Freie Universität Berlin
Kaiserswerther Straße 16-18
14195 Berlin

Sommersemester 2019

Some Aspects of Real Quantifier Elimination

Verfasser:	Tobias Stamm
Betreuer:	Prof. Dr. Rainer Sinn
Ausgabe des Themas:	13.6.2019
Abgabe der Masterarbeit:	21.11.2019

Contents

1 Introduction

The first order language of the reals is one of the most expressive theories that is also decidable. Because of this it has been applied in a diverse set of applications, despite the high computational complexity of its problems. Among these is the quantifier elimination problem of which some aspects are covered in this thesis. In principle this is fascinating because it represents a geometric form of formal reasoning.

This thesis starts out by presenting the mathematical groundwork used in this thesis and some real algebraic geometry to then introduce the notion of quantifier elimination. The primary focus is set on the virtual substitution algorithm where a generalization to arbitrary degrees based on the prior work of Weispfenning and Košta is presented. This is followed by the practical aspects of generating the tables associated with the algorithm, considering how polynomial formulas can be simplified. Finally two applications that can be tackled with quantifier elimination and in particular virtual substitution are given, concluding with an outlook on directions for potential future work.

1.1 Mathematical Foundations

1.1.1 Notation

For logic expressions the truth value is herein denoted as $\top$, the false value is $\bot$ and iff is an abbreviation for if and only if. The naturals $\mathbb{N}$ are generally taken to not include 0, and instead denoted as $\mathbb{N}_0$ if that is desired. A collection of variable $y_1, \ldots, y_n$ will be abbreviated with bold font as $\mathbf{y}$ wherever their amount n is insignificant. $R[x]$ denotes the ring of polynomials in the variable x with coefficients in the ring R, often $\mathbb{Z}$ for univariate, or itself a polynomial ring $\mathbb{Z}[\mathbf{y}]$ for multivariate polynomials. A polynomials i-th derivative is denoted as $f^{(i)}$. The functions that give the coefficients and degree of such a polynomial will herein be denoted by:

$$c : \mathbb{Z}[\mathbf{y}][x] \times \mathbb{N} \to \mathbb{Z}[\mathbf{y}], (f, i) \mapsto f^{(i)}(0)/i! \qquad \deg : \mathbb{Z}[\mathbf{y}][x] \to \mathbb{N}, f \mapsto \max\{i \in \mathbb{N} | c(f, i) \neq 0\}$$

$c(f, i)$ will be written as $c_{f,i}$ and the leading coefficient and remaining terms distinguished as

$$\mathrm{lc} : \mathbb{Z}[\mathbf{y}][x] \to \mathbb{Z}[\mathbf{y}], f \mapsto c_{f,\deg(f)} \qquad \mathrm{red} : \mathbb{Z}[\mathbf{y}][x] \to \mathbb{Z}[\mathbf{y}][x], f \mapsto f - \mathrm{lc}(f)x^{\deg(f)}$$

such that a polynomial $f \in \mathbb{Z}[\mathbf{y}][x]$ has the two representations $f = \sum_{i=0}^{\deg(f)} c_{f,i}x^i = \mathrm{lc}(f)x^{\deg(f)} + \mathrm{red}(f)$.

A polynomial $f \in \mathbb{Z}[\mathbf{y}][x]$ is then called monic iff $\mathrm{lc}(f) = 1$ and depressed iff $c_{f,\deg(f)-1} = 0$.

ρ generally denotes a comparator, usually from the set $\{<, =, >\}$ and relations of polynomials $f \,\rho\, g$ are always taken relative to 0 by implicitly considering $f - g \,\rho\, 0$.

For a set S its power set is denoted as $\mathbb{P}^S$ and its set of ordered sets of arbitrary length as $\mathbf{T}_S$. In a slight abuse of notation $r \in t$ is here taken to mean r is a valid index of such an ordered set t and the element at that index is t_r such that furthermore $\sum_{r \in t} t_r$ is abbreviated as $\sum t$.

1.1.2 Polynomials

Interestingly for most of the constructions of this thesis only elementary properties of the roots of polynomials are needed. In the interest of completeness these are restated or derived here before covering the more obscure properties utilized in this thesis.

In general the coefficients of real polynomials are in $\mathbb{R}$. It is however sufficient to consider those in $\mathbb{Z}$ as done here. This is because non-algebraic real coefficients α will never satisfy any polynomial relation and can therefore be substituted with a variable and bounds $ql < p\alpha < qh$ with $ql, p, qh \in \mathbb{Z}$ of sufficient accuracy. Non-rational algebraic real coefficients which therefore are a root of a polynomial can be replaced by a variable and this polynomial and similar appropriate bounds to select the right root. Finally a polynomial with rational coefficients can be reduced to $\mathbb{Z}[x]$ through multiplication with the lowest common multiple of their denominators.

Starting with the most fundamental properties of polynomials:

Theorem 1.1 (Fundamental Theorem of Algebra)**.** *Every nonzero polynomial $f \in \mathbb{Z}[x]$ has exactly* $\deg(f)$ *roots in $\mathbb{C}$ when counted with multiplicity.*

Obviously that implies that f also has at most $\deg(f)$ real roots. Furthermore complex roots only occur in complex conjugated pairs and so it also implies that polynomials of odd degree always have a real root.

Lemma 1.2. *If $z \in \mathbb{C}\backslash\mathbb{R}$ is a root of $f \in \mathbb{Z}[x]$ then so is $\bar{z}$.*

Additionally there are important relations between the roots of a polynomial and those of its derivatives.

Lemma 1.3 (Rolle's theorem)**.** *For all polynomials $f \in \mathbb{Z}[x]$ and $a, b \in \mathbb{R}$ with $a < b$ and $f(a) = f(b) = 0$ there exists $\xi \in (a, b)$ such that $f'(\xi) = 0$.*

This extends so far that that the real roots are distinguishable purely by the sign sequence of the derivatives at the root. Although this won't be used directly here it is inherent in a lot of the constructions that follow.

Theorem 1.4 (Thoms lemma)**.** *If $a, b \in \mathbb{R}$ are two distinct roots of $f \in \mathbb{Z}[x]$ then the two vectors of signs of derivatives $(\text{sign}(f^{(i)}(a)))_{0<i<\deg(f)}$ and $(\text{sign}(f^{(i)}(b)))_{0<i<\deg(f)}$ differ in at least one entry.*

Furthermore the following construct will be used to track the roots of a polynomial with their multiplicities.

Definition 1.5. A polynomial $f \in \mathbb{Z}[\mathbf{y}][x]$ with $\text{lc}(f) \neq 0$ has root type $(r_1, \ldots, r_n)$ iff it has exactly n distinct real roots $y_1 < \ldots < y_n$ and every root y_i has multiplicity r_i.

The root types can be enumerated and thereby counted in the following way:

Theorem 1.6. *The list of root types for a given degree and its size are given by:*

$$\text{RT} : \mathbb{N} \to \mathbb{P}^{T_\mathbb{N}}, n \mapsto \{t \in T_\mathbb{N} | 0 \leq 2i \leq n, \sum t = n - 2i\} \qquad |\text{RT}(n)| = (2^{n+1} + (-1)^n)/3$$

Proof. By 1.1 a polynomial of degree n has exactly n roots which can occur in any possible order of multiplicities and by 1.2 the complex ones always occur in pairs. Thus $\text{RT}(n)$ contains all possible root types for a polynomial of degree n. The amount of root types with l real roots can be counted by their number of real roots disregarding multiplicity k. Because then this is the distinguished boxes and undistinguished objects without empty boxes problem from combinatorics. Thus with k boxes and l objects there are $\binom{l-1}{k-1}$ root types for any k. Since k can be between 1 and l this gives $\sum_{k=1}^{l} \binom{l-1}{k-1} = 2^{l-1}$ for the total. With the accommodation $l = n - 2i$ for complex root pairs this gives $\sum_{i=0}^{\lfloor n/2 \rfloor} 2^{n-2i-1} = 2^{n-1-2\lfloor n/2 \rfloor}(2^{2+2\lfloor n/2 \rfloor} - 1)/3$. This however counts the empty root type () wrong and so 1/2 needs to be added for even n. A case distinction on the parity of n and successive simplification then gives the claimed formula. □

By considering the parameter space where the root type stays constant a specific root of a parametric polynomial $f \in \mathbb{Z}[\mathbf{y}][x]$ can be defined.

Definition 1.7. The expression (f, t, r) denotes the r-th root of a polynomial $f \in \mathbb{Z}[\mathbf{y}][x]$ that has root type t.

Formally this is a partial function $\mathbf{y} \to \mathbb{R}$ but it is primarily understood and used here as a symbolic expression or real number because $\mathbf{y}$ is usually assumed to be fixed.
Obviously every polynomial partitions the reals into three sets, the preimages of its sign. Additionally:

Theorem 1.8. *For every polynomial $f \in \mathbb{Z}[x]\backslash\{0\}$ and relation $\rho \in \{<, =, >\}$ define $V_\rho(f) = \{x \in \mathbb{R} | f(x)\, \rho\, 0\}$. Then $V_=(f)$ is a finite set of points and $V_<(f)$ and $V_>(f)$ are finite unions of open intervals.*

Proof. By 1.1 $V_=(f)$ is finite and thus, by continuity $V_<(f)$ and $V_>(f)$ are at most finite unions. Also by continuity every point in them has an open neighborhood and so they themselves must be open. □

One example of these sets for a polynomial of degree 4 is visualized here:

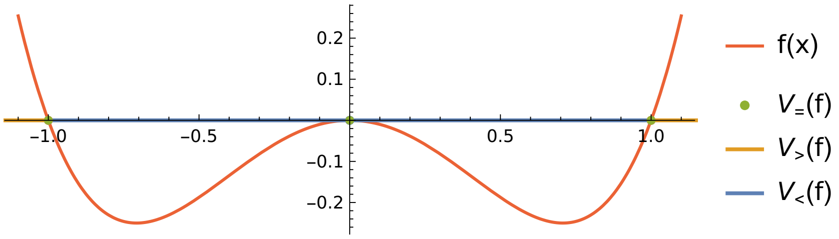


Figure 1: Decompositions $V_\rho(f)$ for $f = (x + 1)x^2(x - 1)$

Furthermore there are transformations of general polynomials that reduce their number of coefficients by two:

Theorem 1.9. *With $n = \deg(f)$ the linear Tschirnhaus transformation gives a monic depressed polynomial:*

$$\mathrm{Tsch} : \mathbb{Z}[\mathbf{y}][x] \to \mathbb{Z}[\mathbf{y}][x], f \mapsto \left(x \mapsto n^n \,\mathrm{lc}(f)^{n-1} f\left(\frac{x - c_{f,n-1}}{n\,\mathrm{lc}(f)}\right)\right)$$

Proof. The result of the transformation is monic because $\mathrm{lc}(\mathrm{Tsch}(f(x))) = n^n \,\mathrm{lc}(f)^{n-1}\,\mathrm{lc}(f)\left(\frac{1}{n\,\mathrm{lc}(f)}\right)^n = 1$ and depressed because $c_{\mathrm{Tsch}(f),\deg(f-1)} = n^n \,\mathrm{lc}(f)^{n-1}\left(\frac{-c_{f,n-1}}{(n\,\mathrm{lc}(f))^{n-1}} + c_{f,n-1}\left(\frac{1}{n\,\mathrm{lc}(f)}\right)^{n-1}\right) = 0$. □

The Tschirnhaus transformation performs 3 transformations at the same time to reach that result. It moves the polynomial along the x-axis to "center" the roots and thereby eliminate the second highest coefficient. It then scales it both in its variable and value to achieve $\mathrm{lc}(f) = 1$ without creating fractional coefficients. Most importantly it doesn't perturb the structure of the roots of the polynomial. It is however not idempotent, instead scaling x by $\frac{1}{n}$ upon application to an already monic depressed polynomial.

Theorem 1.10. *The polynomials f and $\mathrm{Tsch}(f)$ have the same root type if $\mathrm{lc}(f) > 0$.*

Proof. Let t be the root type of f, then it has the factorization $f = \mathrm{lc}(f)\prod_{r\in t}(x - y_r)^{t_r} g(x)$ where g is monic and has only complex roots. Tsch transforms the roots only affinely with positive scaling for $\mathrm{lc}(f) > 0$ and thereby keeps this structure intact, such that $\mathrm{Tsch}(f)$ has root type t as well. □

This especially implies that Tsch reduces the dimension over the coefficients where the polynomial has a certain root type uniformly by 2. At the same time this is the strongest possible reduction with this property:

Theorem 1.11. *The only monic depressed polynomial of degree n with root type (n) is x^n.*

Proof. A polynomial of degree n with root type (n) must have the factorization $a(x - b)^n$. To be monic however a has to be 1 and to be depressed b has to be 0 and so x^n is the only one. □

1.2 Semi-Algebraic Sets and $\mathbf{FOL}_{\mathbb{R}}$

In real algebraic geometry there is a rich connection between first order logic and geometry, similar to that of algebra and geometry in algebraic geometry. Therefore both formulations are introduced here while taking note of their equivalences to later use the form more convenient in the respective context. This exposition is similar to [16] but primarily derived from [8] and [21].

1.2.1 The Geometric Perspective

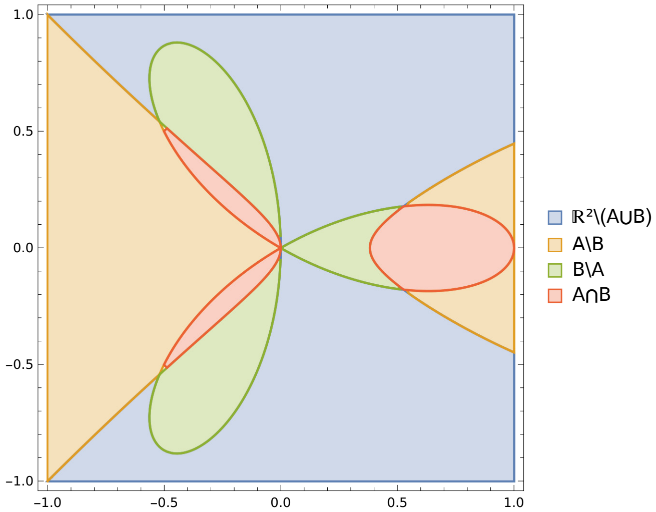
Geometrically semi-algebraic sets are given by:

Definition 1.12. The set of semi-algebraic sets $\mathcal{SA}_n \subset \mathbb{P}^{\mathbb{R}^n}$ is the smallest set such that:

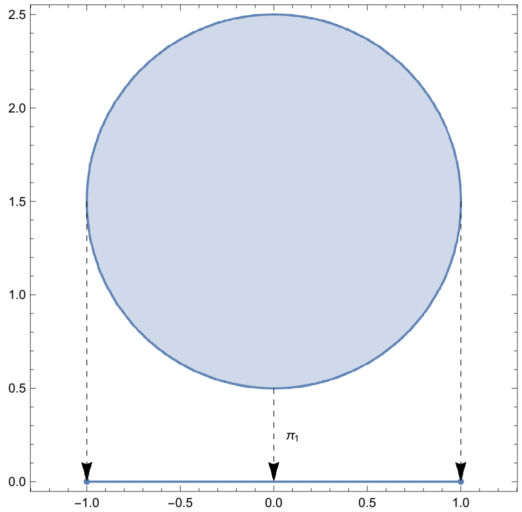
- For all polynomials $f \in \mathbb{R}[x_1, \ldots, x_n]$ it is $V_=(f) \in \mathcal{SA}_n$ and $V_>(f) \in \mathcal{SA}_n$.
- For all sets $A, B \in \mathcal{SA}_n$ it is $A \cup B \in \mathcal{SA}_n, A \cap B \in \mathcal{SA}_n$ and $\mathbb{R}^n \backslash A \in \mathcal{SA}_n$.

