best knowledge, the first extensive account of the real polynomial hierarchy. They also introduce the three exotic quantifiers into the hierarchy and provide first results on how they relate to their non-exotic counterparts [BC09]. We follow their definitions in these regards.

Matoušek provides a comprehensive introduction to the problem ETR and the complexity class $\exists\mathbb{R}$. This article is more of the form of a lecture script and therefore suited for a general introduction [Mat14] (cf. Chapter 3).

While we use results by Schaefer and Štefankovič, we do not use their most recent result found in a pre-print, where they show that the real polynomial hierarchy is also robust under only allowing *strict* inequalities (and no negations) [SŠ17] [SŠ22].

Dobbins, Kleist, Miltzow and Rzążewski prove $\forall\exists\mathbb{R}$-completeness and $\exists\mathbb{R}$-completeness for two variants of AREA UNIVERSALITY respectively. Additionally they provide some tools for further hardness proofs for these complexity classes [DKMR21].

Jungeblut, Kleist and Miltzow show the computation of the Hausdorf distance to be $\forall\exists_<\mathbb{R}$-complete (and with the result mentioned above $\forall\exists\mathbb{R}$-complete) [JKM22].

Erickson, van der Hoog and Miltzow describe a RAM machine model for the complexity class $\exists\mathbb{R}$, which could potentially be generalised to the whole hierarchy [EHM21]. While such a model is immensely helpful in the general study of the real polynomial hierarchy, it seems unlikely that it helps to prove robustness under exotic quantifiers and is therefore omitted.

1.3 Outline

In Chapter 2 we provide the theoretical basis needed to formally define and reason about the real polynomial hierarchy. We also set up some conventions about writing and standard forms, so even the well-versed reader might want to skim this chapter.

The following Chapter 3 introduces the more well-known problem ETR[2], which turns out to be equivalent to one of the standard problems used to define the hierarchy. We learn some useful tricks for working with the real polynomial hierarchy and show some first results.

Chapter 4 generalizes the results of Chapter 3 for the whole real polynomial hierarchy and discusses incongruities in its definition. This is followed by a collection of proofs which help to show relationships between the complexity classes in the hierarchy. The final results are split into two sections which deal with the first and second level of the real polynomial hierarchy respectively.

At the end Chapter 5 summarizes our results and discusses open questions and potential future work in this area. It also uses the results of this thesis to show that PIT (polynomial identity testing) is contained in $\exists\mathbb{R} \cap \forall\mathbb{R}$.

[2]Which already has a wikipedia article in contrast to the general hierarchy [Wikc]

2 Preliminaries

2.1 First-Order Theory of the Reals

The first-order theory of the reals is a first-order logic over the domain of real numbers $\mathbb{R}$. It includes the constants 0 and 1, the binary functions $+$, $-$, $\cdot$ and the binary predicates $\leq$, $<$, $>$, $\geq$, $\neq$ and $=$. Additionally, like in the first-order logic itself, it has symbols for Boolean connectives ($\wedge$, $\vee$, $\neg$), parenthesis, quantifiers and variables.

Semantically all variables range over the real numbers (the domain) and all functions and predicates are interpreted as the usual operations on real numbers.

A formula in the first-order theory of the reals is a formula in first-order logic with the given constants, functions and predicates in arbitrary nesting. While first-order logic only contains the two quantifiers $\exists$ and $\forall$, we introduce further exotic quantifiers to the first-order theory of the reals in Section 2.4.

A formula is called a *sentence* if it does not contain any free variables, meaning every variable is bound by a quantifier.

Definition 2.1: *A* quantifier-free *formula ϕ in the first-order theory of the reals has the general form:*

$$\phi = \phi(X_1, ..., X_n) = F(A_1, ..., A_m)$$

where F is a Boolean formula and each A_i is an atom *of the form $p_i(X_1, ..., X_n) \gtreqless 0$ with p_i a polynomial in $X_1, ..., X_n$ with integer (or rational) coefficients.*

Prenex Form A formula is in *prenex form* if it is of the form $(Q_1X_1)(Q_2X_2)...(Q_nX_n)\phi$, where $Q_1, ..., Q_n$ are quantifiers and ϕ is a quantifier-free formula. For any given formula, such a prenex form can be easily obtained by uniquely renaming all variables to avoid naming conflicts and then simply pushing the quantifiers outside (to the left). Care has to be taken with $\neg$ as $\neg\exists X\phi$ becomes $\forall X\neg\phi$ and vice versa. This changes the formula length by at most a constant factor.

Formula Length The length of a formula is *roughly* the number of bits needed to encode it. Functions, predicates, quantifiers, logical connectives and the constants 0 and 1 can all be encoded with a constant number of bits. For practicality's sake we also write more complex phrases in our formulas, but for the calculation of the formula length they have to be expanded: An integer constant k can be expressed by $O(\log k)$ symbols through binary expansion, e.g. $13 = 2^3 + 2^2 + 1 = (2+1) \cdot 2^2 + 1 = ((1+1)+1) \cdot (1+1) \cdot (1+1) + 1$. A power X^k however has to be expanded into the k-fold multiplication $X \cdot ... \cdot X$. This becomes especially important when considering nested powers like $(...(X^2)^2)^2$ with n nestings which equates to X^{2^n} requiring $O(2^n)$ symbols. While a formula containing n variables would require $O(\log n)$ bits for each variable, this only changes the formula length by this factor $O(\log n)$ which is small enough to be safely ignored in most circumstances.

Quantifier Elimination For every formula in the first-order theory of the reals there exists an equivalent quantifier-free formula. This is a result by Tarski whose proof also implies the decidability of the first-order theory of the reals.

Theorem 2.2 ([Tar51]): *There is an algorithm accepting as input a formula ψ of the first-order theory of the reals, which may contain quantifiers; in general it also contains free variables $Y = (Y_1, ..., Y_n)$, which we write as $\psi = \psi(Y)$. The algorithm outputs a quantifier-free formula $\phi = \phi(Y)$ that is equivalent to ψ; that is, for every choice of $y \in \mathbb{R}^n$ we have $\psi(y) = \phi(y)$.*

Let us now consider such a formula ψ containing m polynomials each of degree at most Δ and with the number of bits used to represent each coefficient bounded by τ. Further let ω be the number of alternating quantifiers (i.e. $\exists\forall$ or $\forall\exists$) with n_i variables in the i-th block and n_0 the number of free variables. Define $N := \prod_{i=0}^{\omega}(n_i + 1)$ and let C be a suitable constant.

Theorem 2.3 ([BPR96] [BPR03] [BPR16]): *There exists a quantifier elimination algorithm which requires $m^N \cdot (\Delta + 1)^{C^\omega \cdot N}$ arithmetic operations on at most $((\Delta + 1)^{C^\omega \cdot N} \cdot \tau)$-bit integers.*

A practical result of Theorem 2.3 is that quantifier elimination requires times doubly exponential in the number of quantifier alternations ω. For a constant number of quantifiers, this reduces to singly exponential in a polynomial of the number of variables.

This result is optimal in the sense that there exists an example of a formula in the first-order theory of the reals which grows to double exponential length through quantifier elimination [Mat14][DH88].

Another quantifier elimination result by Renegar also provides us with information about the resulting formula:

Theorem 2.4 ([Ren92]): *Let*

$$\mathrm{Q}_1 x_1 \in \mathbb{R}^{n_1} \dots \mathrm{Q}_\omega x_\omega \in \mathbb{R}^{n_\omega} : \phi(y, x_1, ..., x_\omega)$$

be a prenex formula in the first order theory of the reals with free variables $y \in \mathbb{R}^s$. Then $\phi(y, x_1, ..x_\omega)$ is a Boolean formula with atomic predicates of the form $p_i(y, x_1, ..., x_\omega) \gtreqless 0$ for $i \in \{1, ..., m\}$. Let $d \geq 2$ be the maximum degree of all these polynomials.

There exists a quantifier-elimination algorithm that produces an equivalent formula ψ of the form

$$\psi = \bigvee_{i=1}^{I} \bigwedge_{j=1}^{J_i} (h_{ij}(y) \gtreqless 0)$$

where

$$I \leq (md)^{\left(2^{O(\omega)} \cdot s \cdot \prod_{k=1}^{\omega} n_k\right)}$$
$$J_i \leq (md)^{\left(2^{O(\omega)} \cdot \prod_{k=1}^{\omega} n_k\right)}$$

and the degree of the non-zero polynomials h_{ij} are bounded by $(md)^{\left(2^{O(\omega)} \cdot \prod_{k=1}^{\omega} n_k\right)}$. If the coefficients of the p_i are integers of bit length at most τ then the coefficients of the h_{ij} are integers with bit length at most $(\tau + s)(md)^{\left(2^{O(\omega)} \cdot \prod_{k=1}^{\omega} n_k\right)}$.

2.2 Algebraic Circuits

Another option to encode calculations and conditions on real numbers are algebraic circuits. They are used in one of the two definitions of the real polynomial hierarchy (cf. Section 4.1). Section 4.1.1 discusses the differences between algebraic circuits and formulas in the hopes of unifying the definitions of the hierarchies. Overall algebraic circuits play a small role in this thesis and are mostly found in the mentioned sections. We define them according to Bürgisser and Cucker [BC09].

Definition 2.5: *An algebraic circuit C (over $\mathbb{R}$) is an acyclic directed graph where each node (or in this context also gate) has indegree 0,1 or 2 and there are five types of nodes:*

- input nodes *have indegree 0 and represent variables.*
- constant nodes *have indegree 0 and are labeled with a single element of $\mathbb{R}$ which they represent.*
- arithmetic nodes *have indegree 2 and are labeled with an arithmetic operation $\circ \in \{+, -, \cdot, \div\}$ which they compute.*
- sign nodes *have indegree 1 and compute*

$$sgn(x) = \begin{cases} 1 & if\, x \geq 0 \\ 0 & if\, x < 0 \end{cases}$$

- output nodes *have indegree 1 and outdegree 0 and represent the output of the circuit.*

Note that the outdegree of a node is not restricted (except for output nodes), but the total outdegree is (each outgoing edge has to be an incoming edge of another node). An algebraic circuit with n input nodes and m output nodes is associated with a function $f_C : \mathbb{R}^n \to \mathbb{R}^m$ which may not be total as division by 0 can occur. To prevent confusion with other types of nodes we also call the nodes of an algebraic circuit *gates*.

Definition 2.6: *For a given algebraic circuit C with nodes V and edges E we define the sub-circuit $C_g := (V', E')$ with*

$$\begin{aligned} V' &:= \{v \in V \mid \text{there is a directed path from } v \text{ to } g\} &\cup\quad& \{o_{new}\} \\ E' &:= \{(u, v) \in E \mid u, v \in V'\} &\cup\quad& \{(g, o_{new})\} \end{aligned}$$

where $o_{new} \notin V$ is a new output node.

We can obtain a new function $C_g : \mathbb{R}^n \to \mathbb{R}$ which computes the output value of the node g. It follows that for a circuit C with output nodes $o_1, ..., o_m$

$$C(x) = \begin{pmatrix} C_{o_1}(x) \\ \dots \\ C_{o_m}(x) \end{pmatrix}.$$

Because this thesis focuses only on decision problems, a certain subclass of algebraic circuits is of particular interest: the algebraic *decision* circuits.

Definition 2.7: *A decision circuit C is an algebraic circuit with exactly one output node whose only parent node is a sign node. Its associated function is $f_C : \mathbb{R}^n \to \{0, 1\}$.*

The next two theorems show how we can transform an algebraic decision circuit into a formula and vice versa. For this purpose we assume the decision circuit to be *constant-free*, meaning that only the constants 0 and 1 are used. The transformations are in polynomial time, but introduce additional variables into the formula to represent the nodes of the circuit:

Theorem 2.8: *Let $C(y)$ be a constant-free algebraic decision circuit over free variables $y \in \mathbb{R}^m$. Then, we can construct in polynomial time a formula $\phi_C(y, x)$ in free variables $y \in \mathbb{R}^m, x \in \mathbb{R}^{O(|C|)}$ such that*

$$C(y) = 1 \quad \Leftrightarrow \quad \exists x \in \mathbb{R}^{O(|C|)} : \phi_C(y, x)$$

for all $y \in \mathbb{R}^m$ and where $|C|$ is the number of nodes in C.

Proof. For every node (gate) g_i of C, introduce a new variable x_i and initialise $\phi_C = \emptyset$ a conjunction of polynomial equations.

1. For every input node g_i, add the equation $x_i - y_i = 0$.
2. For every constant node g_i, with value c add the equation $x_i - c = 0$.
3. For the single output node $g_{|C|}$ with parent node $g_{|C|-1}$, add the equation $x_{|C|-1} - 1 = 0$.
4. For every arithmetic node g_i with parent nodes g_p and g_q and the operation $\circ \in \{+, -, \cdot\}$, add the equation $(x_p \circ x_q) - x_i = 0$.
5. For every arithmetic node g_i with parent nodes g_p and g_q and the operation $\div$, add the variable t_i and the equations $(x_i \cdot x_q) - x_p = 0$ and $(x_q \cdot t_i) - 1 = 0$.
6. For every sign node g_i with parent node g_p, add the variables t_i, s_i,t'_i,s'_i,r_i, the equations $x_i \cdot (x_i - 1) = 0$, $t_i - s_i^2 = 0$, $t'_i - s_i'^2 = 0$, $(t'_i \cdot r_i) - 1 = 0$ and $x_p - (x_i \cdot t_i) + ((1 - x_i) \cdot t'_i) = 0$.

We now prove, in topological ordering, that for given $y \in \mathbb{R}^m$ the variable x_i can only assume the value of the output of the node g_i.

1. $x_i - y_i = 0 \Leftrightarrow x_i = y_i$
2. $x_i - c = 0 \Leftrightarrow x_i = c$
3. $x_{|C|} - 1 = 0 \Leftrightarrow x_{|C|} = 1$ ensures that the output of the second to last node is indeed 1.
4. $(x_p \circ x_q) - x_i = 0 \Leftrightarrow (x_p \circ x_q) = x_i$
5. $(x_i \cdot x_q) - x_p = 0 \Leftrightarrow x_i = (x_p \div x_q)$ and $(x_q \cdot t_i) - 1 = 0$ if and only if $x_q \neq 0$.
6. - $x_i \cdot (x_i - 1) = 0$ ensures that $x_i \in \{0, 1\}$.
   - $t_i - s_i^2 = 0$ and $t'_i - s_i'^2 = 0$ ensure that $t_i, t'_i \geq 0$.
   - $(t'_i \cdot r_i) - 1 = 0$ ensures that $t'_i \neq 0$.
   - $x_p - (x_i \cdot t_i) + ((1 - x_i) \cdot t'_i) = 0$ ensures that either $x_p = t_i \wedge x_i = 1$ or $x_p = -t'_i \wedge x_i = 0$.

   Because $t'_i > 0$ this implies $x_p \geq 0 \Leftrightarrow x_i = 1$.

For every node, we introduce at most six variables and a constant number of constant-size equations. Therefore this construction is polynomial. ∎

Theorem 2.9: *Let $\phi(x)$ be a quantifier-free formula over free variables $x \in \mathbb{R}^n$. Then we can construct, in polynomial time, an algebraic decision circuit C_ϕ with only 0 and 1 as constants, such that for all $x \in \mathbb{R}^n$*

$$\phi(x) \quad \Leftrightarrow \quad C_\phi(x) = 1.$$

Proof. First replace, in ϕ, all occurrences of $\neq$ with an equivalent construction of $<, >$ and then replace all occurrences of $>$ with an equivalent construction of $<$. Lastly, replace all occurrences of $=$, $\leq$ and $<$ with an equivalent construction of $\geq$, possibly using negation. This can be done in polynomial time.

We build the circuit C_ϕ in a similar way as we built the formula in Theorem 2.8. Instead of traversing the circuit we traverse the concrete syntax tree of ϕ in topological order. For every node t_i of this tree that does not represent a variable, we introduce one gate g_i in the algebraic circuit. We introduce further for every x_i a single input gate holding this variable.

Additionally we create a “global” constant node holding the value 1, which we call c_1.

We also construct a helper gadget which holds the constant 2. It is made from two constant gates holding the value 1 and one arithmetic gate computing their sum. We call this arithmetic gate c_2.

1. For every node t_i representing the (Boolean or real) constant 0 or 1, the gate g_i is a constant gate with the constant 0 or 1 respectively.
2. For every node t_i representing a variable x_j, the gate g_i is *the* input gate holding variable x_j.
3. For every node t_i representing an arithmetic operation $\circ$ with parent nodes t_p and t_q, the gate g_i is an arithmetic gate with operation $\circ$ and parent gates g_p and g_q.
4. For every node t_i representing a predicate ≥ 0 with parent node t_p, the gate g_i is a sign gate with parent gate g_p.
5. For every node t_i representing the Boolean connective $\wedge$ with parent nodes t_p and t_q, the gate g_i is an arithmetic gate with the operation $\cdot$ and parent gates g_p and g_q.
6. For every node t_i representing the Boolean connective $\vee$ with parent nodes t_p and t_q, we utilise the equivalence $(a \vee b) \Leftrightarrow (2a + 2b - 1 \geq 0)$ for $a, b \in \{0, 1\}$. The right side can be computed by four arithmetic nodes and the sign node g_i using the output from c_1 and c_2 for the constants 1 and 2 respectively.
7. For every node t_i representing the Boolean connective $\neg$ with parent node t_p, the gate g_i is an arithmetic gate which subtracts the output of g_p from the output of c_1, i.e. the constant 1.

We show the correctness of this construction by traversing the concrete syntax tree in topological order for a given $x \in \mathbb{R}^n$. Note that the resulting circuit is, except for the gadgets and input gates, also a tree. We then show that the value represented by each node t_i in the tree is equivalent to the value represented by the gate g_i.

1. The values clearly correspond for constant nodes. In the circuit however we use 0 and 1 both as Boolean constant and as real constants.
2. Instead of having an input gate for every occurrence of a variable, we only have one gate for every variable.

3 For arithmetic operations, the gates correspond to the nodes in the syntax tree by definition.

4 The equivalence between the predicate ≥ 0 and a sign node is given by definition.

5 The concrete syntax tree already ensures that the inputs to a Boolean connective are Boolean values. Therefore we know that the outputs of g_p and g_q are restricted to $0, 1$. Therefore the multiplication corresponds to the logical conjunction and also only produces value 0 or 1.

6 The construction for $\vee$ first multiplies the input values of 0 or 1 by 2 transforming them to 0 or 2. Adding them together yields a result in $\{0, 2, 4\}$ and subtracting 1 from it yields $\{-1, 1, 3\}$. Clearly the only possibility to obtain -1 is if both input values were 0. Therefore the construction is equivalent to $\vee$.

7 Clearly for $x \in \{0, 1\} : (1 - x) \Leftrightarrow \neg x$ (excusing the abuse of notation).

For each node in the syntax tree we add at most five gates in the circuit. Therefore the construction is polynomial. ∎

2.3 Semialgebraic Sets

Definition 2.10 ([Mat14]): *A set $S \subseteq \mathbb{R}^n$ is called semialgebraic if and only if it can be described by a quantifier-free formula ϕ: $S = \{x \in \mathbb{R} : \phi(x)\}$. It is a set-theoretic combination of finitely many zero sets and non-negativity sets of polynomials.*

Corollary 2.11: *Any set describable by a (potentially quantified) algebraic decision circuit C or a general formula in the first-order theory of the reals ϕ is semialgebraic.*

Proof. Theorem 2.8 allows us to transform any algebraic decision circuit, with free variables, into an equivalent existential formula. Theorem 2.2 provides us with a quantifier elimination algorithm to obtain a quantifier-free formula. ∎

Definition 2.12 ([BC09]): *A semialgebraic set S is* basic *if and only if*

$$S = \{x \in \mathbb{R}^n : \bigwedge_{p \in P} p(x) \geq 0 \wedge \bigwedge_{q \in Q} q(x) > 0\}$$

for some finite sets of polynomials $P, Q \subset \mathbb{Z}[X]$.

Every semialgebraic set is the union of finitely many basic semialgebraic sets. The complement of a semialgebraic set $S = \{x \in \mathbb{R}^n : \phi(x)\}$ is also semialgebraic:

$$S^C = \{x \in \mathbb{R}^n : \neg\phi(x)\}$$

Theorem 2.13 ([Mat14, Theorem 3.7] [BPR16, Proposition 6.34 and Theorem 7.25]): *The number of connected components of any semialgebraic sets is finite.*

2.4 Exotic Quantifiers

Convention From now on throughout this paper we drop the domain of quantified values if it is given from the context or it can be any $\mathbb{R}^n$ for $n \in \mathbb{N}$.

In this section we present three exotic quantifiers as introduced by Bürgisser and Cucker. The original idea of these quantifiers was to simplify completeness proofs for classes in the hierarchy. We can, for example, phrase the problem EDense$_\mathbb{R}$, whether a given semialgebraic set S is (Euclidean) dense, simply as $\forall^* x : x \in S$. From this it is clear that EDense$_\mathbb{R}$ is $\forall^*\mathbb{R}$-complete [BC09, Proposition 5.2]. If we can show robustness of the hierarchy under these quantifiers, then it follows that EDense$_\mathbb{R}$ is $\forall\mathbb{R}$-complete. For any formula ϕ in the first-order theory of the reals, the quantifiers H, $\exists^*$ and $\forall^*$ are defined as follows:

Definition 2.14 ([BC09]):

$$
\begin{aligned}
\mathsf{H}\epsilon : \phi(\epsilon) &= \exists r > 0\ \forall \epsilon \in (0, r) : \phi(\epsilon) \\
\forall^* x : \phi(x) &= \forall x_0\ \forall \epsilon > 0\ \exists x : (\|x - x_0\| < \epsilon \wedge \phi(x)) \\
\exists^* x : \phi(x) &= \exists x_0\ \exists r > 0\ \forall x : (\|x - x_0\| < r \Rightarrow \phi(x))
\end{aligned}
$$

Note that the quantifier H quantifies only over a single real number while $\exists^*$ and $\forall^*$ quantify an arbitrary vector in $\mathbb{R}^n$. Intuitively one can understand the H quantifier as "for any sufficiently small ϵ". The $\forall^*$ quantifier can be understood as "for all elements in a dense subset of the domain" and the $\exists^*$ quantifier as "exists an open ball".

It should be intuitive that for all formulas ϕ:

$$\neg\exists^* x : \phi(x) \Leftrightarrow \forall^* x : \neg\phi(x)$$

We can also find a similar result for the quantifier H which is included in Lemma 2.15.

Lemma 2.15 ([BC09, Proposition 6.1]): *For all formulas ϕ, it holds that*

$$
\begin{aligned}
\neg\mathsf{H}\epsilon : \phi(\epsilon) &\Leftrightarrow \mathsf{H}\epsilon : \neg\phi(\epsilon) \\
\mathsf{H}\epsilon : \phi(\epsilon) &\Leftrightarrow \mathsf{E}r > 0\ \mathsf{A}\epsilon \in (0, r) : \phi(\epsilon)
\end{aligned}
$$

for $\mathsf{E} \in \{\exists, \exists^*\}$ *and* $\mathsf{A} \in \{\forall, \forall^*\}$.

Proof. Let $S := \{\epsilon \mid \phi(\epsilon)\}$ be the semialgebraic set defined by ϕ. W.l.o.g. we can assume the domain to be $(0, \infty)$ (instead of $\mathbb{R}$) as $\epsilon > 0$ is ensured in all cases. We define $S^C = (0, \infty) \setminus S = \{\epsilon \mid \neg\phi(\epsilon)\}$ as the complement of S over $(0, \infty)$.

The definition of H implies that

$$\neg\mathsf{H}\epsilon : \phi(\epsilon) \equiv \forall r > 0\ \exists \epsilon \in (0, r) : \neg\phi(\epsilon)$$

which holds if and only if there exists a sequence of points $(x_n)_{n\in\mathbb{N}} \subseteq S^C \subset (0, \infty)$ which converges to 0. From Theorem 2.13 we know that both S and S^C have only finitely many connected components. Because no value x_n in the sequence can be 0, almost all values x_n have to be contained in the same connected component c of S^C. This component contains a subsequence of $(x_n)_{n\in\mathbb{N}}$ which again converges to 0. Because c is connected, it has to be of the form $c = (0, r')$ for some $r' > 0$. This proves the first part.

The definition of H gives us

$$\mathsf{H}\epsilon : \phi(\epsilon) \equiv \exists r > 0\ \forall\epsilon \in (0, r) : \phi(\epsilon)$$

which is equivalent to

$$\exists^* r' > 0\ \forall\epsilon \in (0, r') : \phi(\epsilon)$$

(e.g. for $r' \in (\frac{r}{2}, r)$). It holds that

$$\exists r > 0\ \forall\epsilon \in (0, r) : \phi(\epsilon) \quad \Rightarrow \quad \exists r > 0\ \forall^*\epsilon \in (0, r) : \phi(\epsilon)$$

and we can now assume $\exists r > 0\ \forall^*\epsilon \in (0, r) : \phi(\epsilon)$ to be true. This implies that there is a sequence $(x_n)_{x\in\mathbb{N}} \subset (0, r)$ which converges to 0. Because the number of connected components of S is finite and no value of the sequence is equal to 0, almost all values of the sequence have to be contained in the same component of S. For $(x_n)_{x\in\mathbb{N}}$ to converge to 0 this component has to be of the form $(0, r')$ for some $r' > 0$ which implies that:

$$\exists r' > 0\ \forall\epsilon \in (0, r') : \phi(\epsilon) \quad \Leftarrow \quad \exists r > 0\ \forall^*\epsilon \in (0, r) : \phi(\epsilon)$$

This shows that

$$\exists r' > 0\ \forall\epsilon \in (0, r') : \phi(\epsilon) \quad \Leftrightarrow \quad \exists r > 0\ \forall^*\epsilon \in (0, r) : \phi(\epsilon)$$

which again is equivalent to

$$\exists^* r^* > 0\ \forall^*\epsilon \in (0, r^*) : \phi(\epsilon)$$

(e.g. for $r^* \in (\frac{r}{2}, r)$). ∎

Corollary 2.16: *For all formulas ϕ it holds that*

$$\mathsf{H}\epsilon : \phi(\epsilon) \quad \Leftrightarrow \quad \mathsf{A}r > 0\ \mathsf{E}\epsilon \in (0, r) : \phi(\epsilon)$$

for $\mathsf{E} \in \{\exists, \exists^*\}$ *and* $\mathsf{A} \in \{\forall, \forall^*\}$.

Proof. Inserting the second result of Lemma 2.15 into the first gives us

$$\begin{array}{ccl} & \mathsf{H}\epsilon & : \neg\neg\phi(\epsilon) \\ \Leftrightarrow & \neg\mathsf{H}\epsilon & : \neg\phi(\epsilon) \\ \Leftrightarrow & \neg\mathsf{E}r > 0\ \mathsf{A}\epsilon \in (0, r) & : \neg\phi(\epsilon) \\ \Leftrightarrow & \mathsf{A}r > 0\ \mathsf{E}\epsilon \in (0, r) & : \neg\neg\phi(\epsilon). \end{array}$$

∎

2.5 Polynomials

In this chapter we summarize some properties of polynomials which are important to this thesis and relate them to the exotic quantifiers introduced in the last section.

Standard Form A polynomial $p(X_1, ..., X_n)$ over the variables $X_1, ..., X_n$ is in *standard form* if and only if it is written as a sum of monomials.

Lemma 2.17: *A polynomial $p : \mathbb{R}^n \to \mathbb{R}$ in variables x is continuous on $\mathbb{R}^n$:*

$$\forall x_0 \in \mathbb{R}^n \; \forall \epsilon > 0 \;\; \exists \delta > 0 \; \forall x \in B(x_0, \delta) \quad : \quad p(x) \in (p(x_0) - \epsilon, p(x_0) + \epsilon)$$

where $B(c, r) \subset \mathbb{R}^n$ denotes the open ball *around $c \in \mathbb{R}^n$ with radius r.*

Corollary 2.18:

$$\exists x : p(x) > 0 \quad \Leftrightarrow \quad \exists^* x : p(x) > 0$$

More precisely, let $x^* \in \mathbb{R}^n$ be some arbitrary, but fixed value:

$$p(x^*) > 0 \quad \Leftrightarrow \quad \exists \delta > 0 \; \forall x \in B(x^*, \delta) : p(x) > 0$$

Proof. The implication from right to left is trivial. The other direction follows from Lemma 2.17:

Let $p(x^*) > 0$ for x^*. Then choose $x_0 = x^*$, $\epsilon = p(x^*) > 0$. It follows that there exists $\delta > 0$ such that all $x \in B(x^*, \delta)$ satisfy $p(x) > 0$. ∎

Because polynomials are continuous, they inherit a property of all continuous functions:

Lemma 2.19: *Let $D \subseteq \mathbb{R}^n$ be a compact set and $f : D \to \mathbb{R}$ a continuous function on D. Then*

$$\exists m \in \mathbb{R} : m = \inf(f(D)) = \min(f(D))$$

meaning that f actually attains its minimum on D.

Proof. Let $m := \inf(f(D))$ be the infimum of f on D. Then there exists a sequence $x_{(k)} \subset D$ converging to $x^* \in D$ such that

$$\lim_{k \to \infty} f(x_{(k)}) = m$$

If $f(x^*) = m$ then f attains its minimum.

So we can assume $f(x^*) \neq m$. We can now apply Lemma 2.17 for continuous functions. By choosing $x_0 = x^*$, $\epsilon = \frac{|f(x^*) - m|}{2}$ we obtain $\delta > 0$ such that for all $x \in B(x^*, \delta) : f(x) \in (f(x^*) - \epsilon, f(x^*) + \epsilon)$.

There exists a k_d such that $\forall k > k_d : x_{(k)} \in B(x^*, \delta)$. There also exists k_i such that $\forall k > k_i : f(x_{(k)}) \in (m - \epsilon, m + \epsilon)$.