This means that the semi-algebraic sets are already axiomatically closed under all set operations and especially under complements; a property that algebraic sets for example don't possess.
Furthermore it shows how general the notion of semi-algebraic sets is because it not only includes a lot of well studied classes of sets, such as conic sections and elliptic curves but also nontrivial fully dimensional components and arbitrary combinations thereof.

Example 1.13. Choosing $A = \{(x, y) \in \mathbb{R}^2 | x - 3x^2 + x^3 + 5y^2 \leq 0\}$, the area enclosed by that particular elliptic curve, and $B = \{(x, y) \in \mathbb{R}^2 | x^4 + 2x^2y^2 + y^4 - x^3 + 3xy^2 \leq 0\}$, the three leaved clover, their combinations tile the plane as shown in the graphic and illustrate how even small combinations of semi algebraic sets with low degree can become complicated.



(a) Example 1.13



(b) Projection of a disk

1.2.2 The Logical Perspective

The model-theoretic approach requires a little bit more machinery. Here, variables are strings, treated as symbolic expressions. Atomic formulas are relations $f\,\rho\,0$ where f is a polynomial in $\mathbb{Z}[V]$ over the variables V and ρ is one of the comparators $\{<, \leq, =, \neq, \geq, >\}$. Formulas then are combinations of atomic formulas with the logical operators $\wedge, \vee, \neg$ and the synonyms $a \Rightarrow b := \neg a \vee b$ and $a \Leftrightarrow b := (a \wedge b) \vee (\neg a \wedge \neg b)$. Already it can be seen how formulas correspond to semi-algebraic sets as laid out in definition 1.12. This is because in the first axiom they are defined from atomic formulas and in the second axiom $\cup, \cap$ and $\mathbb{R}^n\setminus$ correspond directly to $\vee, \wedge$ and $\neg$. Furthermore $\Rightarrow$ corresponds to $\subset$ and $\Leftrightarrow$ to $=$.

So far this defines a propositional logic. To obtain a first order logic it is necessary to introduce quantifiers. Therefore if ϕ is a formula with a free variable x then $\exists x\, \phi(x)$ and $\forall x\, \phi(x)$ are formulas as well. Herein variables are called free if they aren't quantified and bound otherwise.

The set of all such formulas is then the first order language of the reals.

Unfortunately not all concepts that are geometrically easy can be transferred directly to logic. For example the closure of $V_>(x^3 - x^2)$ is not, as one might hope, $V_\geq(x^3 - x^2)$ but instead $V_\geq(x-1)$. Indeed this thesis itself primarily covers how to implement a geometric operation with a logical algorithm.

1.3 Quantifier Elimination

Semi-algebraic sets have the important property of being closed under projections, as was famously established by Tarski through a constructive but algorithmically impractical proof.

Theorem 1.14 (Tarski-Seidenberg)**.** *For all naturals $n \in \mathbb{N}$ and semi-algebraic sets $A \in \mathcal{SA}_{n+1}$ the image of A under the projection $\pi_n : \mathbb{R}^{n+1} \to \mathbb{R}^n, (x_1, \dots, x_{n+1}) \mapsto (x_1, \dots, x_n)$ is also semi-algebraic, $\pi(A) \in \mathcal{SA}_n$.*

Proof. Chapter 2 will provide an algorithm to construct a semi-algebraic representation of $\pi(A)$. □

Following the analogy the effect of the projection operator upon a semi-algebraic set turns out to be equivalent to that of an existential quantifier of the respective variable upon its formula. This is because the existence of such a value allows the construction of an element in the preimage of the projection and vice versa. Graphically this is shown in the (b) part of the figure above for the quantifier elimination $x^2 + (y - 1.5)^2 \leq 1 \overset{\exists y}{\to} x^2 \leq 1$. By repeated application of the Tarski-Seidenberg theorem every formula therefore has a quantifier free equivalent. The question of how this can be found is important enough to have its own name:

Definition 1.15. The quantifier elimination problem is finding a quantifier free formula ψ such that:

$$(Q_k x_{k,1}, \dots, x_{k,r_k}) \dots (Q_1 x_{1,1}, \dots, x_{1,r_1})\phi(\mathbf{x}) \Leftrightarrow \psi \qquad \text{with} \qquad Q_i \in \{\exists, \forall\}$$

Formulas are generally assumed to be in prenex form because non prenex formulas can be viewed as successive parametric prenex problems. Additionally the decision problem, whether a given formula is true or false, is the special case of the quantifier elimination problem with no free variables. For theoretical purposes it is however not necessary for an algorithm to handle the general form with it's blocks of quantifiers. Instead it can be defined without loss of generality for various reduced forms, for example:

Theorem 1.16. *Solving the quantifier elimination problem is equivalent to being able to find quantifier free formulas for all expressions of the special form:*

$$\exists x \bigwedge_i f_i(x)\,\rho_i\,0 \qquad \textit{with} \qquad \rho_i \in \{<,=,>\} \quad \textit{and} \quad \mathrm{lc}(f_i) > 0$$

Proof. $\Rightarrow$: A general method can obviously always eliminate the one quantifier in the special form.
$\Leftarrow$: Thanks to the relation $\forall \mathbf{x}\,\phi \Leftrightarrow \neg\exists\mathbf{x}\,\neg\phi$ it is sufficient to consider existential quantifiers. It is also well known from computer science that every boolean formula can be rewritten in disjunctive normal form (DNF) through application of the de Morgan laws and other equivalences. This gives $\phi \Leftrightarrow \bigvee_i \bigwedge_j \phi_{i,j}$ with all $\phi_{i,j}$ atomic and no negations because those can be absorbed into the relations via $<\overset{\neg}{\leftrightarrow}\geq$, $=\overset{\neg}{\leftrightarrow}\neq$ and $>\overset{\neg}{\leftrightarrow}\leq$. Furthermore this ensures that every possible formula can be handled and therefore the process can be applied repeatedly which means it is enough to be able to eliminate one quantifier at a time. Next the substitutions

$$f \geq 0 \to f > 0 \vee f = 0 \qquad f \neq 0 \to f > 0 \vee f < 0 \qquad f \leq 0 \to f = 0 \vee f < 0$$
$$\phi(f) \to \mathrm{lc}(f) > 0 \wedge \phi(f) \vee \mathrm{lc}(f) = 0 \wedge \phi(\mathrm{red}(f)) \vee \mathrm{lc}(f) < 0 \wedge \phi(-f)$$

can be applied, the second one recursively so, to get the restricted set of relations and positivity of leading coefficients. By extensive usage of the distributive law the resulting formula can once again be converted to DNF. That finally allows for the formula to be reduced to many problems of the desired form through the application of the relations $\exists x\,\phi(x) \vee \psi(x) \Leftrightarrow (\exists x\,\phi(x)) \vee (\exists x\,\psi(x))$ and $\exists x\,\phi \wedge \psi(x) \Leftrightarrow \phi \wedge \exists x\,\psi(x)$. □

1.3.1 Quantifier Elimination Algorithms

There are essentially three classes of quantifier elimination algorithms that have been practically implemented. Historically Collins algorithm [7] was developed first, based on calculating cylindrical algebraic decompositions. Since then it has been improved in many ways and various variations exist. It is still prevalent and its implementations in QEPCAD [3] and Mathematica [13] are used later on in this thesis. The second one, virtual substitution is covered extensively in the next section. Finally there are comprehensive Gröbner basis techniques, introduced by Weispfenning [24] which became effective thanks to further development and were for example used successfully in [10].
The CAD algorithms solve a significantly harder problem than quantifier elimination because they give an explicit representation in terms of cells homeomorphic to cubes $(0,1)^d$ from which additional properties such as dimension and connected components can be easily extracted. This stands in contrast to the algorithm presented here where the representations remain implicit and even questions such as whether the resulting sets are empty are hard to answer. In exchange the asymptotic complexity for a single quantifier elimination of the cylindrical algebraic decomposition algorithms are as bad as $L^3(md)^{2^{\mathcal{O}(n)}}$ where n is the number of variables, m is the number of polynomials, d their degree and L their coefficient size. Interestingly there are other algorithms with significantly better asymptotic complexity such as $L^{1+\epsilon}(md)^{\mathcal{O}(n)}$ although these have not been implemented because their running time is prohibitive even for tiny inputs, as argued in [11].
In contrast in [27] it was shown that for linear polynomials the quantifier elimination problem is doubly exponential only in the number of quantifier alternations, exponential in the number of quantifiers and polynomial in the number of variables. Indeed this is the advantage of virtual substitution as presented here; it handles many parameters and large formulas well but very much struggles with higher polynomial degree and multiple quantifiers. This is why the comprehensive Gröbner basis approach is interesting, because at every step it eliminates blocks of quantifiers while potentially retaining the good complexity properties of virtual substitution in terms of the number of parameters. Note for example that none of the quantifier elimination problems occurring later on have a quantifier alternation and would therefore be well suited to this technique. Unfortunately exploring this is beyond the scope of this thesis.

2 General Virtual Substitution

This section starts by reviewing the historical development of virtual substitution to give an overview of the technique, it's ideas and how they were successively built and developed upon. After that a minimal framework that reduces the problem of performing quantifier elimination upon expressions in the form of theorem 1.16 into guards $\gamma_t(f)$ and substitutions $f[x\backslash\backslash a]\,\rho\,0$ is defined. Guards $\gamma_t(f)$ are the conditions on the coefficients of f that need to hold for it to have root type t and are constructed in section 2.3. Substitutions $f[x\backslash\backslash a]\,\rho\,0$ are formulas purely in the coefficients of f that are equivalent to $f(a)\,\rho\,0$ and are constructed in section 2.4.

2.1 History

The technique of virtual substitution was first conceptualized by Weispfenning in [27] and used to derive bounds on the complexity of the decision and quantifier elimination problems for linear formulas over various fields. Fundamentally this relied on the following two concepts to eliminate one quantifier at a time:
Firstly every quantified formula is equivalent to its evaluation at a finite set of points because polynomials can change their signs only finitely many times. Explicitly this can be expressed as:

$$\exists x\ \phi(x) \Leftrightarrow \bigvee_{y \in Sk(\phi)} \phi[x\backslash\backslash y] \qquad \forall x\ \phi(x) \Leftrightarrow \bigwedge_{y \in Sk(\phi)} \phi[x\backslash\backslash y]$$

Ignoring here some of the optimizations that were made in principle these finite sets $Sk(\phi)$, called the Skolem sets, were constructed as the set of roots $\frac{-b}{a}$ occurring in atomic formulas $ax + b\,\rho\,0$ of ϕ as well as their offsets by ± 1 and the average of any two roots. This way the Skolem set is guaranteed to contain a point for every realizable combination the atomic formulas and their negations could form. This was necessary because ϕ was treated as a black box. Here $a \neq 0$ was assumed, since otherwise an atomic formula is already a quantifier free relation $b\,\rho\,0$.

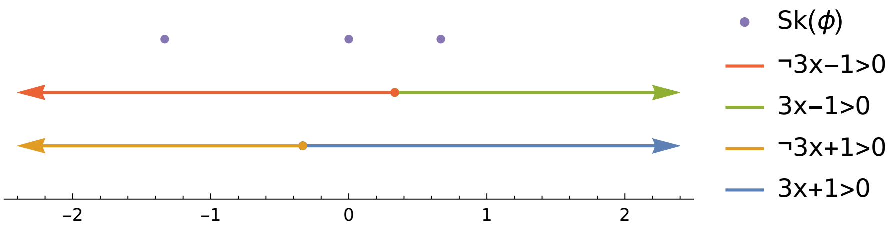


Figure 2: Example of a Skolem set covering all realisable combination of two linear inequalities

Secondly, even though division is not an allowed operation, one can "virtually" substitute the roots, their offsets and averages for x into the formulas so long as the denominators are canceled out afterwards. Formally this is justified by performing the substitution in an extended theory, where it is well defined, and then transforming it back into the original theory. Notice how this method really only requires linearity in the bound variables since by placing the $a \neq 0$ distinction into the formula the coefficients could take arbitrary degree in the free variables.
Over the next years the applicability of the technique was extended to quadratic polynomials, which was then implemented in RedLog, and a framework for arbitrary degrees was sketched in [26]. The quadratic algorithm differs from the linear one in three points:
Firstly the concept of the Skolem set which covered all conceivable cases for the atomic formulas and their negations is replaced with the elimination set which utilizes the structure of ϕ and is chosen to essentially only contain roots of atomic formulas.
Secondly, to do this the fact that atomic formulas can be evaluated at polynomial roots $\pm\epsilon$ as well as $\pm\infty$ by using equivalences such as $f(\infty) = 0 \Leftrightarrow \bigwedge_{i=0}^{\deg(f)} c_{f,i} = 0$ was utilized.
Thirdly the virtual substitution of the well known quadratic solution formula $(-b \pm \sqrt{b^2 - 4ac})/2a$ was defined in a manner similar to the field extensions $\mathbb{Q}[\sqrt{p}]$ such that atomic formulas in the normal form $a + b\sqrt{c}\,\rho\,0$ result, which can then be transformed back into polynomial relations through another set of equivalences like:

$$a + b\sqrt{c} = 0 \Leftrightarrow ab \leq 0 \wedge a^2 = b^2 c \qquad a + b\sqrt{c} \geq 0 \Leftrightarrow (a \geq 0 \wedge a^2 \geq b^2 c) \vee (b \geq 0 \wedge b^2 c \geq a^2)$$

Theoretically this approach could have been extended to degree 3 and 4 with the respective solution formulas for those degrees but due to their unwieldiness this seems to have never been attempted. The proposal for the general case instead kept the first two points, elimination sets and evaluation at nonstandard symbols, and proposed to replace the symbolic computation involving solution formulas with Thoms encoding of the roots of a polynomial. This was not in itself a quantifier elimination algorithm because it assumed that quantifier free equivalents were available for all relevant polynomials f, g and comparators ρ for the formulas:

$$\exists x\; f(x) = 0 \wedge \bigwedge_{i=1}^{\deg(f)-1} f^{(i)}(x)\, \rho_i\, 0 \wedge g(x)\, \rho_{\deg(f)-1}\, 0$$

Diverging from that framework Weispfenning then extended the algorithm to degree 3 in [25] by first deriving formulas to determine the real root type of a polynomial of degree 3 and giving formulas to evaluate relations at their roots. This was based on root interval and recursive techniques which will be build upon extensively later in this section. Additionally he stated that the generalization of this approach to arbitrary degrees was possible and in preparation, but has, to the best of the authors knowledge, not published it.

It was upon this foundation that Košta and Sturm built in [15] when they improved the construction of elimination sets and implemented the general framework of Thom codes with an external quantifier elimination procedure for the aforementioned formulas. Independently the framework based upon Thom codes was also implemented in [20] using Hermitian matrix signatures to determine sign conditions.

In his PhD thesis [14] Košta laid out the modern version of the virtual substitution framework originally introduced by Weispfenning for degree 3, re-derived his degree 3 formulas, implemented them in REDLOG and introduced the practically important techniques of clustering and construction of elimination sets based upon the formula structure. Thus the rest of the chapter is primarily build on his thesis, first formulating a minimal version of his framework and then showing how the two functions that were still missing to apply it to arbitrary degrees, guard-n and vs-prd-at-n, can be implemented.

2.2 Framework

Practically the form of theorem 1.16 introduces needless exponential algorithmic complexity twice, once through the conversion to DNF and once trough the reduction to $\{<, =, >\}$. This is not inherent though, the first can be overcome through the technique in chapter 3 of [14] and the second by also deriving relations for $f \gtreqless 0$ and utilizing $f \neq 0 \Leftrightarrow \neg f = 0$. Both of these were omitted though, because they don't add to the theoretical aspects of what is presented here.

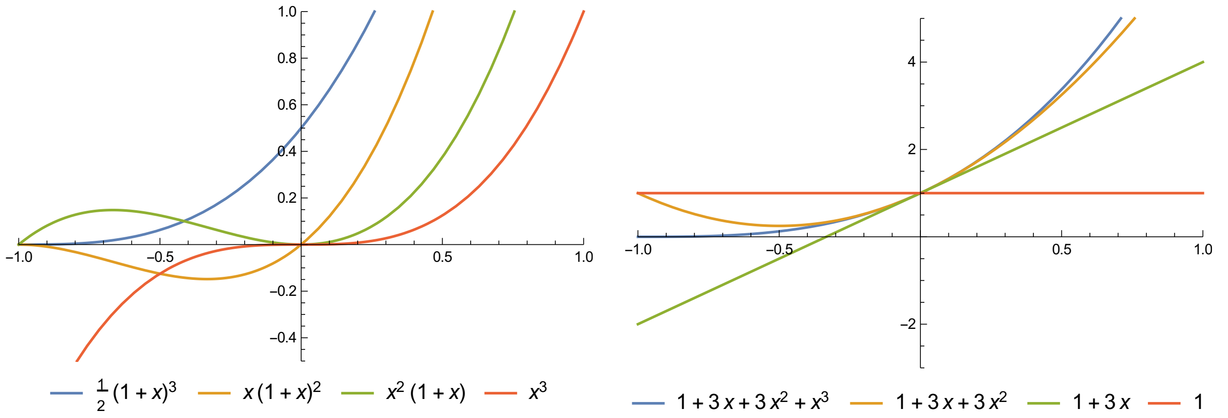
2.2.1 Infinitesimals

Polynomial relations can be evaluated in terms of their coefficients at the nonstandard positions $a \pm \epsilon$ if they can be evaluated at a. To do so derivatives and continuity are utilized in the following way:

Theorem 2.1. *For a polynomial $f \in \mathbb{Z}[\mathbf{y}][x]$ with $\operatorname{lc}(f) > 0$ it is:*

$$f[x \backslash\backslash a \pm \epsilon] = 0 \Leftrightarrow \bot \qquad f[x \backslash\backslash a \pm \epsilon] \gtrless 0 \Leftrightarrow \bigvee_{i=0}^{\deg(f)} (\pm 1)^i f^{(i)}[x \backslash\backslash a] \gtrless 0 \wedge \bigwedge_{j=0}^{i-1} f^{(j)}[x \backslash\backslash a] = 0$$

Proof. If w.l.o.g. $f(a) > 0$ by continuity there is a $\delta > 0$ such that for all $0 < \epsilon < \delta$ it also is $f(a \pm \epsilon) > 0$. If otherwise a is a root of f the close behavior is determined by the sign of the first nonzero derivative. From this the second set of relations follows by recursion on the degree. The first follows from the fact that only the zero polynomial, which is excluded by the premise, is constant on an interval. □



(a) Cases of $f[x\backslash\backslash 0+\epsilon]$ for a polynomial of degree 3 (b) $(x+1)^3$ reduced at $x=0$ with pseudo remainders.

2.2.2 Pseudo-Remainder

The technique of calculating pseudo remainders is of paramount importance in reducing the number of cases that need to be considered for a particular degree to a finite amount. This is because it allows for the reduction of a polynomial relation of arbitrary degree evaluated at a root of f to one of degree at most $\deg(f)-1$.

Theorem 2.2. *For every relation $g[x\backslash\backslash(f,t,r)]\,\rho\,0$ there is an equivalent relation $\tilde{g}[x\backslash\backslash(f,t,r)]\,\rho\,0$ such that $\deg(\tilde{g})<\deg(g)$ if $\deg(g)\geq\deg(f)$.*

Proof. By the premise $\mathrm{lc}(f)>0$ and at (f,t,r) it is $f=0$. So multiplication with $\mathrm{lc}(f)$ and the replacement rule $\mathrm{lc}(f)x^{\deg(f)}\mapsto-\mathrm{red}(f)$ can be applied without changing the truth value of the relation. This gives:

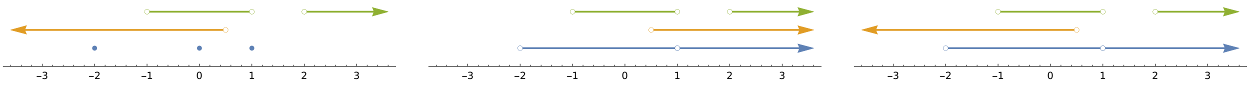
$$\mathrm{lc}(f)g=(\mathrm{lc}(f)x^{\deg(f)})x^{\deg(g)-\deg(f)}+\mathrm{red}(g)\mapsto-\mathrm{red}(f)x^{\deg(g)-\deg(f)}+\mathrm{red}(g)\,\rho\,0$$

Which is a polynomial of degree at most $\deg(g)-1$, giving the desired result. □

Through repeated application of this theorem the degree of a relation can therefore be reduced to $\deg(f)-1$.

2.2.3 Elimination Sets and Formulas

An elimination set E for a formula $\phi=\bigwedge_i f_i(x)\,\rho_i\,0$ is a set of positions such that $\psi=\bigvee_{a\in E}\phi[x\backslash\backslash a]$ is a quantifier free equivalent of $\exists x\,\phi(x)$, meaning there exists a valid x iff ϕ is true at a position of E. With the restrictions of theorem 1.16 the construction of these elimination sets consists of three cases:



(c) equality case (d) infinity case (e) interval case

Theorem 2.3. *If there is an $i\in I_\phi$ such that $\rho_i\in\{=\}$ for ϕ then E and ϕ can be chosen as:*

$$E=\{(f_i,t,r)|t\in\mathrm{RT}(\deg(f_i)),r\in t\}\qquad\psi=\bigvee_{(f_i,t,r)\in E}\gamma_t(f_i)\wedge\bigwedge_{j\in I_\phi\setminus\{i\}}f_j[x\backslash\backslash(f_i,t,r)]\,\rho_j\,0$$

Proof. Since $f_i(x)=0$ has to hold for ϕ to be true x can only be a root of f_i. It is therefore sufficient to construct the elimination set from all the roots under all the root types f_i could assume and evaluate ϕ at them while ensuring the type of f_i. □

In practice it is of course advisable to pick the equality with the lowest degree. The remaining two cases construct the test points based on the potential upper bounds of the polynomials.

Theorem 2.4. *If $\rho_i\in\{>\}$ for all $i\in I_\phi$ of ϕ then $E=\{\infty\}$ is an elimination set for it and $\psi=\top$.*

Proof. By the premise $\mathrm{lc}(f_i)>0$ holds and therefore it is $\lim\limits_{x\to\infty}f(x)=\infty$ for all $i\in I_\phi$ so that at some point all polynomials will be positive, which guarantees the existence of an x at which ϕ is true. □

In contrast to this the third case gives a considerably bigger elimination set and formula:

Theorem 2.5. *If $\rho_i \in \{<,>\}$ for all $i \in I_\phi$ of ϕ then E and ϕ can be chosen as:*

$$E = \{(f_i, t, r) - \epsilon | i \in I_\phi, t \in \mathrm{RT}(\deg(f_i)), r \in t : f_i[x\backslash\backslash(f_i, t, r) - \epsilon]\, \rho_i\, 0\}$$
$$\psi = \bigvee_{(f_i,t,r)-\epsilon \in E} \gamma_t(f_i) \wedge \bigwedge_{j \in I_\phi \setminus \{i\}} f_j[x\backslash\backslash(f_i, t, r) - \epsilon]\, \rho_j\, 0$$

Proof. Each of the inequalities is satisfied on a finite union of open intervals $V_{\rho_i}(f_i)$. The region where ϕ is true is $U = \cap_i V_{\rho_i}(f_i)$ and therefore a union of open intervals itself. By including a test point an epsilon to the left of every possible upper bound of such an interval it is guaranteed that U is not empty iff ϕ is true for one of the $a \in E$, once again allowing for the root type of f. □

Here $f_i[x\backslash\backslash(f_i, t, r) - \epsilon]\, \rho_i\, 0$ can be easily evaluated from the known root type of f_i.

2.3 Guards

The aim of this section is to present the procedure by which the condition $\gamma_t(f)$ under which the polynomial f has root type t can be obtained purely as a formula in the parameters of f. There have also been explicit derivations such as [22] of the conditions on the coefficients for polynomials of degree ≤ 4 that must hold such that specific amounts of multiple roots or complex roots can exist. This has even been extended to algorithmic approaches such as [19], but none of them seem to include the order of the real roots. To remedy this in previous work on virtual substitution the sequence of values of the step function $\mathrm{sign}(f(x))$ has been used under the name real root type of f. This however clusters multiple roots and thereby hides the number of complex roots which is important to the coming constructions. So instead the notion of root types defined in section 1.2 is used here. Note how this is equivalent to real root types for $\deg(f) \leq 2$ and almost equivalent for $\deg(f) = 3$, except for bundling (3) and (1) into $(-1, 0, 1)$. Finding such conditions can itself can be formulated as a quantifier elimination problem, similar to Proposition 8 in [14].

Theorem 2.6. *For each $d \in \mathbb{N}$ the polynomial $f = x^d - \sum_{i=0}^{d-2} y_i x^i \in \mathbb{Z}[\mathbf{y}][x]$ has root type $(r_1, \ldots, r_n)$ iff it is $(d - \sum_{i=1}^{n} r_i)/2 = m \in \mathbb{N}_0$ and with $g = \prod_{i=1}^{n}(x - a_i)^{r_i} \prod_{i=1}^{m}(x^2 + 2b_i + c_i) \in \mathbb{Z}[\mathbf{y}][\mathbf{a}, \mathbf{b}, \mathbf{c}][x]$ it holds that*

$$\exists a_1, \ldots, \exists a_n \exists b_1 \exists c_1 \ldots \exists b_m \exists c_m \bigwedge_{i=1}^{n-1} a_i < a_{i+1} \wedge \bigwedge_{i=1}^{m} b_i^2 < c_i \wedge \bigwedge_{i=0}^{d} c_{f,i} = c_{g,i}$$

Proof. " $\Rightarrow$ " : The condition $m \in \mathbb{N}_0$ follows from theorem 1.2 and a factorization like g with $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}$ exists because of the root type. In fact after sorting $\mathbf{a}$ the formula is satisfied with these values.
" $\Leftarrow$ " : The condition on the a_i ensures that g has root type $(r_1, \ldots, r_n)$ and the b_i and c_i together with m ensure that enough complex conjugated root pairs can be added to g to make it have degree d. Finally the condition on the coefficients ensures $f = g$ and so f has root type $(r_1, \ldots, r_n)$ as well. □

Primarily this ensures that the desired formulas exist because quantifier elimination is possible. But secondarily this formulation is actually efficient enough that it can be used to derive some of the guards with small amount of variables through the cylindrical algebraic decomposition algorithm.
To construct them directly however it is necessary to investigate the relation between the degree of f, its root type, the root types f' can have and how all those roots are positioned relative to one another. To keep track of this the following construction is used in this section:

Definition 2.7. Let $((r_{1,1}, r_{1,2}), \ldots, (r_{n,1}, r_{n,2})) \in \mathbf{T}_{\mathbb{N}_0 \times \mathbb{N}_0}$ be an ordered list of tuples of natural numbers. This is the differential root type of $f \in \mathbb{Z}[x]$ if there exist $y_1 < \ldots < y_n \in \mathbb{R}$ such that the order of y_i as a root of f is $r_{i,1}$ and $r_{i,2}$ as a root of f'.

Analogous to before with the ordinary root type the set of differential root types that can occur together with its size are now derived:

Theorem 2.8. *For a fixed root type and degree the differential root types a polynomial can have are given by:*

$$\mathrm{DRT} : \mathbf{T}_{\mathbb{N}} \times \mathbb{N} \to \mathbb{P}^{\mathbf{T}_{\mathbb{N}_0\times\mathbb{N}_0}}, (t,n) \mapsto \{dt \in \mathbf{T}_{\mathbb{N}_0\times\mathbb{N}_0} | \exists f \in \mathbb{Z}[x] : \deg(f) = n, f \textit{ has root type } t \textit{ and diff. root type } dt\}$$

$$|\,\mathrm{DRT}((), 2m)| = \sum_{i=0}^{m-1} \sum_{j=0}^{m-1-j} \sum_{k=0}^{m-1-i-j} \binom{1+i}{i}\binom{1+2i+j}{j}\binom{2i+j+k}{k} = (3m-4+4^m(3m+4))/27$$

$$|\,\mathrm{DRT}(t, n)| = \sum_{i=0}^{m} \sum_{j=0}^{m-j} \sum_{k=0}^{m-i-j} \binom{l+i}{i}\binom{2l-1+2i+j}{j}\binom{l-2+2i+j+k}{k}$$

where t is nonempty, $m = \left(n - \sum t\right)/2 \in \mathbb{N}_0$ and $l = \sum_{r\in t} 1$ in the last equation.