This is clearly a contradiction proving the lemma. ∎

Next we look at the zero sets of (multivariate) polynomials and how these relate to the zero polynomial.

Theorem 2.20 ([Alo99, Theorem 1.2])**:** *Let $p \in \mathbb{R}[X_1, ..., X_n]$ be a polynomial with degree Δ. If $S_1, ..., S_n \subseteq \mathbb{R}$ with $|S_i| > \Delta$ for all $i \in \{1, ..., n\}$ then*

$$\exists x_1 \in S_1 \dots \exists x_n \in S_n : p(x_1, ..., x_n) \neq 0$$

or p is zero on $\mathbb{R}^n$.

Corollary 2.21: *Let $p \in \mathbb{R}[X_1, ..., X_n]$ be a polynomial with degree Δ. Then the following statements are equivalent:*

- *p is zero on all $\mathbb{R}^n$*

- $\forall x : p(x) = 0$
- $\exists x_1 \ldots \exists x_{(\Delta+1)^n} : \bigwedge_{i,j} x_i \neq x_j \quad \wedge \quad \bigwedge_i p(x_i) = 0$
- $\exists^* x : p(x) = 0$
- $\forall^* x : p(x) = 0$

For non-zero univariate polynomials there is also a lower bound on the distance between two distinct roots, called the *separation*. For higher dimensions such a separation result seems unlikely. The zero sets of multivariate polynomials can contain whole hyperplanes in which distinct zeroes can have arbitrary distance.

Theorem 2.22 ([Mig92, Theorem 4.6]): *Let $p : \mathbb{R} \to \mathbb{R}$ be a non-constant univariate polynomial of degree Δ. We define the* separation *of p as*

$$sep(p) \quad := \quad \min\{\ |a_i - a_j| \ \mid \ a_i, a_j \text{ are distinct roots of } p\ \}$$

or $sep(p) = \infty$ if p has no two distinct roots.

If p has integer coefficients then we can bound

$$sep(p) > \frac{1}{\Delta^{(\Delta+2)/2} \cdot \|p\|^{\Delta-1}}$$

where $\|p\|$ is the (euclidean) norm of the coefficient vector of p.

3 The Problem ETR and $\exists\mathbb{R}$

The problem ETR is especially interesting as many problems from geometric graph theory and game theory are contained in $\exists\mathbb{R}$ and can therefore be reduced to ETR. Among them are the art gallery problem, the recognition of segment graphs, the stretchability of pseudolines, the Steinitz problem or the existence of Nash equilibria [Mat14] [Bjö+99] [SŠ17].

Existential Theory of the Reals The *existential theory of the reals* (ETR) is the problem of deciding whether a given *existential sentence* of the first-order theory of the reals is true or false. This is equivalent to deciding whether a given semialgebraic set is not empty (see Definition 2.10):

Definition 3.1 (see Definition 1.3)**:** *Let $\phi(x)$ be a quantifier-free formula. ETR is the problem deciding*

$$\exists x : \phi(x)$$

which is equivalent to deciding whether

$$S := \{x \in \mathbb{R}^n : \phi(x)\}$$

is non-empty.

Definition 3.2 (see also Definition 1.4)**:** $\exists\mathbb{R}$ *is the class of problems which contains ETR and all problems that can (in polynomial time) be reduced to it. Therefore ETR is $\exists\mathbb{R}$-complete.*

Theorem 3.3: *ETR is* NP*-hard.*

Proof. We reduce SAT in polynomial time to ETR:

Given a Boolean formula ϕ_B over the Boolean variables $A_1, ..., A_n$. For each Boolean variable A_i we introduce one real variable X_i. We construct $\phi_{\mathbb{R}}$ by replacing every occurrence of A_i by $X_i \geq 0$:

$$\phi_{\mathbb{R}} := \phi_B[A_i/(X_i \geq 0)]$$

Note that by replacing $X_i \geq 0$ in $\phi_{\mathbb{R}}$ by A_i we again obtain ϕ_B:

$$\phi_B = \phi_{\mathbb{R}}[(X_i \geq 0)/A_i]$$

This construction requires linear time and is therefore polynomial.

We now prove the equivalence of the two formulas:

$$\exists a \in \{\text{false}, \text{true}\} : \phi_B(a) \quad \Leftrightarrow \quad \exists x \in \mathbb{R}^n : \phi_C(x)$$

First assume $\phi_B(a^*)$ to be true for some certificate a^*. We can now choose

$$x_i^* := \begin{cases} 1 & , \text{ if } a_i^* = \text{ true} \\ -1 & , \text{ if } a_i^* = \text{ false} \end{cases}$$

which ensures that $x_i^* \geq 0 \Leftrightarrow a_i^*$. Therefore replacing a_i^* in ϕ_B by $x_i^* \geq 0$ does not change its truth value and $\phi_{\mathbb{R}}$ is also true.

Now assume $\phi_{\mathbb{R}}(x^*)$ to be true for some x^*. We can choose

$$a_i^* := \begin{cases} \text{true} & , \text{ if } x_i^* \geq 0 \\ \text{false} & , \text{ if } x_i^* < 0 \end{cases}$$

which ensures again $a_i^* \Leftrightarrow x_i^* \geq 0$. Therefore replacing $x_i^* \geq 0$ in $\phi_{\mathbb{R}}$ by a_i^* does not change its truth value and $\phi_B(a^*)$ is also true. ■

Corollary 3.4: NP ⊆ ∃ℝ.

Theorem 3.5 ([Can88])**:** ∃ℝ ⊆ PSPACE

This puts the complexity class ∃ℝ between NP and PSPACE.

3.1 Restrictions on ETR preserving ∃ℝ-completeness

Similarly to how we can restrict SAT to ,for example, 3-SAT without losing NP-completeness, we can restrict ETR without losing ∃ℝ-completeness:

- INEQ restricts the formula to a conjunction of polynomial equations and inequalities with all polynomials in standard form.
- Strict-INEQ is the special case of INEQ which only allows strict inequalities.
- Feasible is the special case of INEQ with only a single (multivariate) polynomial equation $\exists x : p(x) = 0$.

Theorem 3.6 ([Mat14, Propositions 3.2 and 3.5])**:** *INEQ, Strict-INEQ and Feasible are all ∃ℝ-complete.*

Before we can prove Theorem 3.6 we have to introduce some additional results which greatly simplify our proof. While this work is not new (we follow the proof of Matoušek closely), it should provide a lot of intuition and tools for further work with the general real polynomial hierarchy. So can, for example, Lemma 3.7 often be applied to obtain a more restricted standard form.

Lemma 3.7 ([SŠ17, Lemma 3.2] [SŠ22, Lemma 1.6])**:** *Let ϕ be a quantifier-free formula in free variables $X_1, ..., X_n$ and L the length of that formula. Then it is possible, in polynomial time, to create*

- *a set $F = \{p_1, ..., p_k\}$ of quadratic polynomials $p_i : \mathbb{R}^{n+m} \to \mathbb{R}$ in standard form, such that*

$$\phi(x) \quad \Leftrightarrow \quad \left(\exists y \in \mathbb{R}^m : \bigwedge_{1 \leq i \leq k} p_i(x, y) = 0 \right)$$

  for all $x \in \mathbb{R}^n$. Additionally $m, k \leq 3L$ and the coefficients in p_i have bit length at most L.

- *a non-negative quartic polynomial $g : \mathbb{R}^{n+m} \to \mathbb{R}$ in standard form, such that*

$$\phi(x) \quad \Leftrightarrow \quad (\exists y \in \mathbb{R}^m : q(x,y) = 0)$$

for all $x \in \mathbb{R}^n$. Additionally $m \leq 3L$ and the coefficients in q have bit length at most $2L$.

Proof. With Theorem 2.9 we can transform the given formula ϕ into an algebraic circuit C. This allows us to apply Theorem 2.8 on C which gives us an equivalent conjunction of polynomial equations of degree at most two. This shows the first part.

For the second part we obtain polynomials p_i from the first part. Now finding a zero of all p_i is equivalent to finding a zero of the polynomial $\sum_i p_i^2$:

$$\exists x : \bigwedge_i p_i(x) = 0 \quad \Leftrightarrow \quad \exists x : \sum_i p_i(x)^2 = 0$$

The size of $\sum_i p_i^2$ is at most double the size of the conjunction and has at most double the degree of each p_i. This shows the second part. ∎

Another important theorem required for the proof of Theorem 3.6 is a result which we nickname the *ball theorem*. It and its corollary allow us to bound certain properties of polynomials. While both bounds are doubly exponential, we can see in the proof of Corollary 3.9 how we can encode these bounds into a polynomial size formula. This idea is used in many other proofs later on.

Theorem 3.8 (Ball Theorem [Mat14, Theorem 3.4] [BPR96] [BPR03] [BV07])**:** *Let $S \subseteq \mathbb{R}^n$ be a semialgebraic set defined by a quantifier-free formula of length L. If $S \neq \emptyset$ then every connected component of S intersects the ball of radius $R = 2^{2^{CL\log L}}$ centered at 0, where C is a suitable absolute constant. Every bounded connected component is contained within the ball.*

Corollary 3.9: *Let $p(X) > 0$ be a positive polynomial in $X_1, ..., X_n$ with no zeroes and length L. Define $R = 2^{2^{CL\log L}}$ and $R' = 2^{2^{C'L\log(L)^2}}$ for suitable constants C, C'. It then holds that*

$$p(x) > \frac{1}{R'}$$

for all $x \in B(0,R)$.

Proof. Consider the formula

$$\begin{aligned} \phi := \exists (X,Y,Z) \in \mathbb{R}^{n+k+1} : \quad & Z^2 \cdot p(X_1, ..X_n) = 1 \\ \wedge \quad & Y_1 < 4 \ \wedge \ \textstyle\bigwedge_{i=2}^k Y_i < Y_{i-1}^2 \\ \wedge \quad & \textstyle\bigwedge_{i=1}^k Y_i > 0 \\ \wedge \quad & \textstyle\sum_{i=1}^n X_i^2 < Y_k^2 \end{aligned}$$

with $k = CL\log(L)$. This formula has length $K = C^*L\log(L)$ for a suitable constant $C^* > C$. It ensures that $Y_k < R$ and therefore also $\|X\| < R$.

From Lemma 2.19 we know that $p(x)$ assumes a minimum for $x \in B[0,R]$. Because p has no zero, this minimum is positive. Therefore the semialgebraic set described by ϕ is bounded (X is bounded by definition of ϕ and Z is bounded by the inverse of the minimum of p).

Applying Theorem 3.8 yields an upper bound for Z, namely $Z < 2^{2^{CK\log K}} = R'$. This however implies a lower bound on $p(x)$ for $x \in B[0,R]$, namely $p(x) \geq \frac{1}{Z} > \frac{1}{R'}$. ∎

With the above theorems and corollaries, we are now suitably equipped to prove Theorem 3.6 in full.

Proof of Theorem 3.6 [Mat14, Propositions 3.2 and 3.5]. First note that all problems are restrictions on ETR and therefore contained in ∃ℝ.

Lemma 3.7 already shows that every instance of ETR can be reduced in polynomial time to an instance of Feasible. Therefore Feasible is ∃ℝ-complete.
By definition Feasible is a restriction on INEQ and therefore INEQ is also ∃ℝ-complete.

To prove the ∃ℝ-completeness of Strict-INEQ we reduce from Feasible.

Let $I := (\exists x \in \mathbb{R}^n : p(x) = 0)$ be an instance of Feasible. From Theorem 3.8 we now know that if I is satisfiable then there exists a certificate $x^* \in B(0, R)$ inside the ball of radius $R = 2^{2^{CL \log L}}$ where L is the length of p.

We now construct the formula ψ in free variables $X_1, ..., X_n, Y_1, ..., Y_k, Z_1, ..., Z_l$

$$\begin{array}{lll} & \bigwedge_{i=1}^{k} Y_i > 0 & \\ \wedge & Y_1 < 4 \;\; \wedge \;\; \bigwedge_{i=2}^{k} Y_i < Y_{i-1}^2 & \text{ensures } Y_k < R \\ \wedge & \sum_{i=1}^{n} X_i^2 < Y_k^2 & \text{ensures } \|X\| < R \\ \wedge & Z_1 > 4 \;\; \wedge \;\; \bigwedge_{i=2}^{l} Z_l > Z_{l-1}^2 & \text{ensures } Z_l > 2^{2^l} \\ \wedge & Z_l^2 \cdot p(x)^2 < 1 & \end{array}$$

where $k = CL\log(L)$ and $l = C'L\log(L)^2$ for a suitable constants C' (see Corollary 3.9). First note that this formula can be constructed in polynomial time and only requires polynomial space, even when we convert all polynomials to standard form.

Now we prove that ψ is an equivalent instance of Strict-INEQ:

$$\exists x : p(x) = 0 \quad \Leftrightarrow \quad \exists (x, y, z) : \psi(x, y, z)$$

If $p(x^*) = 0$ for some $x^* \in B(0, R)$ then trivially $\|x^*\| < R$ which allows us to find suitable Y which fulfill the conditions. The condition $Z_l^2 \cdot p(x^*)^2 < 1$ holds for any Z_l which allows us to find Z inductively as the final value of Z_l does not matter.

Now assume that $p(x) > 0$ for all $x \in \mathbb{R}$. Applying Corollary 3.9 gives us a lower bound on $p(x)$ for $x \in B(0, R)$, namely $p(x) > \frac{1}{R'}$ for $R' = 2^{2^l}$.

However, we also know that in any solution for ψ it holds that $Z_l > R'$. Therefore $Z_l^2 \cdot p(x)^2 > R' \cdot \frac{1}{R'} = 1$ which violates the last condition in ψ.

Therefore ψ is an equivalent instance of Strict-INEQ which concludes the proof. ∎

We now look at a property related to Strict-INEQ which we apply directly afterwards and also helps in later proofs:

Theorem 3.10: *Any semialgebraic set defined by a quantifier-free formula using only strict inequalities and no negations is open.*

Proof. Because every semialgebraic set is the union of basic semialgebraic sets and the union of open sets is open, we only have to proof Theorem 3.10 for basic semialgebraic sets (see Section 2.3).

Let $S = \{x \in \mathbb{R}^n : \bigwedge_{p \in P} p(x) > 0\}$ be such a basic semialgebraic set defined by a conjunction of strict polynomial inequalities. Define the function $P_{\max} : \mathbb{R}^n \to \mathbb{R}; x \mapsto \max_{p \in P} p(x)$. As the maximum of continuous functions, $P_{\max}$ is also continuous.

From the definition of continuity we know that for any x_0 and $\epsilon > 0$ we can obtain $\delta > 0$ such that

$$\forall x \in B(x_0, \delta) : P_{\max}(x) \in (P_{\max}(x_0) - \epsilon, P_{\max}(x_0) + \epsilon)$$

For a given $x_0 \in S$ we can choose $\epsilon = P_{\max}(x_0) > 0$. Now all $x \in B(x_0, \delta)$ fulfill

$$P_{\max}(x) > P_{\max}(x_0) - P_{\max}(x_0) = 0$$

and are therefore contained in S.