Proof. Counting the number of differential root consists of three cases of counting indistinguishable objects in distinguishable boxes. The general case of the last equation will be covered first because the special case of the empty root type then derives from it. Herein m is the number of complex roots pairs of f and l is the number of roots in f, ignoring multiplicity. By theorem 1.3 f' must have at least one root between two adjacent roots of f and in general their sum of multiplicities has to be odd. Therefore every differential root type of f with root type $(r_1, \ldots, r_n)$ must have at least the form $((r_1, r_1 - 1), (0, 1), (r_2, r_2 - 1), \ldots, (0, 1), (r_n, r_n - 1))$.
Now the three ways a complex conjugated can turn into two real roots needs to be considered:
It can become two single roots $((0, 1), (0, 1)$ of f' which, because of the aforementioned odd multiplicity restriction, can only be inserted in the same box relative to the roots of f and are indistinguishable from one another and the already placed simple roots of f'. Since there are $l + 1$ boxes relative to the roots of f this contributes $\binom{l+i}{i}$ cases, where $0 \le i \le m$ is the number of complex conjugate root pairs used for this.
Alternatively it can become a double root $(0, 2)$ of f' which can be inserted anywhere but is once again not distinguishable from itself. There were l roots of f to start with and by now $l - 1 + 2i$ simple roots of f' have been added. Thus there are $2l - 1 + 2i + 1$ boxes which gives another factor of $\binom{2l-1+2i+j}{j}$ where, once again, $0 \le j \le m - i$ is the number of complex conjugate root pairs used.
Lastly it can increase the order of one of the already inserted roots of f' by 2. To keep the box analogy these are though of as being inserted to the right of the root they're incrementing. In the previous steps $l - 1 + 2i$ simple roots and j multiple roots of f' have been inserted, which gives a factor of $\binom{l-2+2i+j+k}{k}$ for $0 \le k \le m - i - j$ used complex conjugate root pairs.
Because every one of these cases gives a unique differential root type the final result derives from multiplying all the factors and summing over the three indexes. The sum over k of the last factor can be evaluated to $\frac{m+1-i-j}{l-1+2i+j}\binom{m+l-1+i}{m+1-i-j}$, but no reduced formula could be found beyond that.
The case of the root type $()$ is special because the degree reduction of the differentiation is not paid from the real roots of f but instead a complex conjugate root pair becomes the simple root $(0, 1)$. Consequently the corrected formula uses $m - 1$ and $l = 1$, except in the last factor where $l = 2$ is necessary. The explicit formula was then found with the help of Mathematica. □

Note that the root type of f and f' can be easily extracted from a differential root type by taking the first or second component respectively of every tuple. Furthermore if f and f' have differential root type dt then that determines the sign they have at every root in the differential root type because at ∞ they are both positive and at every root their sign gets multiplied by $(-1)^o$ where o is the order of the root.
The formulas for the guards can now be constructed through recursion on the degree by partitioning every root type into its differential root types with additional conditions to ensure their relation holds as desired.
The base case is simple, a constant polynomial has no root and a linear one always has one root which gives $\gamma_{()}(f_0) = \gamma_{(1)}(f_1) = \top$ for $\deg(f_0) = 0$ and $\deg(f_1) = 1$.
By theorem 2.8 DRT gives all the differential root types f can have with root type t. So conversely by constructing conditions for a differential root type to hold with f having root type t gives a formula for the guard γ_t. This can be accomplished in the following way:

Theorem 2.9. *A polynomial $f \in \mathbb{Z}[x]$ with $\deg(f) \geq 2$ has root type t iff $\gamma_t(f)$ is satisfied, defined as:*

$$\gamma_t(f) = \bigvee_{dt \in \mathrm{DRT}(t, \deg(f))} \gamma_{t'}(f') \wedge \bigwedge_{\substack{(r_{f'}+1, r_{f'}) \in dt \\ r_{f'} > 0}} f[x \backslash\backslash (f', t', r')] = 0 \quad \wedge$$

$$\bigwedge_{\substack{(0, r_{f'}) \in dt \\ r_{f'} \ odd}} f[x \backslash\backslash (f', t', r')] \gtrless 0 \wedge \bigwedge_{\substack{((0, r_{f'}), (1,0)) \subset dt \\ ((1,0), (0, r_{f'})) \subset dt \\ r_{f'} \ even}} f[x \backslash\backslash (f', t', r')] \gtrless 0$$

where t' is the root type of f' extracted from dt, r' is the index of $r_{f'}$ determined by dt, and which of $\gtrless$ is used follows from the sign f should have at that root according to dt.

Proof. The conjunction needs to provide for two things, f having differential root type dt and root type t. The guard of f' right away provides for half the differential root type, namely the multiplicity and order of all roots of f'. The next condition ensures that f has multiple roots at the right positions because a root of multiplicity t_r in f has multiplicity $t_r - 1$ in f'. So the other way around if through virtual substitution it is ensured that $f = 0$ at the respective root that increases the multiplicity of the root by 1, as desired. This still leaves the problem of ensuring that all the simple roots of f occur in the right places and only exactly there, which is provided by the second row of conditions. A root of odd order in f' is an extrema of f and fixing their sign though a virtual substitution ensures that the roots of f' lie in the right interval of roots of f, that there are no undesired multiple roots and finally that f itself only has roots in the right intervals of roots of f'. Some of these constraints can be redundant, for example the positivity condition on a maxima in a region where f is positive for a root that is between two other odd roots of f'. The last ambiguity is the position of even roots of f', which are saddle points of f with respect to simple roots of f. However this is easily alleviated by the last set of inequalities. □

This equivalence of course only reformulates the problem in terms of another guard and many virtual substitution, but true to the requirements of the recursion those are all over a polynomial of degree $\deg(f) - 1$.

2.4 Substitution

The only aspect of the virtual substitution algorithm that is left now is also the most complicated one, the virtual substitution, namely constructing formulas for all relations $g[x \backslash\backslash (f, t, r)]$. Thanks to the pseudo-remainder algorithm of 2.2.2 though right away only relations with $\deg(g) < \deg(f)$ need to be considered.
The virtual substitution can also itself be formulated as a quantifier elimination problem.

Theorem 2.10. *For each $d \in \mathbb{N}$ the polynomial $g = \sum_{i=0}^{d-1} y_i x^i \in \mathbb{Z}[\mathbf{y}][x]$ satisfies the relation $g \, \rho \, 0$ at the r-th root of the polynomial $f = x^d - \sum_{i=0}^{d-2} y_{i+d} x^i \in \mathbb{Z}[\mathbf{y}][x]$ with root type $(r_1, \ldots, r_n)$ iff it holds that*

$$\exists z_1, \ldots \exists z_n : \gamma_t(f) \wedge \bigwedge_{i=1}^{n-1} z_i < z_{i+1} \wedge \bigwedge_{i=1}^{n} f(z_i) = 0 \wedge g(z_r) \, \rho \, 0$$

Proof. " $\Rightarrow$ " : By the premise f has root type $(r_1, \ldots, r_n)$ and therefore exactly the n roots $z_1, \ldots, z_n$. The value of the relevant relation can then simply be evaluated at z_r.
" $\Leftarrow$ " : $\gamma_t(f)$ ensures the type of f and $g(z_r) \, \rho \, 0$ the value of the relation. □

Because this formulation uses up to $3d - 1$ variables it is not practically useful for deriving formulas with the aid of the cylindrical algebraic decomposition algorithm.
Instead the algorithm is constructed here once again by a recursion on the degree of f:
If $\deg(f) = 0$ then f is constant in x and so the relation $f[x \backslash\backslash a] \, \rho \, 0$ is equivalent to $f(0) \, \rho \, 0$ for all a. Furthermore, through the pseudo-remainder approach of section 2.2.2 the evaluation of any relation $g[x \backslash\backslash (f, (1), 1)] \, \rho \, 0$ with $\deg(f) = 1$ can be reduced to $\deg(\tilde{g})$ and thereby $\tilde{g}(0) \, \rho \, 0$ which is free of x. Unfortunately though in general the pseudo remainder process invalidates all information on the type and the sign of the leading coefficient of g that might be available. That means that all possible cases for the degree and root type of g have to be considered, which leads to the following:

Theorem 2.11. *The formula for $g[x\backslash\backslash(f,t,r)]\,\rho\,0$ with $n = \deg(g) < \deg(f)$ can be constructed recursively as:*

$$g[x\backslash\backslash(f,t,r)]\,\rho\,0 \Leftrightarrow \left(\mathrm{lc}(g) > 0 \wedge \bigvee_{tg\in\mathrm{RT}(n)} \gamma_{tg}(g) \wedge g_t[x\backslash\backslash(f,t,r)]\,\rho\,0\right) \vee (\mathrm{lc}(g) = 0 \wedge \mathrm{red}(g)[x\backslash\backslash(f,t,r)]\,\rho\,0) \vee$$
$$\left(\mathrm{lc}(g) < 0 \wedge \bigvee_{tg\in\mathrm{RT}(n)} \gamma_{tg}(-g) \wedge -g_t[x\backslash\backslash(f,t,r)]\,\rho\,0\right)$$

Proof. Since the right hand sight partitions the formula into exclusive and exhaustive regions based on sign, degree and root type logical equivalence is maintained. □

This however just reduces the problem to constructing $g_t[x\backslash\backslash(f,t,x)]$, but now with the added benefit of g having know root type tg and $\mathrm{lc}(g) > 0$. To do this it is useful to frame the question whether the relation is true in terms of the roots of f and g.

Theorem 2.12. *With $tg = (rg_1,\dots,rg_n)$ the following equivalences hold:*

$$g_{tg}[x\backslash\backslash(f,t,r)] > 0 \Leftrightarrow (\deg(g)\ \textit{even} \wedge (f,t,r) < (g,tr,1)) \quad\vee \bigvee_{\substack{i=1\\ g[x\backslash\backslash(g,tg,tg_i)+\epsilon]>0}}^{n-1} (g,tr,i) < (f,t,r) < (g,tr,i+1)$$
$$\vee (g,tg,n) < (f,t,r)$$

$$g_{tg}[x\backslash\backslash(f,t,r)] = 0 \Leftrightarrow \bigvee_{rg\in tg} (f,t,r) = (g,tg,rg)$$

$$g_{tg}[x\backslash\backslash(f,t,r)] < 0 \Leftrightarrow (\deg(g)\ \textit{odd} \wedge (f,t,r) < (g,tr,1)) \quad\vee \bigvee_{\substack{i=1\\ g[x\backslash\backslash(g,tg,tg_i)+\epsilon]<0}}^{n-1} (g,tr,i) < (f,t,r) < (g,tr,i+1)$$

Proof. As in theorem 1.8 for every g the relations tile the reals and $g_{tg}[x\backslash\backslash(f,t,r)]\,\rho\,0 \Leftrightarrow (f,t,r) \in V_\rho(g)$. Since the endpoints of the intervals in the $V_{\lessgtr}(g)$ are exactly the roots of g and the intervals are selected based on the sign of g on them, the claimed equivalences follow. □

Note that equivalent to the case of the interval elimination set and the differential root type for the guards the sign of g on any of its intervals, $g[x\backslash\backslash(g,tg,tg_i)+\epsilon] \gtrless 0$, can be easily determined from its root type.
With this decomposition the problem is now also reduced further to finding equivalents for the relations $(f,t,x) = (g,tg,rg)$ and $(f,t,x) \lessgtr (g,tg,rg)$ in terms of the polynomials f and g. And indeed this is the final part of this construction. For this it is however necessary to fix the differential root type of f, which means another case distinction that extends the formula of the form $\bigvee_{dt} \psi_{f,dt}$ derived in theorem 2.9 by additional conjunctions is necessary. Starting with the case of equality:

Theorem 2.13. *For the equality of two roots of polynomials of known type it holds that:*

$$(f,t,r) = (g,tg,rg) \Leftrightarrow \bigvee_{dt\in\mathrm{DRT}(t,\deg(f)))} \psi_{f,dt} \wedge \begin{cases} (f',t',r') = (g,tg,rg) & \textit{for } t_r > 1 \textit{ else:} \\ f[x\backslash\backslash(g,tg,rg)] = 0 \wedge (f',t',r'_r) < (g,tg,rg) < (f',t',r'_{r+1}) \end{cases}$$

where t' is the root type of f' derived from dt, r' is the position of the multiple root in t' and r'_r and r'_{r+1} are the roots of f' immediately to the left and right of (f,t,r).

Proof. If (f,t,r) is a multiple root f' has a root at the same position (with a different index) and so the comparison can be performed with it instead. Otherwise, like in theorem 2.12, $f[x\backslash\backslash(g,tg,rg)] = 0$ is equivalent to $\bigvee_{rf\in t}(g,tg,rg) = (f,t,rf)$. Since (f,t,r) is the only root of f between (f',t',r_r) and (f',t',r_{r+1}) and the root is simple the restriction gives the desired equivalence. □

This relation is important because it reduces a statement over polynomials of degree $\deg(f)$ to ones maximally of degree $\deg(f) - 1$. Now finally the inequality can be tackled:

Theorem 2.14. *For the inequality of two roots of polynomials of known type it holds that:*

$$(f,t,r) \gtrless (g,tg,rg) \Leftrightarrow \bigvee_{dt\in\mathrm{DRT}(t,\deg(f))} \psi_{f,dt} \wedge \begin{cases} (f',t',r') \gtrless (g,tg,rg) & \textit{for } t_r > 1 \\ \varphi & \textit{for } t_r = 1 \end{cases}$$

$$\varphi = \big(f[x\backslash\backslash(g,tg,rg)] \,\tilde{\gtrless}\, 0 \vee (f',t',r'_{\top}) \gtrless (g,tg,rg)\big) \wedge (g,tg,rg) \gtrless (f',t',r'_{\bot})$$

where $\tilde{\gtrless}$ *is the option that contains* $(f,t,r) \gtrless (g,tg,rg)$ *and* $r_{\top}$ *is the closest root from* (f,t,r) *on the satisfied side of the inequality and* $r_{\bot}$ *the closest root on the dissatisfied side. Or, if these don't exist the relations they are contained in are just* $\top$ *and* $\bot$ *respectively.*

Proof. In the case of a multiple root the same argument as in theorem 2.13 applies. For a single root by the decomposition of theorem 2.12 one of the two relations $f[x\backslash\backslash(g,tg,rg)] \,\tilde{\gtrless}\, 0$ contains the desired inequality, along with many others. To extract the right one it is necessary to remove the gaps on the satisfied side with the union and the intervals on the dissatisfied side with the intersection. This works as described because by the root type there is a root of f' in both the satisfied and dissatisfied interval next to (f,t,r) and can be omitted if there are no gaps or undesired intervals because (f,t,r) is already the right- or leftmost root. □

With this the construction of the virtual substitution procedure is finally complete because this relation also reduces a relation of degree $\deg(f)$ to those of only $\deg(f) - 1$.

2.5 Potential Improvements

The version of the virtual substitution algorithm presented above is naive in that it sacrifices algorithmic efficiency in multiple places for simplicity of the algorithm itself. It thereby serves to prove the feasibility of the desired goal but still leaves a lot of room for improvement, some of which will be sketched here.

2.5.1 Root Decomposition Construction

The decomposition of theorem 2.12 gives the expression in terms of the roots that need to be replicated. In the theorems 2.13 and 2.14 this is formulated through the guard on the differential root type, the relation $f[x\backslash\backslash(g,tg,rg)]\,\rho\,0$ and an interval construction.
The differential root type distinction is superfluous for any root type without complex conjugate root pairs and should otherwise be performed ahead of the decomposition of theorem 2.12, potentially combined with that of 2.11, to not needlessly repeat the same formulas for every root. However every equivalent expression is going to need the aforementioned relation as it is the only one to contain the right root. The way this root is extracted and how the algorithm proceeds after that can however be significantly improved, since the above formulation uses no information of prior and potentially posterior recursion steps. What is meant by this can be illustrated through an example:

Example 2.15. For the case $\deg(f) = 3$ and $\deg(g) = 1$ Košta derived the following equivalence:

$$g[x\backslash\backslash(f,(1,1,1),1)] < 0 \Leftrightarrow f[x\backslash\backslash(g,(1),1)] > 0 \vee g[x\backslash\backslash(f',(1,1),1)] < 0$$

By translation to the roots the left hand side is $(f,(1,1,1),1) < (g,(1),1)$ whereas the right hand side is $((f,(1,1,1),1) < (g,(1),1) < (f,(1,1,1),2) \vee (f,(1,1,1),3) < (g,(1),1)) \vee (f',(1,1),1) < (g,(1),1)$. Because it is $(f,(1,1,1),1) < (f',(1,1),1) < (f,(1,1,1),2)$ the equivalence holds.