But now we can find for any value $x \in S$ a neighbourhood which also lies in S which implies that S is open. ∎

We now prove an inclusion of two complexity classes in the (discrete) polynomial hierarchy. The class $\exists^*\mathbb{R}$ is defined in Definition 1.4 and again in Definition 4.2. The equality of these complexity classes is shown with the help of this corollary in Theorem 4.13.

Corollary 3.11: $\exists\mathbb{R} \subseteq \exists^*\mathbb{R}$.

Proof. Let $I = (\exists x : \phi(x))$ be an instance of Strict-INEQ. By replacing the quantifier $\exists$ by $\exists^*$ in I we obtain an instance J of Standard($\exists^*$).

We know that by definition ϕ is a conjunction of strict polynomial inequalities. Therefore, we can apply Theorem 3.10 which states that the set $S_I = \{x \mid \phi(x)\}$ is open.

If I is true, then S_I is non-empty and therefore contains an open ball of full dimension. This implies that J is also true. If I is false, then S_I is empty and therefore J is also false. We have shown that we can transform any instance of Strict-INEQ into an equivalent instance of Standard($\exists^*$) which concludes the proof. ∎

4 The Real Polynomial Hierarchy

4.1 Two Definitions

As already mentioned in Section 1.1, the concept of $\exists\mathbb{R}$ can be generalised to form the *real polynomial hierarchy*. There are two definitions of the real polynomial hierarchy: The first is by Bürgisser and Cucker which is based on a computation model over the real numbers by Blum, Shub and Smale (*BSS-model*). They use *algebraic circuits* to define those complexity classes arising from exotic quantifiers (defined in Section 2.4). Recall also that we omit the domain of quantified variables wherever possible (see Section 2.4).

Definition 4.1 ([BC09, Chapter 3])**:** *We define the problem* Standard-BSS($Q_1, \ldots, Q_\omega$) *as deciding whether for a given algebraic decision circuit* C *the following sentence is true:*

$$Q_1 x_1 \ldots Q_\omega x_\omega : C(x_1, x_2, \ldots, x_\omega) = 1$$

with $Q_i \in \{\exists, \forall, \exists^*, \forall^*, H\}$ *for* $i \in \{1, .., \omega\}$*.*

Additionally define Standard-BSS($\emptyset$) *as deciding whether* $C(\emptyset) = 1$*. The circuit* C *may contain real constants.*

The second definition of the real polynomial hierarchy, which we call the *discrete* real polynomial hierarchy, is by Schaefer and Štefankovič who define it over the standard discrete Turing machine model. We have already given this definition in Section 1.1.1, but we restate it here out of convenience.

Definition 1.3: *We define the problem* Standard($Q_1 \ldots Q_\omega$) *as deciding whether for a given quantifier-free formula* ϕ *the following sentence is true:*

$$Q_1 x_1 \ldots Q_\omega x_\omega : \phi(x_1, x_2, \ldots, x_\omega)$$

with $Q_i \in \{\exists, \forall, \exists^*, \forall^*, H\}$*.*

According to these definitions we define complexity classes:

Definition 4.2:

- *The complexity class* $Q_1 \ldots Q_\omega$ *is the class for which* Standard-BSS($Q_1, \ldots, Q_\omega$) *is complete.*
- *The complexity class* $Q_1 \ldots Q_\omega\mathbb{R}$ *is the class for which* Standard($Q_1, \ldots, Q_\omega$) *is complete.*

We call the union of complexity classes defined by the Standard-BSS($Q_1, \ldots, Q_\omega$) problems, the *real polynomial BSS-hierarchy*, or short BSS-hierarchy. Similarily we call the union of complexity classes defined through the Standard($Q_1, \ldots, Q_\omega$) problems, the *discrete real polynomial hierarchy*. If we speak of the real polynomial hierarchy we mean the discrete version.

Corollary 4.3:

- *The complexity class "$\emptyset$", i.e. the class for which* Standard-BSS($\emptyset$) *is complete, is equal to* $\mathsf{P}_{\mathbb{R}}$.
- *The complexity class* $\emptyset\mathbb{R}$, *i.e. the class for which* Standard($\emptyset$) *is complete, is equal to* P.

This comes from the fact that we can evaluate an algebraic circuit in polynomial time in the BSS-model and a formula in polynomial time in the Turing machine model.

It follows that ETR = Standard($\exists$) which is $\exists\mathbb{R}$-complete.

Because the definition of the BSS-hierarchy allows for real constants and input values, it is generally incompatible with the discrete definition. It is also independent of the size of numbers (input values, constants, values created during computation) whereas the *discrete* hierarchy has to encode them over $\{0,1\}$. A potential compromise could be the *constant-free* and *Boolean part* (BP^0) of the BSS-hierarchy. For this restriction, there are a few known relations between these hierarchies:

Lemma 4.4: *Let C be a sequence of quantifiers, describing a complexity class in the BSS-hierarchy and $C\mathbb{R}$ the related complexity class in the discrete hierarchy:*

$$C\mathbb{R} \subseteq \mathsf{BP}^0(C)$$

$$C\exists\mathbb{R} = \mathsf{BP}^0(C\exists) \text{ and } C\forall\mathbb{R} = \mathsf{BP}^0(C\forall)$$

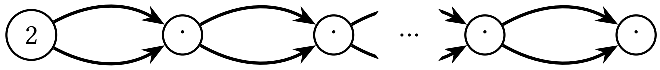
Proof. The first inclusion stems from the fact that every formula can be transformed into an algebraic circuit in polynomial time (see Theorem 2.9). The reverse is only true when we are allowed to introduce additional existentially quantified variables to represent the nodes of the circuit (see Theorem 2.8). If the last quantifier in C is $\forall$, we can first negate the whole statement, then convert the circuit to a formula and lastly negate again. This proves the equalities. ∎

It is currently unknown whether $\mathsf{BP}^0(\mathsf{P}_{\mathbb{R}}) =: \mathsf{PR} \stackrel{?}{=} \mathsf{P}$. And in fact, Bürgisser and Cucker refer to Allender, Kjeldgaard-Pedersen, Bürgisser and Miltersen for evidence to the contrary [AKBM06]. This is likely a result from the fact that the BSS-hierarchy uses a different computation model or that algebraic circuits and formulas seemingly differ in encoding complexity as discussed in the next section.

4.1.1 Algebraic Circuits and Formulas

In this thesis we almost exclusively focus on the discrete real polynomial hierarchy, as defined in Definition 1.3 and Definition 4.2. However, we want to use this chapter to highlight some key differences between formulas and circuits that become relevant later. More precisely, we look at circuits in the discrete setting (constant-free and Boolean part BP^0), i.e. with only the constants 0 and 1 and evaluated by a discrete computation model instead of the BSS-model.

Figure 4.1: Repeated multiplication in an algebraic circuit



Number Encoding Similar to normal binary encoding, we need $O(\log(n))$ bits to encode the number $n \in \mathbb{Z}$ in a formula. To encode a value of size 2^{2^n} in a formula, we would need $O(\log(2^{2^n})) = O(2^n)$ bits, i.e. exponential space. However, an algebraic circuit can write repeated multiplication in a far more efficient manner: Figure 4.1

In essence, an algebraic circuit can reuse constructed values more than once. To achieve the same effect with a formula we have to introduce new variables that store these values. This is done in Theorem 2.8 to convert a circuit in polynomial time and space into a formula. This leads to an even more fundamental difference between formulas and circuits:

Lemma 4.5: *Let A be an arbitrary computation consisting of n arithmetic operations $(+, -, \cdot, \div)$ involving k integer constants $c_1, \ldots, c_k$ and m input values $x_1, \ldots, x_m$. We can encode A into an algebraic circuit with $1 + n + m + O(\sum_{i=1}^{k} \log(c_i))$ gates.*

Proof. First construct for each constant c_i ($i \in \{1, \ldots, k\}$) an algebraic circuit which computes it. This can clearly be done in at most $O(\log(c_i))$ space. Add for every input value x_i ($i \in \{1, \ldots, m\}$) a single input gate.

Now we can iterate over every step in the computation in topological order and do the following: If the operation in the current step is $\circ$ and it operates on two values p and q, we add an arithmetic gate into our algebraic circuit with operation $\circ$ and incoming edges from the already constructed sub-circuits for p and q. These have to exist as each step in the computation A can only access constants, input values or previously calculated values.

The last added gate outputs exactly the same values as the computation A which is done with a last output gate. ∎

The above proof makes use of the fact that algebraic circuits can reuse previously computed values. The same is not true for formulas. Instead, in a formula, each operand has to be constructed anew every time it is needed. Translating the computation in Figure 4.1 into a formula without introducing additional variables would yield

$$(((2 \cdot 2) \cdot (2 \cdot 2)) \ldots ((2 \cdot 2) \cdot (2 \cdot 2)))$$

where the constant 2 appears 2^n times and is therefore not of polynomial size.

Under this observation it seems unlikely that we can find a polynomial transformation between algebraic circuits and formulas. However, in Lemma 4.4 we show that a single $\exists$ or $\forall$ quantifier can absorb this difference and provide us with a polynomial transformation.

Now the question remains if the same holds for $\exists^*$ and $\forall^*$ quantifiers (and to a lesser degree for H as well). To this end we show an intermediary result which may be extended in the future to show that both hierarchies coincide.

Theorem 4.6: *Let C be an algebraic decision circuit over $x \in \mathbb{R}^n$ with only the constants 0 and 1. It is possible to construct, in polynomial time, a formula ϕ_C^* over $(x, \epsilon, g) \in \mathbb{R}^{O(n)}$ such that for all x*

$$C(x) = 1 \quad \Leftrightarrow \quad \mathsf{H}\epsilon\; \exists^* g : \phi_C^*(x, \epsilon, g).$$

We prove Theorem 4.6 in four steps. First we describe the construction of ϕ_C^*, then we show a first equivalence between this construction and the original circuit, then we show the $\Rightarrow$ part and lastly we conclude the proof.

Intuition To make it easier to follow the construction and the proof, we first provide some intuition about the ideas used there.

First note that the only decision that can be made in an algebraic circuit stems from sign gates. These, however, can only distinguish between a value less than 0 and greater or equal to 0. It is therefore not necessary to model a fully accurate computation, as long as the sign of the computed values is identical.

We compute instead open intervals (g_l, g_h) which shall either be entirely negative, entirely positive or collapsed to the value 0. This last case is represented by an additional variable g_0 which signifies whether a collapsed interval is present ($g_0 > 0$) or not ($g_0 < 0$).

The construction works because the binary operations $+, -, \cdot, \div$ are continuous on $(\mathbb{R} \setminus \{0\})^2$. We can apply such an operation $\circ$ (component-wise) onto two open intervals I and J that are either entirely negative or positive respectively. In the case of $\circ \in \{\cdot, \div\}$ the resulting interval $I \circ J$ is again either entirely positive or negative (or collapsed). For $\circ \in \{+, -\}$ it is a little more difficult and we require the help of an H quantifier to obtain similar results.

To simplify the construction and proof, we eliminate all additions $x + y$ by replacing them with $x - (0 - y)$.

To make the formulas introduced in the construction readable we write them in interval notation, i.e. $(p_l, p_h) \circ (q_l, q_h) \subset (g_l, g_h)$. These can be expanded into formulas as

$$\begin{aligned} p_l \circ q_l &\in (g_l, g_h) \\ p_l \circ q_h &\in (g_l, g_h) \\ p_h \circ q_l &\in (g_l, g_h) \\ p_h \circ q_h &\in (g_l, g_h) \end{aligned}$$

and these again to

$$\begin{aligned} r &< g_h \\ r &> g_l \end{aligned}$$

The condition $(p_l, p_h) \cap (q_l, q_h) = \emptyset$ can be expressed similarily.

Construction Let C be an algebraic decision circuit over $x \in \mathbb{R}^n$ with only the constants 0 and 1.

First introduce for each gate g in C three variables g_0, g_l and g_h which model the interval (g_l, g_h) and the formulas

$$g_h - \epsilon \;<\; g_l \;<\; g_h \;<\; g_l + \epsilon \quad \text{(at most } \epsilon \text{ long)} \tag{4.1}$$

$$(g_l > 0) \;\vee\; (g_h < 0) \quad \text{(entirely positive or negative)}.$$

We construct a formula ϕ_C^* as a conjunction of sub-formulas as follows:

- For an input gate (or constant gate) g for the variable x (or the constant value x), add the formulas

$$\begin{aligned}(x \neq 0) &\quad\Rightarrow\quad x \in (g_l, g_h)\\ (x = 0) &\quad\Leftrightarrow\quad g_0 > 0.\end{aligned}$$

- For an arithmetic gate g with operation $-$ and parent gates p and q, add the formulas

$$\begin{aligned}((p_l, p_h) \cap (q_l, q_h) = \emptyset) \wedge (p_0 < 0 \wedge q_0 < 0) &\Rightarrow (p_l, p_h) - (q_l, q_h) \subset (g_l, g_h)\\ (p_0 > 0 \wedge q_0 < 0) &\Rightarrow 0 - (q_l, q_h) \subset (g_l, g_h)\\ (p_0 < 0 \wedge q_0 > 0) &\Rightarrow (p_l, p_h) - 0 \subset (g_l, g_h)\end{aligned}$$

$$\begin{aligned}((p_l, p_h) \cap (q_l, q_h) \neq \emptyset \wedge (p_0 < 0 \wedge q_0 < 0))&\\ \vee\quad &\Leftrightarrow g_0 > 0.\\ (p_0 > 0 \wedge q_0 > 0)&\end{aligned}$$

- For an arithmetic gate g with operation $\cdot$ and parent gates p and q, add the formulas

$$\begin{aligned}(p_0 < 0 \wedge q_0 < 0) &\Rightarrow (p_l, p_h) \cdot (q_l, q_h) \subset (g_l, g_h)\\ (p_0 > 0 \vee q_0 > 0) &\Leftrightarrow g_0 > 0.\end{aligned}$$

- For an arithmetic gate g with operation $\div$ and parent gates p and q, add the formulas

$$\begin{aligned}(p_0 < 0 \wedge q_0 < 0) &\Rightarrow (p_l, p_h) \div (q_l, q_h) \subset (g_l, g_h)\\ (p_0 > 0 \wedge q_0 < 0) &\Leftrightarrow g_0 > 0\\ (q_0 > 0) &\Rightarrow \text{false}.\end{aligned}$$

- For a sign gate g with parent gate p, add the formulas

$$\begin{aligned}(p_0 < 0 \wedge p_h < 0) &\Leftrightarrow g_0 > 0\\ g_0 < 0 &\Rightarrow 1 \in (g_l, g_h).\end{aligned}$$

- For the single output gate g with parent sign gate p, add the formula $p_0 < 0$.

The construction is clearly polynomial as we only introduce a fixed number of variables and fixed sized formulas for each gate of the circuit.

Recall that with C_g we mean the sub-circuit of C which outputs the result of g (see Definition 2.6).

Lemma 4.7: *If* Hϵ $\exists^* g : \phi^*_C(x, \epsilon, g)$ *holds, for given fixed* x*, then for all gates* g *in* C *it holds that*

$$\big(C_g(x) \in (g_l, g_h) \wedge g_0 < 0\big) \;\vee\; \big(C_g(x) = 0 \wedge g_0 > 0\big)$$

Proof. We prove the claim by structural induction in topological ordering.