This way all such relations can be formulated as systems of linear inequalities which means that equivalence can be decided by Fourier Motzkin elimination or any other linear quantifier elimination or optimization method that can handle strict inequalities. Finding a combination of relations with low complexity then becomes a graph search with an equivalence test at every vertex.

2.5.2 Other Ideas

Empirically very few of the many performed case distinctions remain in the formula after simplification. It therefore stands to reason that through heuristics or integration of the steps that the simplification algorithm would otherwise perform many of the case distinctions could be avoided in the first place. Furthermore it might be possible to infer information about the root type of the polynomial resulting from the pseudo remainder algorithm if, like in the case of the recursion steps of the virtual substitution, the root type of f and g are known prior to its application. Additionally for the recursion steps already simplified table entries can be used instead of re-deriving those cases every time as well.

3 Guard and Substitution Tables

Because for any given polynomial degree there are only finitely many relations it makes sense to create a table of their equivalent expressions instead of costly recalculating them every time. With a general algorithm now available exactly this was attempted for the hitherto missing case of degree 4 and will be covered in this section. In pursuit of this end it is naturally useful to know how many cases there are. The number of guards necessary is the same as the number of root types for a given degree and was thereby already derived in theorem 1.6 to be $\Theta(2^n)$. The next question is then what the number of virtual substitutions is for a given degree.

Theorem 3.1. *There are* $(n-1)(2^n(3n+1)-(-1)^n)/3$ *relations* $g[x\backslash\backslash(f,t,r)]\,\rho\,0$ *with* $n = \deg(f) > \deg(g)$.

Proof. The counting is performed in the same way as in theorem 1.6, with the difference that every root is counted, thrice for each of $\rho \in \{<,=,>\}$, $n-1$ times for every degree of g and the empty root type is omitted. This then gives for the number of relations that need to be derived and stored:

$$3(n-1)\sum_{i=0}^{\lfloor\frac{n-1}{2}\rfloor}\sum_{k=1}^{n-1} k\binom{n-2i-1}{k-1} = \frac{1}{3}2^{n-2}(n-1)\left(4^{\lceil\frac{1}{2}-\frac{n}{2}\rceil}\left(6\left\lfloor\frac{n-1}{2}\right\rfloor-3n+5\right)+12n+4\right)$$

Here the right hand side was calculated with Mathematica. Once again performing a case distinction on the parity of n, simplifying individually and then merging gives the claimed formula. □

Consequently the number of entries of such a table grows as $\Theta(n^2 2^n)$. For small n it is:

n	1	2	3	4	5	6	7	8	9	10
guards	1	3	5	11	21	43	85	171	341	683
vs	0	9	54	207	684	2025	5634	14931	38232	95229

Thus the number of entries quickly becomes prohibitive for dealing with each case by hand but stays manageable for computers for small degrees. The significant obstacle instead is the size and complexity of the formulas, especially with the humongous number of case distinctions the algorithm performs. One way of tackling this is by trying to simplify the expressions, which is a hard problem in itself.

3.1 Simplification

The problem of what constitutes a simple expression is generally ill defined because there are many different measures for complexity, such as term count, formula depth, polynomial degree and coefficient size. Furthermore choices about whether normal forms are maintained and which logical operators and comparators are utilized have significant impact. For example $\prod_{i\in I} f_i > 0$ can be factorized as $\neg\bigoplus_{i\in I} f_i > 0$ with the xor operator, but any representation with only $\wedge, \vee$ and $\neg$ needs exponentially many terms in the size of I.
Generally research has focused on heuristic approaches to formula simplification to remain practical without attempts to find "optimal" representations. Because of their importance to the quantifier elimination algorithm presented here and the table generation some of these are covered next.

Factorization and Monomial Inequalities

In [2] it was shown how expressions whose atomic formulas can be factorized and contain common factors can be reduced to a simpler form. This is done by treating every factor as a monomial in a new variable and then developing algorithms to simplify monomial inequalities and discovering implications of them. In this thesis formulas are mostly factorized to keep the degree low, even though it is debatable whether the representation $(a > 0 \wedge b > 0) \vee (a < 0 \wedge b < 0)$ is simpler than $ab > 0$.

Mathematica

In [5] a combination of factorization, optimization on the boolean structure and deduction are used for simplification and their positive impact on quantifier elimination methods is noted. This is relevant because it was implemented in the Mathematica kernel and thereby used extensively here. Although doubtlessly Mathematicas simplification algorithm has evolved significantly since then.

QEPCAD

The idea of employing cylindrical algebraic decompositions is explored among others in [12] and [4]. The principle therein is to simplify the CAD, which is an easier problem, and transform the result back into semi-algebraic form. The SLFQ program associated with QEPCAD did not end up being used here since it tends to give answers in terms of extended Tarski formulas, still as roots of polynomials, instead of semi-algebraically. Calling QEPCAD directly however, without quantifiers, also gives simpler formulas and this has been used extensively through the aforementioned implementation of a primitive interface.

Radical Ideals and Reductions

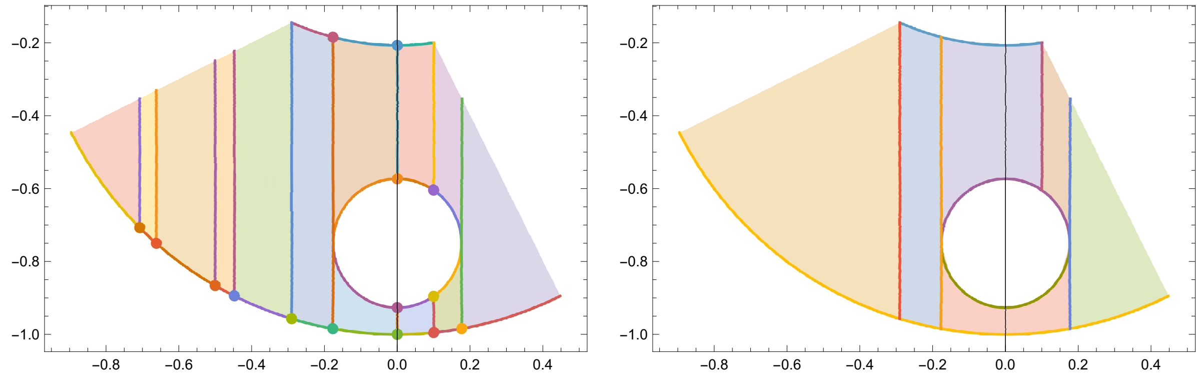
When a formula contains a conjunction of equalities $f_i = 0$ the resulting semi-algebraic set is a subset of the corresponding algebraic variety. From algebraic geometry it is known that the associated ideal is the radical ideal $\sqrt{\langle f_i \rangle} := \{f \in \mathbb{Z}[\mathbf{y}][x] | \exists n \in \mathbb{N} : f^n \in \langle f_i \rangle\}$. The calculation of this ideal is an intricate topic in itself [17] but through it simpler representations of the equations can be found. For example $f^2 + g^2 = 0$ would be simplified to $f = 0 \wedge g = 0$. Additionally the remaining atomic formulas can be simplified modulo $\sqrt{\langle f_i \rangle}$ through Gröbner basis techniques. Incidentally this has been used for some of the guards of degree 4.

Decision Procedures

Finally there are optimization ideas based on solving decision problems. Such are presented in [9] together with those already covered. Specifically these are trying to find (semi-)positive-definite polynomials to reduce a relation and implications $f \Rightarrow g$ to reduce $f \vee g$ to g. In the paper calculating the second case directly was considered computationally excessive due to their focus on larger formulas and instead handled heuristically. For the final stages of the table formula simplification their conversion to DNF and successive elimination of both entire conjunctions and atomic formulas by checking equivalence using Mathematicas Reduce was however very successful and effective, once the formula was small enough for this technique to be applicable.

Construction Ideas

Similar techniques for the simplification of CADs have been applied to the output of Mathematicas CylindricalAlgebraicDecomposition command here. Naturally that output is already a CAD but might be split into too many cells, as is the case in figure (f). By giving up on the cylindricality of the decomposition the reduction step becomes relatively easy. Two cells can then be merged if they are adjacent, have the same dimension and the intersection of their closures is a cell as well. Applying this in ascending order of dimension of the cells gives a decomposition without redundant splits, for example that of figure (g).



(f) Direkt Output

(g) Reduced Form

Figure 3: One component of the example from `https://mathematica.stackexchange.com/questions/67116/decomposition-of-a-semialgebraic-set-into-connected-components`

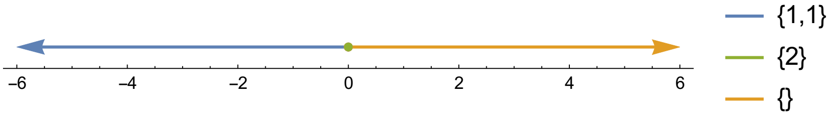
From this representation it should be possible to construct a semi-algebraic representation with a minimal amount of disjunctions. The idea being that at least every polynomial that defines a boundary needs to be included and two polynomials where one cuts off a relevant region from the other need to be in separate components. Additionally polynomials that don't themselves contribute to the boundary but instead cut off undesired parts of polynomials that do would need to be introduced. These are also generally not uniquely determined.
Working out the details of this with the complexity of connected components and even just deciding which polynomials contribute to the boundary however proved beyond the scope of this thesis. Indeed the properties of such normal form representations of semi-algebraic sets are subject to active research such as[1].

3.2 Table Generation

To derive the concrete formulas for the guards and substitutions the procedure of section 2 was implemented in Mathematica. Additionally an elementary interface to QEPCAD was implemented and Mathematicas Simplify and Reduce function were used extensively to simplify the resulting formulas. The corresponding Mathematica notebooks and other digital resources related to this are hosted at[1] or available from the author upon request.

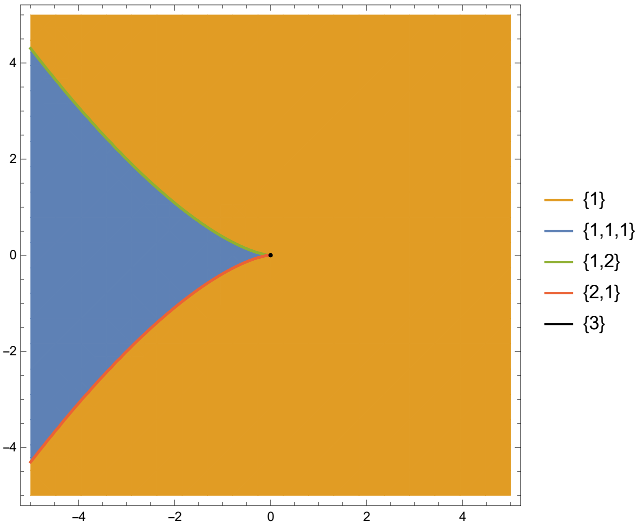
3.2.1 Guards

As shown in theorem 2.6 it is sufficient to have a table for the guards of monic depressed polynomials because the Tschirnhaus transformation can be applied, the condition looked up from the table and then reversed again. Consequently a polynomial of degree n has $n-1$ parameters and its guards partition $\mathbb{R}^{n-1}$.
The well known degree 2 case is just the sign of the discriminant:



$$\begin{aligned}\gamma_{(1,1)}(x^2+p) &\Leftrightarrow p<0\\ \gamma_{(2)}(x^2+p) &\Leftrightarrow p=0\\ \gamma_{()}(x^2+p) &\Leftrightarrow p>0\end{aligned}$$

For degree 3 the picture starts to becomes more complicated and begins to illustrate how every multiple root decreases the dimension of the parameter space by one.

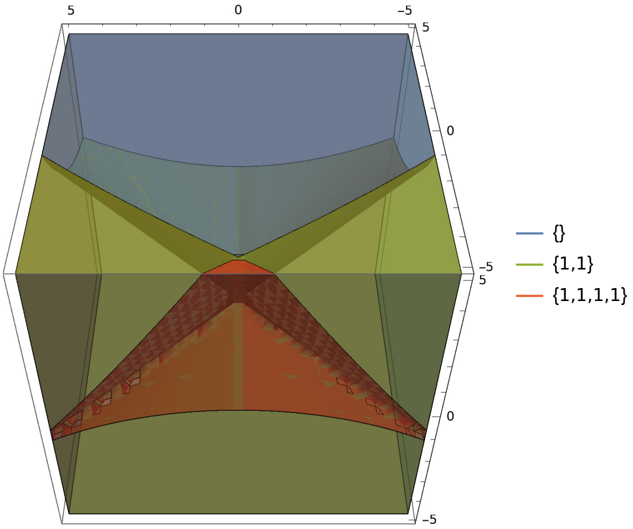


$$\begin{aligned}\gamma_{(1)}(x^3+px+q) &\Leftrightarrow 4p^3+27q^2>0\\ \gamma_{(1,1,1)}(x^3+px+q) &\Leftrightarrow 4p^3+27q^2<0\\ \gamma_{(1,2)}(x^3+px+q) &\Leftrightarrow 4p^3+27q^2=0\wedge q<0\\ \gamma_{(2,1)}(x^3+px+q) &\Leftrightarrow 4p^3+27q^2=0\wedge q>0\\ \gamma_{(3)}(x^3+px+q) &\Leftrightarrow p=0\wedge q=0\end{aligned}$$

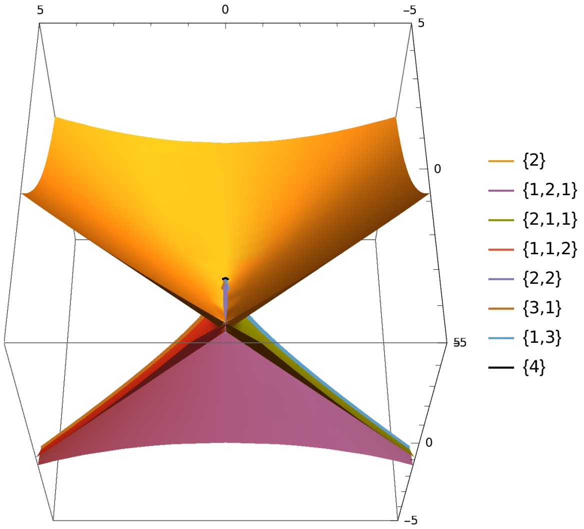
Finally for degree 4 the visualization becomes three dimensional. Therefore the full dimensional components are portrayed separately for better visibility. Also the discriminant will be abbreviated as

$$\Delta = 16p^4r-4p^3q^2-128p^2r^2+144pq^2r-27q^4+256r^3$$

[1] https://github.com/Athlici/RealQuantifierElimination



(a) Full-dimensional components



(b) Sub-dimensional components

$$\begin{aligned}
\gamma_{()}(x^4+px^2+qx+r) &\Leftrightarrow (\Delta > 0 \land (p > 0 \lor p^2-4r < 0)) \lor (q = 0 \land p^2 - 4r = 0 \land p > 0)\\
\gamma_{(1,1)}(x^4+px^2+qx+r) &\Leftrightarrow \Delta < 0\\
\gamma_{(1,1,1,1)}(x^4+px^2+qx+r) &\Leftrightarrow \Delta > 0 \land p < 0 \land p^2 - 4r > 0\\
\gamma_{(2)}(x^4+px^2+qx+r) &\Leftrightarrow \Delta = 0 \land (p^2-4r > 0 \lor (p > 0 \land (4r - p^2 < 0 \lor q \neq 0))\\
\gamma_{(2,1,1)}(x^4+px^2+qx+r) &\Leftrightarrow \Delta = 0 \land p < 0 \land q < 0 \land p^4 - 16p^2r + 9pq^2 + 48r^2 < 0\\
\gamma_{(1,2,1)}(x^4+px^2+qx+r) &\Leftrightarrow \Delta = 0 \land p < 0 \land p^2 - 6r > 0 \land p^4 - 16p^2r + 9pq^2 + 48r^2 > 0\\
\gamma_{(1,1,2)}(x^4+px^2+qx+r) &\Leftrightarrow \Delta = 0 \land p < 0 \land q > 0 \land p^4 - 16p^2r + 9pq^2 + 48r^2 < 0\\
\gamma_{(2,2)}(x^4+px^2+qx+r) &\Leftrightarrow q = 0 \land p^2 - 4r = 0 \land p < 0\\
\gamma_{(3,1)}(x^4+px^2+qx+r) &\Leftrightarrow 8p^3 + 27q^2 = 0 \land p^2 + 12r = 0 \land p < 0 \land q > 0\\
\gamma_{(1,3)}(x^4+px^2+qx+r) &\Leftrightarrow 8p^3 + 27q^2 = 0 \land p^2 + 12r = 0 \land p < 0 \land q < 0\\
\gamma_{(4)}(x^4+px^2+qx+r) &\Leftrightarrow p = 0 \land q = 0 \land r = 0
\end{aligned}$$

No attempt has been made to derive the guards for degree 5 because of the upcoming difficulties for substitutions of degree 4, upon which the guards would rely. Furthermore the result would be hard to visualize here.

3.2.2 Substitutions

For the substitution tables the number of parameters that are needed can be reduced by 1 compared to theorem 2.10 by making g monic as well. For a general relation $g[x\backslash\backslash(f,x,r)]\,\rho\,0$ with $\mathrm{lc}(f) > 0$ and $\mathrm{lc}(g) > 0$ as well as $n = \deg(f)$ and $m = \deg(g)$, first the Tschirnhaus transformation is applied to f to make it monic and depressed. This however changes x, which must be propagated to g, giving:

$$\overline{g} := \mathrm{Tsch}_f(g) = (n\,\mathrm{lc}(f))^m g\left(\frac{x - c_{f,n-1}}{n\,\mathrm{lc}(f)}\right) = \sum_{i=0}^{m} c_{g,i}(n\,\mathrm{lc}(f))^{m-i}(x - c_{f,n-1})^i$$

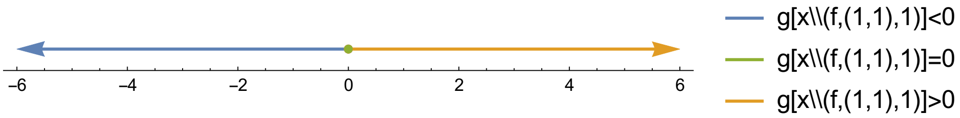
However it still is $\mathrm{lc}(\overline{g}) = \mathrm{lc}(g) > 0$ and so $\mathrm{lc}(\overline{g})c_{\tilde{g},i} = c_{\overline{g},i}$ can be defined for a monic polynomial $\tilde{g}$ such that

$$\overline{g}\,\rho\,0 \Leftrightarrow \mathrm{lc}(\overline{g})\tilde{g}\,\rho\,0 \Leftrightarrow \tilde{g}\,\rho\,0$$

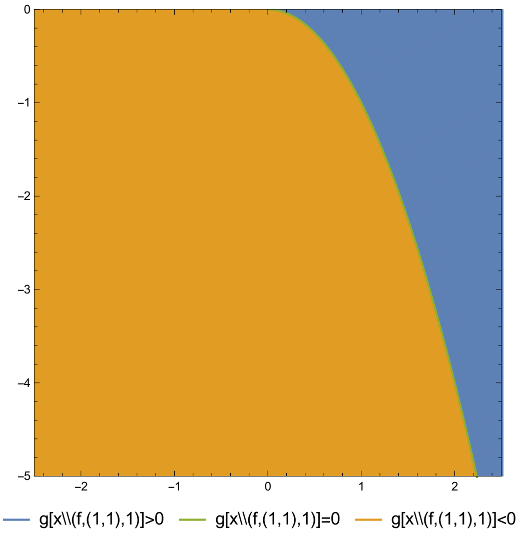
The corresponding formula from the table can then be used and the above transformations reversed.
For degree 2 there are only 9 cases and with the above reduction three are only two parameters p and a for the polynomials $f = x^2 + p$ and $g = x + a$. The sign at every root then partitions its guard type and this can still

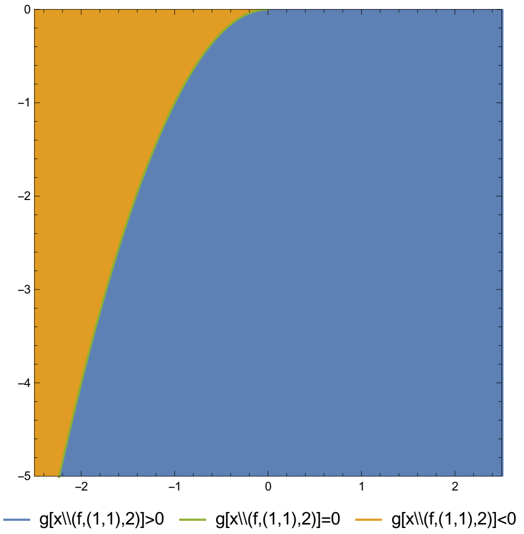
be visualized pretty easily:

$$g[x\backslash\backslash(f,(2),1)] > 0 \Leftrightarrow a > 0 \qquad g[x\backslash\backslash(f,(2),1)] = 0 \Leftrightarrow a = 0 \qquad g[x\backslash\backslash(f,(2),1)] < 0 \Leftrightarrow a < 0$$



$$\begin{aligned} g[x\backslash\backslash(f,(1,1),1)] > 0 &\Leftrightarrow a^2 + p > 0 \wedge a > 0 \\ g[x\backslash\backslash(f,(1,1),1)] = 0 &\Leftrightarrow a^2 + p = 0 \vee a < 0 \\ g[x\backslash\backslash(f,(1,1),1)] < 0 &\Leftrightarrow a^2 + p < 0 \wedge a < 0 \end{aligned} \qquad \begin{aligned} g[x\backslash\backslash(f,(1,1),2)] > 0 &\Leftrightarrow a^2 + p < 0 \vee a > 0 \\ g[x\backslash\backslash(f,(1,1),2)] = 0 &\Leftrightarrow a^2 + p = 0 \wedge a < 0 \\ g[x\backslash\backslash(f,(1,1),2)] < 0 &\Leftrightarrow a^2 + p > 0 \wedge a < 0 \end{aligned}$$





For degree 3 there are 54 cases and most of them could only be visualized in 3− or 4−dimensional space. Thus, to preserve readability, the many derived formulas were delegated to the appendix. These have been extensively simplified semi-manually with the techniques of 3.1 but an obviously "simplest" form could not generally be found.

For degree 4 all the substitution formulas could be calculated using the implementation of the naive algorithm but in part ended up being as big as $\approx$ 1GB in Mathematicas formula representation. Subsequent simplification generally managed to reduce the formula size to multiple pages at which point none of the algorithms available to the author made progress anymore or were applicable. Therefore for degree 4 only those formulas are included in the appendix for which a reasonably simple form could be found.

4 Applications

The two applications presented here are specifically well suited to the virtual substitution approach. They also show how, even though the algorithm is limited by complexity in terms of the degrees, variable counts and formula sizes it can handle, this can still be sufficient for practical purposes.

4.1 Positivity Conditions

In the past some attention has been devoted to deriving the conditions under which polynomials of certain degree are positive, either on an interval or globally. The global case is given directly by the guard $\gamma_{()}(f)$ for f of even degree. This way a problem that required significant manual effort previously, such as the derivation of global positivity for degree 4 in [18], is reduced to a purely algorithmic one. This can even be applied successively to multivariate polynomials, although the degree might grow prohibitively. The case of positivity on an open interval (a, b) can be reduced to $(0, \infty)$ through the transformation

$$f \mapsto (1+x)^{\deg(f)} f\left(\frac{a+bx}{1+x}\right)$$

as done in [23]. By deriving the conditions for this case those for closed intervals follow directly by adding point evaluations such as $f(a)$. The remaining problem $P_\rho = \forall x\, x \leq 0 \vee f(x)\, \rho\, 0$ for $\rho \in \{>, \geq\}$ can now be handled very well with the techniques developed before. The reason is that $P_>$ is equivalent to the right-most root of f being non-positive, which is $x[x\backslash\backslash(f,t,r)] \leq 0$ and $P_\geq$ is equivalent to the right-most odd root of f being non-positive which is $x[x\backslash\backslash(f,t,\tilde{r})] \leq 0$. The case $\mathrm{lc}(f) < 0$ is trivially false and $\mathrm{lc}(f) = 0$ degenerates to lower degree. Thus only $\mathrm{lc}(f) > 0$ is relevant and the usual case distinction on the root type of f gives the desired conditions.

$$P_> = \bigvee_{t \in \mathrm{RT}(\deg(f))} \gamma_t(f) \wedge \neg x[x\backslash\backslash(f,t,r)] > 0 \qquad P_\geq = \bigvee_{t \in \mathrm{RT}(\deg(f))} \gamma_t(f) \wedge \neg x[x\backslash\backslash(f,t,\tilde{r})] > 0$$

Applying the Tschirnhaus transformation represents the problem with $x^2 + p, x^3 + px + q$ and $x^4 + px^2 + qx + r$ for the different degrees of f and $g = x + a$. Simplified versions of the conditions then are:

$$\begin{aligned}
&P_>^{(2)} = a^2 + p \geq 0 \wedge (a \leq 0 \vee p > 0) \qquad P_\geq^{(2)} = a^2 + p \geq 0 \wedge (a < 0 \vee p \geq 0)\\
&P_>^{(3)} = q \geq a^3 + ap \wedge \left(4p^3 + 27q^2 > 0 \vee \left(a \leq 0 \wedge 3a^2 + p \geq 0\right)\right)\\
&P_\geq^{(3)} = a^3 + ap \leq q \wedge \left(\left(3a^2 + p \geq 0 \wedge a \leq 0\right) \vee \left(4p^3 + 27q^2 \geq 0 \wedge \left(a^3 + ap < q \vee 3a^2 + p > 0\right)\right)\right)
\end{aligned}$$

4.2 Minkowski Sum

A useful construction in geometry is that of the Minkowski sum $A + B := \{a + b | a \in A, b \in B\}$ for two sets embedded in $\mathbb{R}^n$. So naturally the question arises whether semi-algebraic sets are closed under this operation as well and how it can be calculated. This is answered in the affirmative by formulating it as a quantifier elimination problem.

Theorem 4.1. *For the Minkowski sum of two semialebraic sets $A, B \in \mathcal{SA}_n$ it holds that*

$$\{x \mid \exists a \in A : x - a \in B\} = \{x \mid \exists b \in B : x - b \in A\} = A + B \in \mathcal{SA}_n$$

Proof. The elements of $A + B$ are those $x \in \mathbb{R}^n$ such that $a \in A$ and $b \in B$ exist with $x = a + b$. Conversely $a = x - b$ and $b = x - a$ can be determined that way and the sets are semi-algebraic because of theorem 1.14. □

This construction is especially useful in operations research for encoding geometric no overlap conditions. These are employed in finding dense arrangements for irregular objects, for example for packing objects or cutting shapes while minimizing material wasted. More specifically it will be shown here how 2D problems with semi-algebraic shapes subject to translation rotation and scaling can be formulated as a polynomial programming problem. Ordinarily this is handled by discretizing the geometry to polygons and if rotations are involved selecting a finite amount of them and then performing overlap tests either with standard polygon algorithms or via the Minkowski sum for that case, as in [28]. Placement is then performed heuristically and largely this gives good results. To achieve higher packing densities for small amounts of objects, especially under rotations, continuous models have been proposed, as in [6] where quadratic formulas are derived for combinations of line and circle segments. These are a special case of a parametric Minkowski sum:

Corollary 4.2. *Let $A, B \in \mathcal{SA}_2$ and f_a, f_b be a scaling or translation and rotation:*

$$\sigma \times (x, y) \mapsto (\sigma_r * x, \sigma_r * y) \qquad \sigma \times (x, y) \mapsto (\sigma_x + \sigma_c * x - \sigma_s * y, \sigma_y + \sigma_s * x + \sigma_c * y)$$

Then the sets $f_a(\sigma a, A)$ and $f_b(\sigma b, B)$ intersect iff $\exists x\, \exists y\, (x, y) \in f_a(\sigma a, A) \wedge (x, y) \in f_b(\sigma b, B)$.

The rotation herein is a polynomial condition because the sine and cosine lie on the unit circle, $\sigma_s^2 + \sigma_c^2 = 1$. Adding this constraint for every object subject to rotation and the quantifier free equivalent of the negation of the intersection condition for every pair of objects then gives the desired polynomial program.
By restricting the design space from general polynomials to rational functions $\mathbb{R} \to \mathbb{R}^2$ this can even be reduced

to eliminating just one quantifier. A result from algebraic geometry is that every rational function has an implicit form $\mathbb{R}^2 \to \mathbb{R}$ that can be found with Gröbner basis techniques. From this a rational distance function $\mathbb{R} \to \mathbb{R}$ from a segment of the boundary of one of the sets to the other can then be defined by concatenation. Because two closed two dimensional sets intersect iff $\partial A \cap B \neq \emptyset$ or for a $b \in \overset{\circ}{B}$ it is $b \in A$ checking intersection can essentially be reduced to calculating the minimum distance of a point on the border of A to B. Treating every segment separately and then taking the minimum over them results in resolving a bunch of formulas of the form $\exists t \in I_\phi \wedge \Phi_b(\sigma b, \phi(\sigma a, t)) \leq 0$ where Φ is the aforementioned implicit distance function of B and ϕ is the parametrization mapping I_ϕ to a segment of the boundary of A. This is just a more complicated form of positivity on an interval, as encountered in the previous application. The biggest problem with this approach is that it contains polynomials of degree $(\max_{f \in A} \deg(f))(\max_{f \in B} \deg(f))$ and so requires the degree 4 substitution tables already in the case of quadratic rational functions.