Induction Base Case There are two base cases which are treated identically in the construction: input and constant gates. For $x \neq 0$ we obtain $\big(C_g(x) = x \in (g_l, g_h) \land g_0 < 0\big)$ by definition. For $C_g(x) = x = 0$ we obtain $g_0 > 0$ by definition.

Induction Step Let g be the current gate with parent gate(s) p (and q). We know that

$$\big(C_p(x) \in (p_l, p_h) \land p_0 < 0\big) \quad \lor \quad \big(C_p(x) = 0 \land p_0 > 0\big)$$

(and the same for q).

- **g is a sign gate**

  If $\big(C_p(x) \in (p_l, p_h) \land p_0 < 0\big)$ and $p_h < 0$, then $\big(C_g(x) = 0 \land g_0 > 0\big)$.

  All other cases imply $\neg(p_0 < 0 \land p_h < 0)$ and therefore $C_g(x) = 1 \in (g_l, g_h)$.

- **g is an arithmetic gate with operation $\cdot$**

  If $\big(C_p(x) = 0 \land p_0 > 0\big)$ or $\big(C_q(x) = 0 \land q_0 > 0\big)$, then $\big(C_g(x) = 0 \land g_0 > 0\big)$.

  In the other case $(p_0 < 0 \land q_0 < 0)$ and therefore

  $$C_g(x) = C_p(x) \cdot C_q(x) \in (p_l, p_h) \cdot (q_l, q_h) \subseteq (g_l, g_h).$$

- **g is an arithmetic gate with operation $\div$**

  $\big(C_q(x) = 0 \land q_0 > 0\big)$ cannot be true as ϕ_C^* would then be false which is a contradiction to our assumption.

  If $\big(C_p(x) = 0 \land p_0 > 0\big)$, then $\big(C_g(x) = 0 \land g_0 > 0\big)$.

  In the other case $(p_0 < 0 \land q_0 < 0)$ and therefore

  $$C_g(x) = C_p(x) \div C_q(x) \in (p_l, p_h) \div (q_l, q_h) \subseteq (g_l, g_h).$$

- **g is an arithmetic gate with operation $-$**

  If $\big(C_p(x) \in (p_l, p_h) \land p_0 < 0\big)$ and $\big(C_q(x) = 0 \land q_0 > 0\big)$, then

  $$C_g(x) = C_p(x) - 0 \in (p_l, p_h) \subseteq (g_l, g_h).$$

  If $\big(C_p(x) = 0 \land p_0 > 0\big)$ and $\big(C_q(x) \in (q_l, q_h) \land q_0 < 0\big)$, then

  $$C_g(x) = 0 - C_q(x) \in 0 - (q_l, q_h) \subseteq (g_l, g_h).$$

  If $\big(C_p(x) = 0 \land p_0 > 0\big)$ and $\big(C_q(x) = 0 \land q_0 > 0\big)$, then $\big(C_g(x) = 0 + 0 = 0 \land g_0 > 0\big)$.

  Now if $C_p(x) = C_q(x)$ then $(p_l, p_h) \cap (q_l, q_h) \neq \emptyset$ and therefore

  $$\big(C_g(x) = C_p(x) - C_q(x) = 0 \land g_0 > 0\big)\,.$$

  Else if $C_p(x) \neq C_q(x)$ then w.l.o.g. $C_p(x) < C_q(x)$ and there exists $0 < \epsilon < |C_p(x) - C_q(x)|/2$ and with that

  $$\begin{aligned} p_h &< p_l + \epsilon && \textit{Equation } (4.1) \\ &< C_p(x) + \epsilon && C_p(x) > p_l \\ &< C_q(x) - \epsilon && \epsilon < |C_p(x) - C_q(x)|/2 \\ &< q_h - \epsilon && C_q(x) < q_h \\ &< q_l && \textit{Equation } (4.1) \end{aligned}$$

This implies $(p_l, p_h) \cap (q_l, q_h) = \emptyset$ and therefore

$$\left(C_g(x) = C_p(x) - C_q(x) \in (p_l, p_h) - (q_l, q_h) \subseteq (g_l, g_h)\right).$$

∎

Lemma 4.8: *If, for given x, $C(x) = 1$ then* $\mathsf{H}\epsilon\ \exists^* g : \phi^*_C(x, \epsilon, g)$.

Proof. We first drop the condition of Equation (4.1). Now we can, in topological order of the circuit, greedily choose values for g_0, g_l and g_h that satisfy the construction. With this choice we are almost certain to violate Equation (4.1) and we have to repair it.

Repairing $(g_h - \epsilon < g_l)$ Let (g_l, g_h) be such an interval for which holds

$$g_l \leq g_h - \epsilon.$$

If $g_0 > 0$ then we can just chose $0 < g_l < g_h < g_l + \epsilon$ without affecting any other conditions.
If g is an input, constant or sign gate we can choose

$$C_g(x) - \epsilon/2 < g_l < C_g(x) < g_h < C_g(x) + \epsilon/2$$

without affecting other conditions.
Else g is an arithmetic gate with operation $\circ \in \{+, -, \cdot, \div\}$ and parent gates p and q.. Let

$$\min\nolimits_\circ := \min\left((p_l, p_h) \circ (q_l, q_h)\right)$$
$$\max\nolimits_\circ := \max\left((p_l, p_h) \circ (q_l, q_h)\right).$$

If $|\min_\circ - \max_\circ| < \epsilon$ we can choose

$$\max\nolimits_\circ - \epsilon < g_l < \min\nolimits_\circ < C_g(x) < \max\nolimits_\circ < g_h < \min\nolimits_\circ + \epsilon$$

without affecting other conditions.
Otherwise if $|\min_\circ - \max_\circ| \geq \epsilon$ we can leverage the continuity of $\circ$ on $(p_l, p_h) \times (q_l, q_h)$ to obtain for $a_0 = C_p(x)$, $b_0 = C_q(x))$ that

$$\exists \delta > 0\ \forall a \in (a_0 - \delta, a_0 + \delta)\ \forall b \in (b_0 - \delta, b_0 + \delta) : a \circ b \in (C_g(x) - \epsilon/2, C_g(x) + \epsilon/2).$$

Therefore, by introducing the additional conditions $p_h - 2\delta < p_l$ and $q_h - 2\delta < q_l$ we can repair the condition $g_h - \epsilon < g_l$.
Because these new conditions have the same form as the original one we can, should they be violated, recursively repeat this argument to repair all of these conditions in the circuit C_g. This recursion terminates because a circuit is a directed *acyclic* graph.

By repairing all intervals that violate $g_h - \epsilon < g_l$ in the above way we obtain

$$\mathsf{H}\epsilon\ \exists g : \phi^*_C(x, \epsilon, g).$$

Because all inequalities in ϕ^*_C are strict, we can apply (for fixed x and ϵ) Theorem 3.10 to derive

$$\mathsf{H}\epsilon\ \exists^* g : \phi^*_C(x, \epsilon, g).$$

∎

Proof of Theorem 4.6. We consider the circuit C and formula ϕ^*_C as above. Let x be arbitrary but fixed.

We know from Lemma 4.8 that $C(x) = 1$ implies $\mathsf{H}\epsilon\ \exists^* g : \phi(x, \epsilon, g)$.

Now assume that $C(x) = 0$ but $\mathsf{H}\epsilon\ \exists^* g : \phi(x, \epsilon, g)$ still true. In this case we know from Lemma 4.7 that for the last sign gate g (which feeds into the single output gate) the following holds

$$\left(C_g(x) \in (g_l, g_h) \wedge g_0 < 0\right) \quad \vee \quad \left(C_g(x) = 0 \wedge g_0 \geq 0\right).$$

Because our formula is true we also know that $g_0 < 0$ (condition from the single output gate) which implies $1 \in (g_l, g_h)$ and with that $0 < g_l < C_g(x)$. This however shows that the output of the last sign gate is 1 and therefore $C(x) = 1$ which is a contradiction to our assumption. ∎

Consequences If the single H quantifier could be, in polynomial time, removed we would have shown that an algebraic decision circuit can be transformed, in polynomial time, into a formula quantified by $\exists^*$. This implies that $\mathsf{BP}^0(C\exists^*) \subseteq C\exists^*$ and by negation, application and negation (see Lemma 4.9) $\mathsf{BP}^0(C\forall^*) \subseteq C\forall^*$. This would strengthen Lemma 4.4 such that the discrete real polynomial hierarchy and the constant-free Boolean part of the BSS-hierarchy coincide. The case where H is the last quantifier would be handled by Theorem 4.11 which allows us to remove it at the end of the quantifier sequence. We can repeat this until the last quantifier is no longer H in which case we can use the other results.

4.2 Helpful Results

Before we show inclusions between the complexity classes in the (discrete) real polynomial hierarchy, we note some results which are helpful in further proofs.

The first Lemma allows us to draw another conclusion from almost every inclusion of complexity classes; namely the inclusion for their complements.

Lemma 4.9: *Let $C = \mathsf{Q}_1 ... \mathsf{Q}_\omega$ and $D = \mathsf{Q'}_1 ... \mathsf{Q'}_{\omega'}$ be sequences of quantifiers describing complexity classes in the BSS-hierarchy and $C\mathbb{R}, D\mathbb{R}$ their related classes in the discrete real polynomial hierarchy.*

$$\begin{aligned} C\mathbb{R} \subseteq D\mathbb{R} &\Leftrightarrow \mathrm{co}C\mathbb{R} \subseteq \mathrm{co}D\mathbb{R} \\ \mathsf{BP}^0(C) \subseteq \mathsf{BP}^0(D) &\Leftrightarrow \mathrm{co}\mathsf{BP}^0(C) \subseteq \mathrm{co}\mathsf{BP}^0(D) \end{aligned}$$

and

$$\begin{aligned} \mathrm{co}C\mathbb{R} &= \overline{\mathsf{Q}}_1 ... \overline{\mathsf{Q}}_\omega \mathbb{R} \\ \mathrm{co}\mathsf{BP}^0(C) &= \mathsf{BP}^0(\overline{\mathsf{Q}}_1 ... \overline{\mathsf{Q}}_\omega) \end{aligned}$$

with $\overline{\exists} = \forall$, $\overline{\forall} = \exists$, $\overline{\exists^*} = \forall^*$, $\overline{\forall^*} = \exists^*$ *and* $\overline{\mathsf{H}} = \mathsf{H}$.

Proof. The equivalences come form the fact that we can simply form the complement of a problem, apply a polynomial transformation and form the complement of the transformed problem.

Because the complement of a formula is its negation the equalities also hold. ∎

The next result is an extension of a result by Koiran who has formulated the Lemma 4.10 without free variables [Koi99, Lemma 1].

Lemma 4.10: *Let $\phi(y,x)$ be a formula in free variables $x \in \mathbb{R}^n, y \in \mathbb{R}^m$. Let $p_i(y,x) \gtreqless 0$ for $i \in \{1,..,k\}$ be the atoms of ϕ. Let $\textsc{isZero}_i(y)$ be a predicate deciding*

$$\textsc{isZero}_i(y) \quad \Leftrightarrow \quad \forall x : p_i(y,x) = 0, \tag{4.2}$$

i.e. whether $x \mapsto p_i(y,x)$ is the zero polynomial for x.

It holds that the formula

$$\psi(y,x) \quad := \quad \phi(y,x) \wedge \bigwedge_{i=1}^{k} (\neg\textsc{isZero}_i(y) \Rightarrow p_i(y,x) \neq 0)$$

fulfills the equivalence

$$\exists^* x : \phi(y,x) \quad \Leftrightarrow \quad \exists^* x : \psi(y,x) \quad \Leftrightarrow \quad \exists x : \psi(y,x)$$

for all $y \in \mathbb{R}^m$

Proof. The original version by Koiran does not include the $\textsc{isZero}_i(y)$ predicate, but also does not include free variables y. The $\textsc{isZero}_i(y)$ predicate is necessary, because these y variables are coefficients of the polynomials $x \mapsto p_i(y,x)$. If such a polynomial would be the zero polynomial for x, which might depend on y, the simple condition $p_i(y,x) \neq 0$ would be impossible to satisfy. Therefore we preface the condition with a non-zero test.

Now let y be arbitrary but fixed.

"$\Rightarrow$"
If $\exists^* x \in \mathbb{R}^n : \phi(y,x)$ is true then there exists an open ball with center x^* and radius r such that all points $x \in B(x^*,r)$ in that ball satisfy $\phi(y,x)$. W.l.o.g. we can assume that all polynomials $x \mapsto p_i(y,x)$ which fulfill $p_i(y,x^*) \neq 0$ keep their sign on this ball.

If $\neg\textsc{isZero}_i(y,x) \Rightarrow p_i(y,x) \neq 0$ holds for all $i \in \{1,...,m\}$ and $x \in B(x^*,r)$ then we are done.

Now we can assume that for some value $x' \in B(x^*,r)$ and some $i \in \{1,...,m\}$ the non-zero condition does not hold. This implies that $\textsc{isZero}_i(y,x')$ is false, meaning $p_{i,y} : x \mapsto p_i(y,x)$ is not the zero polynomial, and $p_i(y,x') = 0$.

Let $Z(p_{i,y})$ be the zero set of $p_{i,y}$. Define the set $Z_B := Z(p_{i,y}) \cap B$. We know that Z_B cannot be (Euclidean) dense for B or it would contradict Theorem 2.20 which implies that a polynomial with such a dense zero set is the zero polynomial.

Therefore we can find a non-empty open ball $B' \subseteq B$ such that all $x \in B'$ fulfill $p_{i,y}(x) \neq 0$. For all $x \in B'$ the non-zero condition is now fulfilled by one additional polynomial, namely p_i (we know that all other polynomials that already fulfill the condition keep their sign on B and therefore B'). Repeated application of the above arguments shows that:

$$\exists^* x : \phi(y,x) \quad \Rightarrow \quad \exists^* x : \psi(y,x) \quad \Rightarrow \quad \exists x : \psi(y,x) \tag{4.3}$$

"$\Leftarrow$"
Now let x^* fulfill $\psi(y,x^*)$. With Corollary 2.18 we can obtain a ball $b_i = B(x^*,r_i) \subseteq \mathbb{R}^n$ for each polynomial $p_{i,y}$ such that it keeps its signs on this ball. If $p_{i,y}(x^*) \neq 0$ then this follows from the continuity of polynomials and if $p_{i,y}$ is the zero polynomial it keeps its sign everywhere. Because all of these balls are non-empty and have the same center point x^*, their intersection is a non-empty ball $B^* = \bigcap b_i$.

Because the sign of all polynomials stays the same on B^*, we have found a ball that satisfies ψ and therefore also ϕ:

$$\exists^* x : \phi(y,x) \quad \Leftarrow \quad \exists^* x : \psi(y,x) \quad \Leftarrow \quad \exists x : \psi(y,x) \tag{4.4}$$

■

4.3 Inclusions and Equalities

In this section we draw on all previous results to show the relations between the standard and "exotic" complexity classes of the real polynomial hierarchy. These culminate in the robustness results for the zero-th, first and second level and go even beyond.

4.3.1 The Exotic Quantifier H

The exotic quantifier H has somewhat of a special role as it not directly related to either $\exists$ or $\forall$. In many cases H does not seem to have much power at all and can be removed:

Theorem 4.11 ([BC09] [SŠ22]): *Let* $C = \mathsf{Q}_1 \ldots \mathsf{Q}_\omega$, D *be sequences of quantifiers,* $C\mathbb{R}$*and*$D\mathbb{R}$ *their corresponding complexity classes in the real polynomial hierarchy, and* $\mathsf{E} \in \{\exists, \exists^*\}$, $\mathsf{A} \in \{\forall, \forall^*\}$. *Then:*

$$\mathsf{H}\ C\mathbb{R} = C\mathbb{R} \tag{4.5}$$

$$C\mathsf{E}\ \mathsf{H}\ \mathsf{A}D\mathbb{R} = C\mathsf{E}\ \mathsf{A}D\mathbb{R} \tag{4.6}$$

$$C\ \mathsf{H}\mathbb{R} = C\ \mathbb{R} \tag{4.7}$$

Proof of the first equality. We closely follow the proof for $\mathsf{BP}^0(\mathsf{H}C) = \mathsf{BP}^0(C)$ by Bürgisser and Cucker [BC09, Proof of Theorem 9.2].

First note that $C\mathbb{R} \subseteq \mathsf{H}C\mathbb{R}$. So we only have to show $\mathsf{H}C\mathbb{R} \subseteq C\mathbb{R}$.

Let $I_\mathsf{H} = \mathsf{H}\epsilon\ Cx : \phi(\epsilon, x)$ be an instance of Standard($\mathsf{H}C$). Let the length of I_H be L and let the number of quantifiers in C be ω (depending on the algorithm used, it may only be dependent on the number of quantifier alternations in which case ω is an upper bound). First replace all exotic quantifiers in C by their definition containing only non-exotic ones to obtain C'. This at most doubles the number of quantifiers. We can now apply a quantifier elimination algorithm on $C'x : \phi(\epsilon, x)$ which gives us by Theorem 2.4 an equivalent formula $\psi(\epsilon)$ of the form

$$\psi(\epsilon) = \bigvee_{i=1}^{I} \bigwedge_{j=1}^{J_i} (h_{ij}(\epsilon) \gtreqless 0)$$

where I, J_i, the number of atoms, the degree of h_{ij} and the bit length of the integer coefficients are bounded by $L^{\left(2^{O(\omega)} \cdot \prod_{k=1}^{\omega} n_k\right)} \leq L^{\left(2^{O(\omega)} \cdot L \cdot \omega\right)}$. Furthermore, none of the h_{ij} is the zero-polynomial which allows us to define the non-zero polynomial

$$h : \mathbb{R} \to \mathbb{R}\ ;\ h(\epsilon) = \epsilon \cdot \prod_{i=1}^{I} \prod_{j=1}^{J_i} h_{ij}(\epsilon)$$

which fulfills:

$$\{e \mid h(e) = 0\} \quad = \quad \{0\} \cup \bigcup_{i=1}^{I} \bigcup_{j=1}^{J_i} \{e \mid h_{ij}(e) = 0\} \quad \neq \quad \mathbb{R}$$

It is non-zero because the number of zeroes of each h_{ij} is finite and therefore their finite union is also finite. We can now apply Theorem 2.22 to obtain a lower bound on the separation $\operatorname{sep}(h)$ of its roots. This allows us to show the equivalence

$$\mathrm{H}\epsilon : \psi(\epsilon) \quad \Leftrightarrow \quad \psi(\epsilon_0)$$

for $0 < \epsilon_0 < \operatorname{sep}(h)$. This equivalence holds because $h(0) = 0$ and the separation tells us that $h(\epsilon) \neq 0$ for all $\epsilon \in (0, \operatorname{sep}(h))$. This implies that for all $\epsilon \in (0, \operatorname{sep}(h))$ the signs of the polynomials h_{ij} stay the same.

If $\psi(\epsilon_0)$ is true then the separation implies that $\psi(\epsilon)$ is true for all $\epsilon \in (0, \epsilon_0)$. Therefore $\mathrm{H}\epsilon : \psi(\epsilon)$ is true as well.

If $\mathrm{H}\epsilon : \psi(\epsilon)$ is true then we know that there exists an $r > 0$ such that all $0 < \epsilon < r$ fulfill $\psi(\epsilon)$. With the separation this implies that all $\epsilon \in (0, \operatorname{sep}(h))$ fulfill $\psi(\epsilon)$. Therefore $\psi(\epsilon_0)$ is also true.

Next we find a simple bound on $\operatorname{sep}(h)$. Theorem 2.22 gives us the bound

$$\operatorname{sep}(h) > \frac{1}{d_h^{(d_h+2)/2} \cdot \|h\|^{d_h-1}} > \frac{1}{d_h^{(d_h+2)} \cdot \|h\|^{d_h}}$$

with d_h the degree of h and $\|h\|$ the (Euclidean) norm of the coefficient vector of h. We can bound $\|h\| < L \cdot L^{\left(2^{O(\omega)} \cdot L \cdot \omega\right)} = L^{\left(2^{O(\omega)} \cdot L \cdot \omega\right)}$. Introducing the bounds on d_h and $\|h\|$ we obtain

$$\operatorname{sep}(h) > \frac{1}{\left[L^{2^{O(\omega)} L\omega}\right]^{\left[L^{2^{O(\omega)} L\omega}\right]+2} \cdot \left[L^{2^{O(\omega)} \cdot L \cdot \omega}\right]^{\left[L^{2^{O(\omega)} L\omega}\right]}}$$

which is equivalent to

$$\operatorname{sep}(h) > \frac{1}{L^{\left(2^{O(\omega)} \cdot L \cdot \omega \cdot L^{\left(2^{O(\omega)} L\omega\right)} + 2 \cdot 2^{O(\omega)} \cdot L\omega\right)}}$$

We can simplify the bound even further by combining the addition in the exponent and then absorbing the multiplication with 2 into $2^{O(\omega)}$

$$\operatorname{sep}(h) > \frac{1}{L^{\left(2^{O(\omega)} \cdot L\omega \cdot L^{\left(2^{O(\omega)} L\omega\right)}\right)}}$$

Because for a given complexity class $C\mathbb{R}$ the number of quantifiers ω is constant, we can choose a suitable constant $c \in O(2^{O(\omega)})$ which allows us to simplify the bound to:

$$\operatorname{sep}(h) > \frac{1}{L^{c \cdot L^{cL}}} = L^{-c \cdot L^{cL}} > 0$$

We can now construct a formula $\rho(r)$ with additional existentially quantified variables $r \in \mathbb{R}^{cL+1}$ that computes a value $\epsilon_0 \in (0, \operatorname{sep}(h))$:

$$\begin{aligned} \rho \equiv \quad & r_1 \cdot L < 1 && \text{(ensures } r_1 < \tfrac{1}{L}\text{)} \\ \wedge \quad & \textstyle\bigwedge_{i=2}^{(cL+1)} r_{i+1} < r_i^{cL} \\ \wedge \quad & \textstyle\bigwedge_{i=1}^{(cL+1)} r_i > 0 \end{aligned}$$

It holds that $\exists r \in \mathbb{R}^{cL+1} : \rho(r) \Rightarrow r_{cL+1} < \frac{1}{L^{(cL)^{cL}}} < \text{sep}(h)$. The formula ρ is also satisfiable by construction. The size of ρ is polynomial in the size of ϕ. We now obtain the equivalence:

$$\mathsf{H}\epsilon\; Cx : \phi(\epsilon, x) \quad \Leftrightarrow \quad Cx\; \exists r : \phi(r_{cL+1}, x) \wedge \rho(r)$$

If the last quantifier in C is $\exists$ then we can absorb the additional variables from ρ into this last quantifier and obtain an instance of Standard(C). Because ρ only uses strict inequalities we can also quantify its variables with $\exists^*$ instead of $\exists$ (see Theorem 3.10) which can be absorbed if the last quantifier of C is $\exists^*$. If the last quantifier is $\forall$ or $\forall^*$, we can first negate the whole instance then apply our construction and absorb the $\exists$ or $\exists^*$ into the last quantifier and lastly negate the formula again (cf. Lemma 4.9). If the last quantifier is H then we can use the third equality in Theorem 4.11 to remove it. We can apply this equality until another quantifier ($\neq \mathsf{H}$) is the last one or all quantifiers have been removed. This repeated application is possible in polynomial time because we treat ω as constant. ∎

Proof of the second equality. By Lemma 2.15 we can replace $\mathsf{H}\epsilon$ by

$$\mathsf{E}r > 0\; \mathsf{A}\epsilon \in (0, r)$$

for $\mathsf{E} \in \{\exists, \exists^*\}$ and $\mathsf{A} \in \{\forall, \forall^*\}$. In the case EHA these quantifiers can by absorbed by the preceeding and succeeding ones.

Corollary 2.16 gives us the other case. ∎

Proof of the third equality. For any quantifier-free formula $\phi(\epsilon)$ we can, by Theorem 2.3, apply quantifier elimination on the expansion of $\mathsf{H}\epsilon : \phi(\epsilon)$ twice in polynomial time, yielding an equivalent formula of polynomial size without H [SŠ22, Lemma 3.6 and Corollary 3.7]. ∎

Remark 4.12: *In the proof of the first equality we had to introduce additional variables to compute a bound for sep(h). In the BSS-hierarchy this can be achieved without introducing additional variables by encoding the bound into an algebraic circuit. In the BSS-model one can even leverage the constant time and space operations for real numbers, to simply compute this bound in polynomial time (and space)! This is done in the proof of* $\mathsf{BP}^0(\mathsf{H}C) \subseteq \mathsf{BP}^0(C)$ *by Bürgisser and Cucker [BC09, Proof of Theorem 9.2]. The same is not possible in the discrete Turing model.*

With the results above we may conjecture that the H quantifier can be always eliminated. However, to the best of my knowledge, this has not yet been proven. We also weren't able to eliminate H in Theorem 4.6.

To this end we would require a result which allows the elimination between two similar quantifiers, i.e. $\exists\mathsf{H}\exists$. The problem in these cases is that the H quantifier includes an implicit *quantifier alternation.* This topic should be investigated further in the future.

4.3.2 First Level of the Hierarchy

For the first level of the real polynomial hierarchy, the following equalities are known:

Theorem 4.13 ([BC09, Corollary 9.3]):

$$\exists^*\mathbb{R} = \exists\mathbb{R} \quad = \quad \mathsf{BP}^0(\exists) = \mathsf{BP}^0(\exists^*)$$

and

$$\forall^*\mathbb{R} = \exists\mathbb{R} \quad = \quad \mathsf{BP}^0(\forall) = \mathsf{BP}^0(\forall^*)$$

Proof.

$$\begin{aligned}
\exists^*\mathbb{R} &\subseteq \mathsf{BP}^0(\exists^*) && \text{Lemma 4.4}\\
\mathsf{BP}^0(\exists^*) &= \mathsf{BP}^0(\exists) && \text{[BC09, Corollary 9.3]}\\
\mathsf{BP}^0(\exists) &= \exists\mathbb{R} && \text{Lemma 4.4}\\
\exists\mathbb{R} &\subseteq \exists^*\mathbb{R} && \text{Corollary 3.11}
\end{aligned}$$

The analogous equalities for the universal quantifier follow from Lemma 4.9. ∎

On the first level of the discrete real polynomial hierarchy, the two exotic quantifiers $\exists^*$ and $\forall^*$ are computationally equivalent to their non-exotic counterparts. Theorem 4.11 shows that a single H quantifier can be removed and therefore $\mathsf{H}\mathbb{R} = \mathsf{P}$. This shows that the real polynomial hierarchy is robust on the first level.

4.3.3 Second Level of the Hierarchy

Instead of only showing results for the second level of the hierarchy, Bürgisser and Cucker have shown an even stronger result:

$$\exists^* C \subseteq \exists C$$

with C being a sequence of quantifiers. Their result is based on work by Koiran, which we introduce now.

Koiran Let $F(u, v)$ be any formula in the first-order theory of the reals with free variables $u \in \mathbb{R}^s$ and $v \in \mathbb{R}^k$. Define the formula $\tilde{F}(u, y_1, \ldots, y_{k+s+2})$ as

$$\exists x \in \mathbb{R}^k\ \exists \epsilon > 0 : \bigwedge_{i=1}^{k+s+2} F(u, x + \epsilon y_i)$$

where each variable $y_i \in \mathbb{R}^k$.

We can now define the set of *witness sequences* $W(F)$ as the set of all $y = (y_1, \ldots, y_{k+s+2}) \in \mathbb{R}^{k(k+s+2)}$ that fulfill

$$\forall u \in \mathbb{R}^s \quad : \quad \left[\tilde{F}(u, y_1, \ldots, y_{k+s+2}) \iff \exists^* v : F(u, v)\right]. \tag{4.8}$$

Theorem 4.14 ([Koi99, Theorem 2])**:** *For any formula F in the first-order theory of the reals $W(F)$ is dense in $\mathbb{R}^{k(k+s+2)}$.*

Theorem 4.15 ([Koi99, Theorem 3])**:** *Let $F(u, v)$ be a prenex formula in the first-order theory of the reals with free variables $u \in \mathbb{R}^s$ and $v \in \mathbb{R}^k$. Let n denote the total number of bound variables, ω the number of (distinct) quantifier blocks, m the number of atoms, D the maximum degree of all atoms and L the largest bit length of any integer coefficient.*

One can construct an integer point y in $W(F)$ in $O(\log L) + (s + k + n)^{O(\omega)} \log(mD)$ arithmetic operations.

To illuminate these probably somewhat unintuitive results we, use an example. Consider the formula

$$F(v) :\equiv p(v) = 0$$

with $v \in \mathbb{R}^k$ and no parameter variables u, i.e. $s = 0$. We are now interested in the sentence

$$\exists^* v : F(v)$$

which we would like to transform so it only contains non-exotic quantifiers. Because of the specific choice of F, Corollary 2.21 already provides us with an equivalent formula that does not contain $\exists^*$:

$$\exists v_1 \dots \exists v_{(\Delta+1)^k} : \bigwedge_{i,j} v_i \neq v_j \ \wedge \ \bigwedge_i p(v_i) = 0$$

where Δ is the degree of p. This formula is, however, no longer of polynomial size in the dimension of v because we introduce $(\Delta + 1)^k$ variables.

The result of Koiran above states that we can still construct a polynomial size formula. Instead of testing $p(v) = 0$ at $(\Delta + 1)^k$ many points it is enough to test at $k + s + 2$ many, i.e. polynomially many, specific points. These points are dependent on the formula F and the free variables v and are thus written as shifts $y_1, \dots, y_{k+2}$ ($s = 0$). Taken together these shifts are the vector $y \in \mathbb{R}^{k(k+2)}$.

Theorem 4.14 states that the set of all such possible y is dense in $\mathbb{R}^{k(k+2)}$ which allows an easy computation of at least some of these y. This is given in Theorem 4.15.