5 Conclusion

Due to time limits many properties and extensions of the generalization of the virtual substitution method presented here are still open for exploration. Aside from those already mentioned most importantly a derivation of the overall asymptotic complexity is still missing. While the number of cases could be derived exactly for the root types similar bounds are missing for the size of the formula, coefficients and how the algorithm behaves with multiple quantifiers. Furthermore it could be interesting to use the virtual substitution algorithm for decision or optimization problems since in that context every subformula can itself be simplified directly to $\top$ or $\bot$, potentially keeping the amount of memory required low.

Additionally, with a tuned algorithm and a better understanding of formula simplification the formulas for degree 4 could surely be put into a practically useful form and integrated into REDLOG.

6 References

[1] Gennadiy Averkov. *Representing elementary semi-algebraic sets by a few polynomial inequalities: A constructive approach.* 2008. arXiv: 0804.2134 [math.AG].

[2] Christopher W. Brown. "Fast simplifications for Tarski formulas based on monomial inequalities". In: *Journal of Symbolic Computation* 47.7 (2012). International Symposium on Symbolic and Algebraic Computation (ISSAC 2009), pp. 859–882. URL: http://www.sciencedirect.com/science/article/pii/S0747717111002045.

[3] Christopher W. Brown. "QEPCAD B: A Program for Computing with Semi-algebraic Sets Using CADs". In: *SIGSAM Bull.* 37.4 (Dec. 2003), pp. 97–108. URL: http://doi.acm.org/10.1145/968708.968710.

[4] Christopher W. Brown. "Simple CAD Construction and its Applications". In: *Journal of Symbolic Computation* 31.5 (2001), pp. 521–547. URL: http://www.sciencedirect.com/science/article/pii/S0747717100903948.

[5] Christopher W. Brown and Adam Strzeboński. "Black-box/White-box Simplification and Applications to Quantifier Elimination". In: *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation.* ISSAC '10. Munich, Germany: ACM, 2010, pp. 69–76. URL: http://doi.acm.org/10.1145/1837934.1837953.

[6] N. Chernov et al. "Phi-Functions for 2D Objects Formed by Line Segments and Circular Arcs". In: *Advances in Operations Research* 2012 (May 2012).

[7] George E. Collins. "Quantifier elimination for real closed fields by cylindrical algebraic decompostion". In: *Automata Theory and Formal Languages 2nd GI Conference Kaiserslautern, May 20–23, 1975.* Ed. by H. Brakhage. Berlin, Heidelberg: Springer Berlin Heidelberg, 1975, pp. 134–183.

[8] Michel Coste. "An Introduction to Semialgebraic Geometry". In: (2002). URL: http://gcomte.perso.math.cnrs.fr/M2/CosteIntroToSemialGeo.pdf.

[9] ANDREAS DOLZMANN and THOMAS STURM. "Simplification of Quantifier-free Formulae over Ordered Fields". In: *Journal of Symbolic Computation* 24.2 (1997), pp. 209–232. URL: http://www.sciencedirect.com/science/article/pii/S0747717197901231.

[10] Ryoya Fukasaku, Hidenao Iwane, and Yosuke Sato. "Real Quantifier Elimination by Computation of Comprehensive GröBner Systems". In: *Proceedings of the 2015 ACM on International Symposium on Symbolic and Algebraic Computation.* ISSAC '15. Bath, United Kingdom: ACM, 2015, pp. 173–180. URL: http://doi.acm.org/10.1145/2755996.2756646.

[11] Hoon Hong. *Comparison of Several Decision Algorithms for the Existential Theory of the Reals.* Tech. rep. 1991.

[12] Hoon Hong. "Simple Solution Formula Construction in Cylindrical Algebraic Decomposition Based Quantifier Elimination". In: *Papers from the International Symposium on Symbolic and Algebraic Computation.* ISSAC '92. Berkeley, California, USA: ACM, 1992, pp. 177–188. URL: http://doi.acm.org/10.1145/143242.143306.

[13] Wolfram Research Inc. *Mathematica, Version 12.0.* Champaign, IL, 2019. URL: https://www.wolfram.com/mathematica.

[14] Marek Košta. "New concepts for real quantifier elimination by virtual substitution". In: (2016). URL: http://www.ui.sav.sk/home/mkosta/mkosta-phd-thesis.pdf.

[15] Marek Kosta and Thomas Sturm. "A Generalized Framework for Virtual Substitution". In: *CoRR* abs/1501.05826 (2015). arXiv: 1501.05826. URL: http://arxiv.org/abs/1501.05826.

[16] Salma Kuhlmann. "Real Algebraic Geometry Lecture Notes". In: (2009). URL: http://www.math.uni-konstanz.de/algebra/WS0910/Notes11.pdf.

[17] Santiago Laplange. "The computation of the radical of an ideal". In: (2006). URL: http://cms.dm.uba.ar/Members/slaplagn/archivos/linz_2006-02.pdf.

[18] Daniel Lazard. "Quantifier elimination: Optimal solution for two classical examples". In: *Journal of Symbolic Computation* 5.1 (1988), pp. 261–266. URL: http://www.sciencedirect.com/science/article/pii/S0747717188800154.

[19] Songxin Liang and David J. Jeffrey. “Automatic computation of the complete root classification for a parametric polynomial”. In: *Journal of Symbolic Computation* 44.10 (2009), pp. 1487–1501. URL: http://www.sciencedirect.com/science/article/pii/S0747717109001047.

[20] Daniel Perrucci and Marie-Françoise Roy. *Elementary recursive quantifier elimination based on Thom encoding and sign determination.* 2016. arXiv: 1609.02879 [math.AG].

[21] André Platzer. “Lecture Notes on Virtual Substitution and Real Equations”. In: (2016). URL: http://symbolaris.com/course/fcps16/20-virteq.pdf.

[22] E. L. Rees. “Graphical Discussion of the Roots of a Quartic Equation”. In: *The American Mathematical Monthly* 29.2 (1922), pp. 51–55. URL: http://www.jstor.org/stable/2972804.

[23] Gary. Ulrich and Layne T. Watson. “Positivity Conditions for Quartic Polynomials”. In: *SIAM Journal on Scientific Computing* 15.3 (1994), pp. 528–544. eprint: https://doi.org/10.1137/0915035. URL: https://doi.org/10.1137/0915035.

[24] V. Weispfenning. “A New Approach to Quantifier Elimination for Real Algebra”. In: *Quantifier Elimination and Cylindrical Algebraic Decomposition.* Ed. by Bob F. Caviness and Jeremy R. Johnson. Vienna: Springer Vienna, 1998, pp. 376–392.

[25] Volker Weispfenning. “Quantifier Elimination for Real Algebra — the Cubic Case”. In: ISSAC ’94 (1994), pp. 258–263. URL: http://doi.acm.org/10.1145/190347.190425.

[26] Volker Weispfenning. “Quantifier Elimination for Real Algebra — the Quadratic Case and Beyond”. In: *Applicable Algebra in Engineering, Communication and Computing* 8 (1997), pp. 85–101. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.52.4480&rep=rep1&type=pdf.

[27] Volker Weispfenning. “The complexity of linear problems in fields”. In: *Journal of Symbolic Computation* 5.1 (1988), pp. 3–27. URL: http://www.sciencedirect.com/science/article/pii/S0747717188800038.

[28] Sheng Xie, Gary Wang, and Y. Liu. “Nesting of two-dimensional irregular parts: An integrated approach”. In: *Int. J. Computer Integrated Manufacturing* 20 (Dec. 2007), pp. 741–756.

7 Appendix

$f = x^3 + px + q$ and $g = x + a$

$$g[x\backslash\backslash(f,(3),1)] > 0 \Leftrightarrow a > 0 \qquad g[x\backslash\backslash(f,(3),1)] = 0 \Leftrightarrow a = 0 \qquad g[x\backslash\backslash(f,(3),1)] < 0 \Leftrightarrow a < 0$$

$$g[x\backslash\backslash(f,(2,1),1)] > 0 \Leftrightarrow 3a^2 + p > 0 \wedge a > 0 \qquad g[x\backslash\backslash(f,(2,1),2)] > 0 \Leftrightarrow 3a > 0 \vee a^3 + ap - q > 0$$

$$g[x\backslash\backslash(f,(2,1),1)] = 0 \Leftrightarrow 3a^2 + p = 0 \wedge a > 0 \qquad g[x\backslash\backslash(f,(2,1),2)] = 0 \Leftrightarrow a^3 + ap = q \wedge 3a^2 + p > 0$$

$$g[x\backslash\backslash(f,(2,1),1)] < 0 \Leftrightarrow 3a^2 + p > 0 \vee a < 0 \qquad g[x\backslash\backslash(f,(2,1),2)] < 0 \Leftrightarrow q > a^3 + ap$$

$$g[x\backslash\backslash(f,(1,2),1)] > 0 \Leftrightarrow q < a^3 + ap \qquad g[x\backslash\backslash(f,(1,2),2)] > 0 \Leftrightarrow 3a^2 + p < 0 \vee a > 0$$

$$g[x\backslash\backslash(f,(1,2),1)] = 0 \Leftrightarrow a^3 + ap = q \wedge 3a^2 + p > 0 \qquad g[x\backslash\backslash(f,(1,2),2)] = 0 \Leftrightarrow 3a^2 + p = 0 \wedge a < 0$$

$$g[x\backslash\backslash(f,(1,2),1)] < 0 \Leftrightarrow a^3 + ap - q < 0 \vee a < 0 \qquad g[x\backslash\backslash(f,(1,2),2)] < 0 \Leftrightarrow 3a^2 + p > 0 \wedge a < 0$$

$$g[x\backslash\backslash(f,(1,1,1),1)] > 0 \Leftrightarrow q < a^3 + ap$$

$$g[x\backslash\backslash(f,(1,1,1),1)] = 0 \Leftrightarrow a^3 + ap = q \wedge 3a^2 + p > 0$$

$$g[x\backslash\backslash(f,(1,1,1),1)] < 0 \Leftrightarrow q > a^3 + ap \vee 3a^2 + p < 0 \vee a < 0$$

$$g[x\backslash\backslash(f,(1,1,1),2)] > 0 \Leftrightarrow (3a^2 + p \geq 0 \wedge a > 0) \vee (a^3 + ap - q < 0 \wedge 3a^2 + p < 0)$$

$$g[x\backslash\backslash(f,(1,1,1),2)] = 0 \Leftrightarrow a^3 + ap = q \wedge 3a^2 + p < 0$$

$$g[x\backslash\backslash(f,(1,1,1),2)] < 0 \Leftrightarrow (a^3 + ap - q > 0 \wedge 3a^2 + p \leq 0) \vee (3a^2 + p > 0 \wedge a < 0)$$

$$g[x\backslash\backslash(f,(1,1,1),3)] > 0 \Leftrightarrow q < a^3 + ap \vee 3a^2 + p < 0 \vee a > 0$$

$$g[x\backslash\backslash(f,(1,1,1),3)] = 0 \Leftrightarrow a^3 + ap = q \wedge 3a^2 + p > 0 \wedge a < 0$$

$$g[x\backslash\backslash(f,(1,1,1),3)] < 0 \Leftrightarrow q > a^3 + ap \wedge 3a^2 + p > 0 \wedge a < 0$$

$$g[x\backslash\backslash(f,(1),1)] > 0 \Leftrightarrow a^3 + ap - q > 0 \quad g[x\backslash\backslash(f,(1),1)] = 0 \Leftrightarrow a^3 + ap = q \wedge 3a^2 + p > 0 \quad g[x\backslash\backslash(f,(1),1)] < 0 \Leftrightarrow a^3 + ap - q < 0$$

$f = x^3 + px + q$ and $g = x^2 + ax + b$

$$g[x\backslash\backslash(f,(3),1)] > 0 \Leftrightarrow b > 0 \qquad g[x\backslash\backslash(f,(3),1)] = 0 \Leftrightarrow b = 0 \qquad g[x\backslash\backslash(f,(3),1)] < 0 \Leftrightarrow b < 0$$

$$g[x\backslash\backslash(f,(2,1),1)] > 0 \Leftrightarrow \left(a < 0 \wedge \left(p < 3b \vee 3a^2p + (p-3b)^2 < 0\right)\right) \vee \left(p < 3b \wedge 3a^2p + (p-3b)^2 > 0\right)$$

$$g[x\backslash\backslash(f,(2,1),1)] = 0 \Leftrightarrow 3a^2p + (p-3b)^2 = 0 \wedge (a \geq 0 \vee b < 0) \wedge \left(a \geq 0 \vee 3a^2 + 2p > 6b\right) \wedge$$
$$\left(a < 0 \vee b > 0 \vee 3a^2 + 2p \leq 6b\right) \wedge p < 0 \wedge 4b \leq a^2$$

$$g[x\backslash\backslash(f,(2,1),1)] < 0 \Leftrightarrow \left(a > 0 \wedge \left(3a^2p + (p-3b)^2 < 0 \vee p > 3b\right)\right) \vee \left(p > 3b \wedge 3a^2p + (p-3b)^2 > 0\right)$$

$$g[x\backslash\backslash(f,(2,1),2)] > 0 \Leftrightarrow a^2 + p < b \vee b\left(a^2p + (b-p)^2\right) + q^2 > aq\left(a^2 - 3b + p\right) \vee (a > 0 \wedge b > 0)$$

$$g[x\backslash\backslash(f,(2,1),2)] = 0 \Leftrightarrow b\left(a^2p + (b-p)^2\right) + q^2 = aq\left(a^2 - 3b + p\right) \wedge 4b \leq a^2 \wedge (3a^2p + (p-3b)^2 > 0 \vee$$
$$\left(a < 0 \wedge \left(\left(3a^2p + (p-3b)^2 = 0 \wedge \left(b = -2a^2 \vee 2a^2 = 9b\right)\right) \vee \left(-2a^2 < b \wedge 9b < 2a^2 \wedge 3a^2p + (p-3b)^2 < 0\right)\right)\right))$$

$$g[x\backslash\backslash(f,(2,1),2)] < 0 \Leftrightarrow b\left(a^2p + (b-p)^2\right) + q^2 < aq\left(a^2 - 3b + p\right) \vee \left(b < 0 \wedge 2q > a\left(a^2 - 3b + p\right)\right)$$

$$g[x\backslash\backslash(f,(1,2),1)] > 0 \Leftrightarrow a^2 + p < b \vee b\left(a^2p + (b-p)^2\right) + q^2 > aq\left(a^2 - 3b + p\right) \vee (a < 0 \wedge b > 0)$$

$$g[x\backslash\backslash(f,(1,2),1)] = 0 \Leftrightarrow b\left(a^2p + (b-p)^2\right) + q^2 = aq\left(a^2 - 3b + p\right) \wedge$$
$$\left(\left(\left(a^2 + p \geq b \vee 2q < a\left(a^2 - 3b + p\right)\right) \wedge \left(a^2 + p < b \vee 2q \geq a\left(a^2 - 3b + p\right)\right) \wedge a > 0\right) \vee 3a^2p + (p-3b)^2 > 0\right)$$