To account for arbitrarily small open balls on which F is fulfilled, an additional ϵ variable is introduced which scales the shifts $y_1, \dots, y_{k+2}$ to arbitrary small, but non-zero, values.

Together we obtain an equivalent formula for $\exists^* v : F(v)$ as

$$\exists v \, \exists \epsilon > 0 : \bigwedge_{i=1}^{k+2} p(v + \epsilon y_i).$$

Remark 4.16: *It is very important to note that this formula cannot necessarily be constructed in polynomial time. Theorem 4.15 only provides a calculation containing polynomial many steps for the shifts y which is an important distinction. However, Lemma 4.5 allows us to encode the calculation into an algebraic circuit which we can then transform with Theorem 2.8 into a formula by introducing additional existentially quantified variables. In our case this is no problem as we can simply absorb these variables into the $\exists$ quantifier already present.*

Remark 4.17: *Because any point $y \in W(F)$ fulfills Equation* (4.8) *the computation of y has to be independent of u (and also of v). Curiously, this observation has not been used or even been noted by Bürgisser and Cucker [BC09] who could have strengthened their result to $C\exists^* D \subseteq C\exists D$ as we do next.*

Remark 4.18: *We use the above results by Koiran as a black box and trust their validity. However, the author is not entirely convinced that the results are correct or have been interpreted correctly and urges the reader to consider the above citations and following proof with care.*

Theorem 4.19: *Let C and D be sequences of quantifiers. Then:*

$$C\exists^* D \subseteq C\exists D$$
$$\mathrm{BP}^0(C\exists^* D) \subseteq \mathrm{BP}^0(C\exists D)$$
$$C\exists^* D\mathbb{R} \subseteq C\exists D\mathbb{R}$$

Proof. We mostly follow the proof of Bügisser and Cucker [BC09, Theorem 8.2] except for the parts where we refer to Remark 4.17.

In all cases, let $I = Cx\ \exists^* v\ Dz : \phi(u,x,v,z)$ be an instance of the standard problem of the included complexity class (in the BSS-hierarchy $\phi(u,x,v,z)$ is the equality $C(u,x,v,z) = 1$ where C is an algebraic decision circuit). This formula may contain free variables $u \in \mathbb{R}^{n_u}$ which stand for the constants in the BSS-model interpretation. In the discrete classes $n_u = 0$. We denote with n_u, n_x, n_v, n_z the dimension of u, x, v, z respectively.

Define $F(u,x,v) := Dz : \phi(u,x,v,z)$ which is clearly a prenex formula in the first-order theory of the reals. Theorem 4.15 allows us to compute a sequence $y = (y_1, \ldots, y_{n_u+n_x+n_v+2}) \in W(F)$ in polynomial many arithmetic operations. From Remark 4.16 we obtain a formula $\psi(c,y)$ such that for all y

$$\exists c : \psi(c,y) \quad \Rightarrow \quad y \in W(F).$$

From Equation (4.8) we obtain for all $(u,x) \in \mathbb{R}^{n_u+n_x}$ the equivalence

$$\exists v\ \exists \epsilon > 0\ \exists (c,y) : \psi(c,y) \wedge \bigwedge_{i=1}^{n_u+n_x+n_v+2} F(u,x,v+\epsilon y_i) \quad \Leftrightarrow \quad \exists^* v : F(u,x,v).$$

The formula ψ has polynomial size by construction and therefore the whole formula

$$\psi(c,y) \wedge \bigwedge_{i=1}^{n_u+n_x+n_v+2} F(u,x,v+\epsilon y_i)$$

is of polynomial size. We can push the quantifiers inside F outside of the conjunction to obtain an instance J of the enclosing complexity class:

$$J \quad = \quad Cx\ \exists(v,\epsilon,c,y)\ D'z' : \psi(c,y) \wedge \bigwedge_{i=1}^{n_u+n_x+n_v+2} \phi(u,x,v+\epsilon y_i, z_i)$$

where in D' every quantifier block of D is extended by the factor $n_u + n_x + n_v + 2$ and therefore $z' = (z_1, \ldots, z_{n_u+n_x+n_v+2}) \in \mathbb{R}^{n_z(n_u+n_x+n_v+2)}$. The sequence D' contains all quantifiers from D for every part of the conjunction independently. This formula is still of polynomial size.

We have now shown that we can transform any instance I of the standard problem for the included complexity class into an instance J of the standard problem for the enclosing complexity class which concludes the proof. ∎

While we can, by definition, always replace an $\exists^*$ quantifier by $\exists\forall$, we now show that an $\exists^*$ quantifier at the end of a quantifier sequence can be replaced by $\forall\exists$.

Theorem 4.20: *Let C be a (possibly empty) sequence of quantifiers. Then it holds that:*

$$C\exists^*\mathbb{R} \subseteq C\forall\exists\mathbb{R}$$

Proof. Consider Standard($C\exists^*$) defined as deciding whether the following sentence is true:

$$Cy \in \mathbb{R}^m\ \exists^* x \in \mathbb{R}^n : \phi(y,x)$$

with ϕ being a quantifier-free formula as defined in Definition 2.1. Let $p_i(y,x) \gtreqless 0$ for $i \in \{1, \ldots, k\}$ be the atoms of ϕ. Let $y \in \mathbb{R}^m$ be arbitrary but fixed.

We can obtain by Lemma 4.10 a formula

$$\psi(y,x) \quad :\equiv \quad \phi(y,x) \wedge \bigwedge_{i=1}^{k} (\neg \textsc{isZero}_i(y) \Rightarrow p_i(y,x) \neq 0)$$

such that

$$\exists^* x : \phi(y,x) \quad \Leftrightarrow \quad \exists x : \psi(y,x)$$

holds. However the constructed formula $\psi(y,x)$ contains the predicate $\textsc{isZero}$ which would have to introduce additional $\forall$ quantifiers or not be of polynomial length (see Theorem 2.20 and Corollary 2.21). Converting this formula into prenex form would position these $\forall$ quantifiers after the $\exists$ quantifier and therefore showing

$$C\exists^*\mathbb{R} \subseteq C\exists\forall\mathbb{R}$$

which is neither wrong nor helpful. We have already derived the same result from the definition of $\exists^*$.

But in this special case, we can construct an equivalent formula in polynomial time, by introducing an additional $\forall$ quantifier before the $\exists$ quantifier instead of after. In essence we are showing that the $\exists$ and $\forall$ quantifiers in this case are independent and can be swapped. We define

$$\psi'(y,z,x) \quad :\equiv \quad \phi(y,x) \wedge \bigwedge_{i=1}^{k} (p_i(y,z) \neq 0 \Rightarrow p_i(y,x) \neq 0) \tag{4.9}$$

and propose that (y is still arbitrary but fixed from above):

$$\exists x : \psi(y,x) \quad \Leftrightarrow \quad \forall z\, \exists x : \psi'(y,z,x) \tag{4.10}$$

If $\exists x : \psi(y,x)$ is true then there exists x^* such that $\phi(y,x^*)$ and $p_i(y,x^*) \neq 0 \vee \textsc{isZero}_i(y)$ for all $i \in \{1,..,k\}$. This implies that

$$\forall z : \psi'(y,z,x^*)$$

because any polynomial $p_i(y,x)$ is either the zero polynomial for x, fulfilling $\forall z : p_i(y,z) = 0$, or $p_i(y,x^*) \neq 0$. Therefore,

$$\exists x : \psi(y,x) \quad \Rightarrow \quad \forall z\, \exists x : \psi'(y,z,x). \tag{4.11}$$

Now assume $\exists x : \psi(y,x)$ to be false. Then for every value x^* which satisfies $\phi(y,x)$ there exists a polynomial $p_{i(x^*)}(y,x^*) = 0 \wedge \neg\textsc{isZero}_{i(x^*)}(y)$. But because the polynomial is not the zero polynomial for x there exists $z'(x^*)$ such that $p_i(y,z'(x^*)) \neq 0$. Therefore,

$$\exists x : \psi'(y,z'(x),x)$$

is false, which in turn implies that

$$\forall z\, \exists x : \psi'(y,z,x)$$

is also false. Therefore,

$$\neg\exists x : \psi(y,x) \quad \Rightarrow \quad \neg\forall z\, \exists x : \psi'(y,z,x). \tag{4.12}$$

Equation (4.11) and Equation (4.12) prove Equation (4.10) and with Lemma 4.10 it follows that

$$\exists^* x : \phi(x) \quad \Leftrightarrow \quad \forall z\, \exists x : \psi'(y,z,x)$$

for all $y \in \mathbb{R}^m$. The formula $\psi'(y,x,z)$ can be constructed in polynomial time which proves the theorem. ∎

Remark 4.21: *The result of Theorem 4.20 is in fact a weaker version of Theorem 4.19 for $D = \emptyset$. We still include it as its proof does not rely as strongly on results that the author has to take on trust. The proof also shows a different approach to a similar problem and might help to improve the readers intuition of these problems.*

We now look at inclusions in the other direction, i.e. cases in which the $\exists$ quantifier can be replaced by an $\exists^*$ quantifier.

Theorem 4.22 ([BC09, Proposition 8.4])**:** *Let C be a sequence of quantifiers. Then:*

$$C\exists\mathbb{R} \subseteq C\mathsf{H}\mathsf{H}\exists^*\mathbb{R}$$

Proof. Let

$$Cy \in \mathbb{R}^m\ \exists x \in \mathbb{R}^{n_0} : \phi(y,x)$$

be an instance of Standard($C\exists$). With Lemma 3.7 we can obtain an equivalent modified instance

$$Cy \in \mathbb{R}^m\ \exists x \in \mathbb{R}^n : p(y,x) = 0$$

by introducing additional existentially quantified variables ($n \geq n_0$).

Let $y \in \mathbb{R}^m$ now be arbitrary but fixed. It then holds that:

$$\exists x \in \mathbb{R}^n : p(y,x) = 0 \quad \Leftrightarrow \quad \mathsf{H}\delta\ \exists x \in \mathbb{R}^n : (\|x\|^2 \leq \delta^{-1} \wedge p(y,x) = 0)$$

For x^* such that $p(y,x^*) = 0$ we can choose $\delta \leq \|x^*\|^{-2}$.

Because polynomials are continuous and the ball $B[0,\delta^{-1}]$ is compact, p attains its minimum on the ball (see Lemma 2.19). Therefore we can choose an ϵ smaller than this minimum and we obtain an equivalent statement:

$$\mathsf{H}\delta\ \mathsf{H}\epsilon\ \exists x \in \mathbb{R}^n : (\|x\|^2 \leq \delta^{-1} \wedge p(y,x)^2 < \epsilon)$$

The order of these H quantifiers is determined by the fact that there exist multivariate polynomials whose image contains a convergent subsequence converging to 0 but that do not have a zero (i.e. $(x,y) \mapsto y^2 + (1-xy)$). However Lemma 2.19 tells us that this can only happen for $\|x\| \to \infty$. By therefore bounding $\|x\|$ first ($\mathsf{H}\delta$) and then requiring convergence to 0 ($\mathsf{H}\epsilon$) we indeed test for a zero (and not only convergence).

For all given values of ϵ we can obtain with Corollary 2.18 an open ball around a given zero of p. This allows us to replace the $\exists$ quantifier with an $\exists^*$ quantifier:

$$\mathsf{H}\delta\ \mathsf{H}\epsilon\ \exists^* x \in \mathbb{R}^n : (\|x\|^2 \leq \delta^{-1} \wedge p(y,x)^2 < \epsilon)$$

This concludes the proof. ∎

Schaefer and Štefankovič found an even stronger result, even if it may not look like it on first glance.

Theorem 4.23 ([SŠ22, Theorem 2.4])**:** *Let $p(y,x) \in \{0,1\}[Y_1,\dots,Y_m,X_1,\dots,X_n]$ a polynomial of degree at most Δ. Then for arbitrary but fixed $y \in \mathbb{R}^m$, the sentence:*

$$\exists x \in \mathbb{R}^n : p(y,x) = 0$$

is equivalent to

$$\forall r > 0\ \exists \delta \in (0,r)\ \exists x \in \mathbb{R}^n : \left(\|p(y,x)\| < \delta^C \wedge \bigwedge_{i=1}^{n} \|\delta \cdot x_i\|^2 < 1 \right)$$

where $C = n \cdot \Delta^{n^c} + (\Delta+1) \cdot n$ and c is a universal constant.

Corollary 4.24: *For arbitrary but fixed $y \in \mathbb{R}^m$, the sentence*

$$\exists x \in \mathbb{R}^n : p(y,x) = 0$$

is equivalent to

$$\mathrm{H}\delta\ \exists x \in \mathbb{R}^n : \left(\|p(y,x)\| < \delta^C \wedge \bigwedge_{i=1}^{n} \|\delta \cdot x_i\|^2 < 1 \right)$$

.

Proof. We can replace $\forall r > 0\ \exists \delta \in (0,r)$ by $\mathrm{H}\delta$. This is shown in Corollary 2.16. ∎

Theorem 4.25: *Let D be a (possibly empty) sequence of quantifiers. Then:*

$$D\exists\mathbb{R} \subseteq D\mathrm{H}\exists^*\mathbb{R}$$

Proof. Let

$$Dy \in \mathbb{R}^m\ \exists x \in \mathbb{R}^{n_0} : \phi(y,x)$$

be an instance of Standard($D\exists$). With Lemma 3.7 we can obtain an equivalent modified instance

$$Dy \in \mathbb{R}^m\ \exists x \in \mathbb{R}^n : p(y,x) = 0$$

by introducing additional existentially quantified variables ($n \geq n_0$).

By introducing even more existentially quantified variables we can construct a formula ρ which calculates a value $0 < d_{n^c+1} < \delta^C$ in polynomial time. First we decompose

$$\delta^C = \delta^{n \cdot \Delta^{n^c} + (\Delta+1)\cdot n} = \delta^{n\Delta^{n^c}} \cdot \delta^{(\Delta+1)n}.$$

We can write $\delta^{(\Delta+1)n}$ in polynomial time and space $O(\Delta n)$. By introducing polynomially many variables $d_i > 0$ for $i \in \{1, \dots, n^c\}$ and a formula

$$d_1 < \delta^\Delta \quad \wedge \quad \bigwedge_{i=2}^{n^c} d_i < d_{i-1}^\Delta$$

we obtain $d_{n^c} < \delta^{\Delta^{n^c}}$. We introduce another variable $d_{n^c+1} > 0$ and formula

$$d_{n^c+1} < d_{n^c}^n \quad \cdot \quad \delta^{n(\Delta+1)}$$

which ensures

$$d_{n^c+1} < d_{n^c}^n \cdot \delta^{n(\Delta+1)} \quad < \quad \left(\delta^{\Delta^{n^c}}\right)^n \cdot \delta^{n(\Delta+1)} \quad = \quad \delta^{n\Delta^{n^c} + n(\Delta+1)} \quad = \quad \delta^C.$$

We combine the construction of d_{n^c+1} into a single formula and call it $\rho(\delta, d)$.

Because ρ is independent of y we can apply Corollary 4.24 to obtain that for all $y \in \mathbb{R}^m$

$$\exists x \in \mathbb{R} : p(y,x) = 0$$

is equivalent to

$$\mathrm{H}\delta\ \exists x \in \mathbb{R}^n\ \exists d \in \mathbb{R}^{n^c+1} : \left(\|p(y,x)\| < d_{n^c+1} \wedge \bigwedge_{i=1}^{n} \|\delta \cdot x_i\| < 1 \wedge \rho(\delta, d) \right).$$

The formula

$$\left(\|p(y,x)\| < d_{n^c+1} \wedge \bigwedge_{i=1}^{n} \|\delta \cdot x_i\| < 1 \wedge \rho(\delta, d)\right)$$

contains only strict inequalities and no negations. The semialgebraic set described by it for given y, δ is therefore open (cf. Theorem 3.10). This allows us to replace the last $\exists$ quantifier with an $\exists^*$ quantifier to obtain an instance of STANDARD(DH$\exists^*$). ∎

We now focus specifically on the second level of the real polynomial hierarchy. We look at all complexity class that consist of two different quantifiers.

Theorem 4.26:

$$\exists\exists^*\mathbb{R} = \exists^*\exists\mathbb{R} = \exists\exists\mathbb{R} = \exists\mathbb{R} = \exists^*\mathbb{R} = \exists^*\exists^*\mathbb{R}$$

$$\forall\forall^*\mathbb{R} = \forall^*\forall\mathbb{R} = \forall\forall\mathbb{R} = \forall\mathbb{R} = \forall^*\mathbb{R} = \forall^*\forall^*\mathbb{R}$$

Proof.

$$\begin{array}{rll} \exists^*\exists\mathbb{R} & \subseteq \exists\exists\mathbb{R} = \exists\mathbb{R} & \text{Theorem 4.19} \\ \exists\mathbb{R} & \subseteq \exists^*\exists\mathbb{R} & \text{by definition} \\ \exists\exists^*\mathbb{R} & \subseteq \exists\exists\mathbb{R} = \exists\mathbb{R} & \text{Theorem 4.19} \\ \exists\mathbb{R} & \subseteq \exists\exists^*\mathbb{R} & \text{by definition} \\ \exists\mathbb{R} & = \exists^*\mathbb{R} & \text{Theorem 4.13} \\ \exists^*\mathbb{R} & = \exists^*\exists^*\mathbb{R} & \text{by definition} \end{array}$$

The analogous equalities for the universal quantifiers follow from Lemma 4.9. ∎

On the second level of the discrete polynomial hierarchy, the $\exists^*$ and $\exists$ ($\forall^*$ and $\forall$) quantifiers collapse together and reduce to the first level.

Theorem 4.27:

$$\exists^*\forall^*\mathbb{R} = \exists\forall^*\mathbb{R} = \exists\forall\mathbb{R} = \exists^*\forall\mathbb{R}$$

$$\forall^*\exists^*\mathbb{R} = \forall\exists^*\mathbb{R} = \forall\exists\mathbb{R} = \forall^*\exists\mathbb{R}$$

Proof.

$$\begin{array}{rlr} \exists^*\forall^*\mathbb{R} & \subseteq \exists\forall^*\mathbb{R} & \text{Theorem 4.19} \\ \exists\forall^*\mathbb{R} & \subseteq \exists\exists\forall\mathbb{R} = \exists\forall\mathbb{R} & \text{Theorem 4.20} \\ \exists\forall\mathbb{R} & = \exists^*\forall\mathbb{R} & \text{[SŠ22, Corollary 3.10]} \\ \exists^*\forall\mathbb{R} & \subseteq \exists^*\mathsf{H}\forall^*\mathbb{R} = \exists^*\forall^*\mathbb{R} & \text{Theorem 4.25 and Theorem 4.11} \end{array}$$

The other part follows again from Lemma 4.9. ∎

We can also derive inclusion for combinations with the H quantifier which we note in the next corollary.

Corollary 4.28 (of Theorem 4.11)**:**

$$\begin{array}{rcccl} \mathsf{H}\exists\mathbb{R} & = & \exists\mathbb{R} & = & \exists\mathsf{H}\mathbb{R} \\ \mathsf{H}\exists^*\mathbb{R} & = & \exists^*\mathbb{R} & = & \exists^*\mathsf{H}\mathbb{R} \\ \mathsf{H}\forall\mathbb{R} & = & \forall\mathbb{R} & = & \forall\mathsf{H}\mathbb{R} \\ \mathsf{H}\forall^*\mathbb{R} & = & \forall^*\mathbb{R} & = & \forall^*\mathsf{H}\mathbb{R} \\ \mathsf{H}\mathsf{H}\mathbb{R} & = & \mathsf{P} & & \end{array}$$

We now have shown the *robustness* of the discrete real polynomial hierarchy under the exotic quantifiers for the first two levels. This means that in this hierarchy the exotic quantifiers H, $\exists^*$ and $\forall^*$ are computationally equivalent (or, in the case of H weaker) to their non-exotic counterparts $\exists$ and $\forall$. We show these results in Figure 4.2.

$$\exists\forall\mathbb{R} \qquad \leftarrow \text{co} \rightarrow \qquad \forall\exists\mathbb{R}$$

$$\exists^*\forall\mathbb{R} \;\; \exists\forall^*\mathbb{R} \;\; \exists^*\forall^*\mathbb{R} \qquad\qquad \forall^*\exists\mathbb{R} \;\; \forall\exists^*\mathbb{R} \;\; \forall^*\exists^*\mathbb{R}$$

$$\cup\!\!\mathsf{I}$$

$$\exists\mathbb{R} \qquad \leftarrow \text{co} \rightarrow \qquad \forall\mathbb{R}$$

$$\exists^*\exists\mathbb{R} \;\; \exists\exists^*\mathbb{R} \;\; \exists^*\mathbb{R} \qquad\qquad \forall^*\forall\mathbb{R} \;\; \forall\forall^*\mathbb{R} \;\; \forall^*\mathbb{R}$$

$$\mathsf{H}\exists\mathbb{R} \;\; \mathsf{H}\exists^*\mathbb{R} \;\; \exists\mathsf{H}\mathbb{R} \;\; \exists^*\mathsf{H}\mathbb{R} \qquad\qquad \mathsf{H}\forall\mathbb{R} \;\; \mathsf{H}\forall^*\mathbb{R} \;\; \forall\mathsf{H}\mathbb{R} \;\; \forall^*\mathsf{H}\mathbb{R}$$

$$\cup\!\!\mathsf{I}$$

$$\mathsf{P}$$

$$\mathsf{H}\mathsf{H}\mathbb{R} \;\; \mathsf{H}\mathbb{R}$$

Figure 4.2: Distinct complexity classes in the discrete real polynomial hierarchy. Boxes represent the different levels of the hierarchy and all classes written closely together are computationally equivalent.

5 Conclusion and Future Work

In Section 4.1 we define the (discrete) real polynomial hierarchy. We introduce the exotic quantifiers $\exists^*$, $\forall^*$ and H. We shed light onto two (potentially) different definitions of this hierarchy and discuss in Section 4.1.1 the difficulties in unifying these definitions. The difficulties lie primarily in varying perceived encoding complexity of algebraic circuits and quantifier-free formulas. While we could not show any equivalence or separation, we provide an intermediary result that may help to close these differences in the future.

From Section 4.2 onward we show the robustness of the real polynomial hierarchy under the exotic quantifiers. Figure 4.2 contains these robustness results up to the second level of the hierarchy.

Some results are more general and go beyond the second level. These include Theorem 4.19 ($C\exists^*D\mathbb{R} \subseteq C\exists D\mathbb{R}$), Theorem 4.25 ($C\exists\mathbb{R} \subseteq C\mathsf{H}\exists^*$) and Theorem 4.11 ($\mathsf{H}C\mathbb{R} = C\mathsf{H}\mathbb{R} = C\mathbb{R}$ and $\mathsf{AHE} = \mathsf{AE}$). To show robustness for the third and higher levels, only a few additional results would be necessary. Among them are cases in which the H quantifier appears between two similar quantifiers, e.g. $\exists\mathsf{H}\exists$ and the other direction of Theorem 4.19, i.e. $C\exists D\mathbb{R} \subseteq C\exists^*D\mathbb{R}$.

The Complexity Class $\exists\mathbb{R} \cap \forall\mathbb{R}$ We can also answer a question posed by Schaefer and Štefankovič in the conclusion to their paper [SŠ22]: What interesting problems exist in the intersection complexity class of $\exists\mathbb{R} \cap \forall\mathbb{R}$? With the results from this thesis we can easily show that the well-studied problem PIT (polynomial identity testing) is contained in this class.

PIT is defined as testing whether two polynomials, given in arbitrary form, are identical. This is equivalent to asking whether their subtraction polynomial is the zero polynomial. For a given polynomial p, Corollary 2.21 gives us two equivalent statements

$$\forall x \in \mathbb{R}^n : p(x) = 0$$

and

$$\exists^* x \in \mathbb{R}^n : p(x) = 0.$$

These problems clearly lie in $\forall\mathbb{R}$ and $\exists\mathbb{R}$ respectively (even when p is given as an algebraic circuit, see Theorem 4.13). Therefore PIT $\in \forall\mathbb{R} \cap \exists\mathbb{R}$.

Due to this, it might be interesting to investigate this complexity class further and find complete problems. Currently, only very rough computation bounds are known (derived from Theorem 3.5): $\mathsf{P} \subseteq (\mathsf{NP} \cap \mathsf{coNP}) \subseteq (\exists\mathbb{R} \cap \forall\mathbb{R}) \subseteq \mathsf{PSPACE}$. Especially its relation to NP (and coNP) may provide new insights.

Bibliography

[AKBM06] E. Allender, J. Kjeldgaard-Pedersen, P. Burgisser, and P. Miltersen. “On the complexity of numerical analysis”. In: *21st Annual IEEE Conference on Computational Complexity (CCC’06)*. 2006, 9 pp.–339. DOI: *10.1109/CCC.2006.30*.

[Alo99] Noga Alon. “Combinatorial Nullstellensatz”. In: *Combinatorics, Probability and Computing* Volume 8 (1999), pp. 7–29. DOI: *10.1017/S0963548398003411*.

[BC09] Peter Bürgisser and Felipe Cucker. “Exotic Quantifiers, Complexity Classes, and Complete Problems”. In: *Foundations of Computational Mathematics* Volume 9 (2009), pp. 135–170. ISSN: 1615-3383. DOI: *10.1007/s10208-007-9006-9*.

[Bjö+99] Anders Björner, Michel Las Vergnas, Bernd Sturmfels, Neil White, and Gunter M. Ziegler. *Oriented Matroids*. 2nd ed. Cambridge University Press, 1999. DOI: *10.1017/CBO9780511586507*.

[BPR03] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in Real Algebraic Geometry (Algorithms and Computation in Mathematics)*. Berlin, Heidelberg: Springer-Verlag, 2003. ISBN: 978-3-662-05355-3.

[BPR16] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in Real Algebraic Geometry (Algorithms and Computation in Mathematics, Volume 10)*. Berlin, Heidelberg: Springer-Verlag, 2016.

[BPR96] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. “On the Combinatorial and Algebraic Complexity of Quantifier Elimination”. In: *J. ACM* Volume 43 (Nov. 1996), pp. 1002–1045. ISSN: 0004-5411. DOI: *10.1145/235809.235813*.

[BV07] Saugata Basu and Nicolai Vorobjov. *On the number of homotopy types of fibres of a definable map*. 2007. arXiv: *math/0605517*.

[Can88] John Canny. “Some Algebraic and Geometric Computations in PSPACE”. In: *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*. Chicago, Illinois, USA: Association for Computing Machinery, 1988, pp. 460–467. ISBN: 0897912640. DOI: *10.1145/62212.62257*.

[DH88] James H. Davenport and Joos Heintz. “Real quantifier elimination is doubly exponential”. In: *Journal of Symbolic Computation* Volume 5 (1988), pp. 29–35. ISSN: 0747-7171. DOI: *https://doi.org/10.1016/S0747-7171(88)80004-X*.

[DKMR21] Michael G. Dobbins, Linda Kleist, Tillmann Miltzow, and Paweł Rzążewski. *Completeness for the Complexity Class $\forall\exists\mathbb{R}$ and Area-Universality*. 2021. arXiv: *1712.05142*.

[EHM21] Jeff Erickson, Ivor van der Hoog, and Tillmann Miltzow. *Smoothing the gap between NP and ER*. 2021. arXiv: *1912.02278*.

[JKM22] Paul Jungeblut, Linda Kleist, and Tillmann Miltzow. *The Complexity of the Hausdorff Distance*. 2022. arXiv: *2112.04343*.

[Koi99] Pascal Koiran. “The Real Dimension Problem Is NPR-Complete”. In: *Journal of Complexity* Volume 15 (1999), pp. 227–238. ISSN: 0885-064X. DOI: *https://doi.org/10.1006/jcom.1999.0502*.

[Mat14] Jiří Matoušek. “Intersection graphs of segments and $\exists\mathbb{R}$”. In: *CoRR* Volume abs/1406.2636 (2014). arXiv: *1406.2636*.

[Mig92] Maurice Mignotte. *Mathematics for Computer Algebra*. New York: Springer-Verlag, 1992. ISBN: 978-1-4613-9171-5. DOI: *10.1007/978-1-4613-9171-5*.

[Ren92] James Renegar. “On the computational complexity and geometry of the first-order theory of the reals. Part III: Quantifier elimination”. In: *Journal of Symbolic Computation* Volume 13 (1992), pp. 329–352. ISSN: 0747-7171. DOI: *https://doi.org/10.1016/S0747-7171(10)80005-7*.

[Son85] Eduardo D. Sontag. “Real addition and the polynomial hierarchy”. In: *Information Processing Letters* Volume 20 (1985), pp. 115–120. ISSN: 0020-0190. DOI: *https://doi.org/10.1016/0020-0190(85)90076-6*.

[SŠ17] Marcus Schaefer and Daniel Štefankovič. “Fixed Points, Nash Equilibria, and the Existential Theory of the Reals”. In: *Theory of Computing Systems* Volume 60 (2017), pp. 172–193. ISSN: 1433-0490. DOI: *10.1007/s00224-015-9662-0*.

[SŠ22] Marcus Schaefer and Daniel Štefankovič. *Beyond the Existential Theory of the Reals*. 2022. arXiv: *2210.00571*.

[Tar51] A. Tarski. *A Decision Method for Elementary Algebra and Geometry*. University of California Press, 1951. ISBN: 9780598987778.

[Wika] Wikipedia. “Analytical hierarchy”. URL: *https://en.wikipedia.org/wiki/Analytical_hierarchy*.

[Wikb] Wikipedia. “Arithmetical hierarchy”. URL: *https://en.wikipedia.org/wiki/Arithmetical_hierarchy*.

[Wikc] Wikipedia. “Existential theory of the reals”. URL: *https://en.wikipedia.org/wiki/Existential_theory_of_the_reals*.

[Wikd] Wikipedia. “Polynomial hierarchy”. URL: *https://en.wikipedia.org/wiki/Polynomial_hierarchy*.