$$g[x\backslash\backslash(f,(1,2),1)] < 0 \Leftrightarrow b\left(a^2p + (b-p)^2\right) + q^2 < aq\left(a^2 - 3b + p\right) \vee \left(b < 0 \wedge 2q < a\left(a^2 - 3b + p\right)\right)$$

$$g[x\backslash\backslash(f,(1,2),2)] > 0 \Leftrightarrow \left(p < 3b \wedge \left(a > 0 \vee 3a^2p + (p-3b)^2 > 0\right)\right) \vee \left(3a^2p + (p-3b)^2 < 0 \wedge a > 0\right)$$

$$g[x\backslash\backslash(f,(1,2),2)] = 0 \Leftrightarrow 3a^2p + (p-3b)^2 = 0 \wedge \left(\left(a < 0 \wedge \left(3a^2 + 2p < 6b \vee b > 0\right)\right) \vee \left(a \geq 0 \wedge 3a^2 + 2p \geq 6b \wedge b < 0\right)\right)$$

$$g[x\backslash\backslash(f,(1,2),2)] < 0 \Leftrightarrow \left(a < 0 \wedge \left(3a^2p + (p-3b)^2 < 0 \vee p > 3b\right)\right) \vee \left(p > 3b \wedge 3a^2p + (p-3b)^2 > 0\right)$$

$$g[x\backslash\backslash(f,(1,1,1),1)] > 0 \Leftrightarrow \left(a < 0 \wedge \left(p < 3b \vee 3a^2p + (p-3b)^2 < 0\right)\right) \vee$$
$$\left(3a^2p + (p-3b)^2 > 0 \wedge p < 3b \wedge \left(a^2 + p < b \vee b\left(a^2p + (b-p)^2\right) + q^2 > aq\left(a^2 - 3b + p\right)\right)\right) \vee$$
$$\left(a^2 + p < b \wedge \left(b\left(a^2p + (b-p)^2\right) + q^2 < aq\left(a^2 - 3b + p\right) \vee 2q > a\left(a^2 - 3b + p\right)\right)\right) \vee$$
$$\left(2q > a\left(a^2 - 3b + p\right) \wedge b\left(a^2p + (b-p)^2\right) + q^2 > aq\left(a^2 - 3b + p\right)\right)$$

$$g[x\backslash\backslash(f,(1,1,1),1)] = 0 \Leftrightarrow 4b \leq a^2 \wedge b\left(a^2p + (b-p)^2\right) + q^2 = aq\left(a^2 - 3b + p\right) \wedge$$
$$(\left(3a^2 + 2p > 6b \wedge \left(a^2 = 4b \vee \left(a > 0 \wedge a^2 + p \leq b \wedge 2q \leq a\left(a^2 - 3b + p\right)\right)\right)\right) \vee \left(a > 0 \wedge a^2 + p > b \wedge 2q > a\left(a^2 - 3b + p\right)\right) \vee$$
$$\left(3a^2p + (p-3b)^2 > 0 \wedge \left(\left(a^2 + p > b \wedge b < 0\right) \vee \left(4b < a^2 \wedge 2q \leq a\left(a^2 - 3b + p\right)\right)\right)\right))$$

$$g[x\backslash\backslash(f,(1,1,1),1)] < 0 \Leftrightarrow 4b < a^2 \wedge ((b < 0 \wedge (\left(3a^2p + (p-3b)^2 > 0 \wedge b\left(a^2p + (b-p)^2\right) + q^2 < aq\left(a^2 - 3b + p\right)\right) \vee$$
$$\left(2q < a\left(a^2 - 3b + p\right) \wedge \left(\left(a > 0 \wedge a^2 + p > b\right) \vee \left(p > 3b \wedge 3a^2p + (p-3b)^2 > 0\right)\right)\right)))\vee$$
$$\left(a > 0 \wedge \left(\left(2q < a\left(a^2 - 3b + p\right) \wedge 3a^2p + (p-3b)^2 < 0\right) \vee b\left(a^2p + (b-p)^2\right) + q^2 < aq\left(a^2 - 3b + p\right)\right)\right))$$

$$g[x\backslash\backslash(f,(1,1,1),2)]>0 \Leftrightarrow 4b>a^2 \vee (2q<a(a^2-3b+p) \wedge ((a<0 \wedge (3a^2+2p<6b \vee 3a^2p+(p-3b)^2<0 \vee b>0)) \vee$$
$$(3a^2p+(p-3b)^2>0 \wedge 3a^2+2p<6b))) \vee (2q>a(a^2-3b+p) \wedge$$
$$((3a^2+2p<6b \wedge (a>0 \vee 3a^2p+(p-3b)^2>0)) \vee (a>0 \wedge (3a^2p+(p-3b)^2<0 \vee b>0)))) \vee$$
$$(a^2+p>b \wedge ((b(a^2p+(b-p)^2)+q^2<aq(a^2-3b+p) \wedge (3a^2p+(p-3b)^2<0 \vee b>0)) \vee (b>0 \wedge 3a^2p+(p-3b)^2>0)))$$
$$g[x\backslash\backslash(f,(1,1,1),2)]=0 \Leftrightarrow b(a^2p+(b-p)^2)+q^2=aq(a^2-3b+p) \wedge$$
$$((a<0 \wedge ((2q<a(a^2-3b+p) \wedge ((p<3b \wedge a^2+p<b) \vee (3a^2p+(p-3b)^2<0 \wedge a^2+p \leq b))) \vee$$
$$(a^2+p \geq b \wedge 3a^2p+(p-3b)^2<0 \wedge 2q \geq a(a^2-3b+p)))) \vee$$
$$(p<3b \wedge ((a>0 \wedge 2q \geq a(a^2-3b+p) \wedge a^2+p<b) \vee (3a^2p+(p-3b)^2>0 \wedge a^2+p \leq b))) \vee$$
$$((a^2+p \geq b \vee 2q \geq a(a^2-3b+p)) \wedge 3a^2p+(p-3b)^2<0 \wedge (2q<a(a^2-3b+p) \vee a^2+p \leq b) \wedge a>0))$$
$$g[x\backslash\backslash(f,(1,1,1),2)]<0 \Leftrightarrow (p>3b \wedge 3a^2p+(p-3b)^2>0) \vee$$
$$((a<0 \vee 2q<a(a^2-3b+p)) \wedge a \neq 0 \wedge (a^2+p<b \vee b(a^2p+(b-p)^2)+q^2>aq(a^2-3b+p)) \wedge$$
$$(3a^2p+(p-3b)^2<0 \vee p>3b) \wedge a(a^2-3b+p) \neq 2q \wedge (a>0 \vee 2q>a(a^2-3b+p)))$$
$$g[x\backslash\backslash(f,(1,1,1),3)]>0 \Leftrightarrow (b(a^2p+(b-p)^2)+q^2>aq(a^2-3b+p) \wedge (a<0 \vee p<3b \vee 2q<a(a^2-3b+p))) \vee$$
$$(p<3b \wedge ((3a^2p+(p-3b)^2>0 \wedge a^2+p<b) \vee a>0)) \vee (a^2+p<b \wedge 2q<a(a^2-3b+p)) \vee (3a^2p+(p-3b)^2<0 \wedge a>0)$$
$$g[x\backslash\backslash(f,(1,1,1),3)]=0 \Leftrightarrow 4b \leq a^2 \wedge 3a^2+2p \neq 6b \wedge b(a^2p+(b-p)^2)+q^2=aq(a^2-3b+p) \wedge$$
$$((3a^2+2p>6b \wedge (a^2=4b \vee (2q>a(a^2-3b+p) \wedge a<0 \wedge a^2+p<b))) \vee$$
$$(a^2+p \geq b \wedge ((a<0 \wedge 2q \leq a(a^2-3b+p)) \vee (3a^2p+(p-3b)^2>0 \wedge b<0))) \vee$$
$$(4b<a^2 \wedge 3a^2p+(p-3b)^2>0 \wedge 2q>a(a^2-3b+p)))$$
$$g[x\backslash\backslash(f,(1,1,1),3)]<0 \Leftrightarrow (a<0 \wedge ((2q>a(a^2-3b+p) \wedge ((a^2+p>b \wedge b<0) \vee 3a^2p+(p-3b)^2<0)) \vee$$
$$b(a^2p+(b-p)^2)+q^2<aq(a^2-3b+p))) \vee$$
$$(3a^2p+(p-3b)^2>0 \wedge ((b<0 \wedge b(a^2p+(b-p)^2)+q^2<aq(a^2-3b+p)) \vee (p>3b \wedge 2q>a(a^2-3b+p))))$$

$$g[x\backslash\backslash(f,(1),1)]>0 \Leftrightarrow 4b-a^2>0 \vee a^3(-q)+a^2bp+3abq-apq+b^3-2b^2p+bp^2+q^2>0$$
$$g[x\backslash\backslash(f,(1),1)]=0 \Leftrightarrow 4b-a^2 \leq 0 \wedge a^3(-q)+a^2bp+3abq-apq+b^3-2b^2p+bp^2+q^2=0$$
$$g[x\backslash\backslash(f,(1),1)]<0 \Leftrightarrow b(a^2p+(b-p)^2)+q^2<aq(a^2-3b+p) \wedge (b<0 \vee (a<0 \wedge q<0) \vee (a>0 \wedge q>0))$$

$f=x^4+px^2+qx+r$ and $g=x+a$

$$g[x\backslash\backslash(f,(4),1)]>0 \Leftrightarrow a>0 \quad g[x\backslash\backslash(f,(4),1)]=0 \Leftrightarrow a=0 \quad g[x\backslash\backslash(f,(4),1)]<0 \Leftrightarrow a<0$$

$$g[x\backslash\backslash(f,(3,1),1)]>0 \Leftrightarrow 6a^2+p>0 \wedge a>0 \quad g[x\backslash\backslash(f,(3,1),2)]>0 \Leftrightarrow a>0 \vee a^4+a^2p+r<aq$$
$$g[x\backslash\backslash(f,(3,1),1)]=0 \Leftrightarrow 6a^2+p=0 \wedge a>0 \quad g[x\backslash\backslash(f,(3,1),2)]=0 \Leftrightarrow q>4a^3+2ap \wedge a^4+a^2p+r=aq$$
$$g[x\backslash\backslash(f,(3,1),1)]<0 \Leftrightarrow 6a^2+p<0 \vee a<0 \quad g[x\backslash\backslash(f,(3,1),2)]<0 \Leftrightarrow a<0 \wedge q>4a^3+2ap \wedge a^4+a^2p+r>aq$$

$$g[x\backslash\backslash(f,(1,3),1)]>0 \Leftrightarrow a>0 \wedge a^4+a^2p+r>aq \qquad g[x\backslash\backslash(f,(1,3),2)]>0 \Leftrightarrow 6a^2+p<0 \vee a>0$$
$$g[x\backslash\backslash(f,(1,3),1)]=0 \Leftrightarrow a>0 \wedge 6a^2+p>0 \wedge a^4+a^2p+r=aq \qquad g[x\backslash\backslash(f,(1,3),2)]=0 \Leftrightarrow 6a^2+p=0 \wedge a<0$$
$$g[x\backslash\backslash(f,(1,3),1)]<0 \Leftrightarrow q>4a^3+2ap \vee a^4+a^2p+r<aq \qquad g[x\backslash\backslash(f,(1,3),2)]<0 \Leftrightarrow 6a^2+p>0 \wedge a<0$$

$$g[x\backslash\backslash(f,(2,2),1)]>0 \Leftrightarrow 6a^2+p<0 \vee a>0 \qquad g[x\backslash\backslash(f,(2,2),2)]>0 \Leftrightarrow a>0 \vee 2a^2+p<0$$
$$g[x\backslash\backslash(f,(2,2),1)]=0 \Leftrightarrow 6a^2+p=0 \wedge a<0 \qquad g[x\backslash\backslash(f,(2,2),2)]=0 \Leftrightarrow a<0 \wedge 2a^2+p=0$$
$$g[x\backslash\backslash(f,(2,2),1)]<0 \Leftrightarrow a<0 \vee 2a^2+p<0 \qquad g[x\backslash\backslash(f,(2,2),2)]<0 \Leftrightarrow a<0 \wedge 2a^2+p>0$$

$$g[x\backslash\backslash(f,(2,1,1),1)]>0 \Leftrightarrow a>0 \wedge 6a^2+p>0 \wedge q<4a^3+2ap$$
$$g[x\backslash\backslash(f,(2,1,1),1)]=0 \Leftrightarrow a>0 \wedge 6a^2+p>0 \wedge 4a^3+2ap=q$$
$$g[x\backslash\backslash(f,(2,1,1),1)]<0 \Leftrightarrow 6a^2+p<0 \vee a<0 \vee q>4a^3+2ap$$
$$g[x\backslash\backslash(f,(2,1,1),2)]>0 \Leftrightarrow (a^4+a^2p+r>aq \wedge (q<4a^3+2ap \vee a>0)) \vee (a>0 \wedge 6a^2+p>0)$$
$$g[x\backslash\backslash(f,(2,1,1),2)]=0 \Leftrightarrow q<4a^3+2ap \wedge a^4+a^2p+r=aq$$
$$g[x\backslash\backslash(f,(2,1,1),2)]<0 \Leftrightarrow a^4+a^2p+r<aq \vee (a<0 \wedge q>4a^3+2ap)$$
$$g[x\backslash\backslash(f,(2,1,1),3)]>0 \Leftrightarrow q<4a^3+2ap \vee a>0 \vee a^4+a^2p+r<aq$$
$$g[x\backslash\backslash(f,(2,1,1),3)]=0 \Leftrightarrow q>4a^3+2ap \wedge a^4+a^2p+r=aq$$
$$g[x\backslash\backslash(f,(2,1,1),3)]<0 \Leftrightarrow a<0 \wedge q>4a^3+2ap \wedge a^4+a^2p+r>aq$$

$$
\begin{aligned}
&g[x\backslash\backslash(f,(1,2,1),1)]>0\Leftrightarrow a>0\wedge a^4+a^2p+r>aq\\
&g[x\backslash\backslash(f,(1,2,1),1)]=0\Leftrightarrow a>0\wedge 6a^2+p>0\wedge a^4+a^2p+r=aq\\
&g[x\backslash\backslash(f,(1,2,1),1)]<0\Leftrightarrow q>4a^3+2ap\vee 6a^2+p<0\vee a^4+a^2p+r<aq\\
&g[x\backslash\backslash(f,(1,2,1),2)]>0\Leftrightarrow \big(q>4a^3+2ap\wedge\big(6a^2+p<0\vee a>0\big)\big)\vee\big(a>0\wedge 6a^2+p>0\big)\\
&g[x\backslash\backslash(f,(1,2,1),2)]=0\Leftrightarrow 6a^2+p<0\wedge 4a^3+2ap=q\\
&g[x\backslash\backslash(f,(1,2,1),2)]<0\Leftrightarrow \big(a<0\wedge\big(q<4a^3+2ap\vee 6a^2+p>0\big)\big)\vee\big(6a^2+p<0\wedge q<4a^3+2ap\big)\\
&g[x\backslash\backslash(f,(1,2,1),3)]>0\Leftrightarrow 6a^2+p<0\vee a>0\vee a^4+a^2p+r<aq\\
&g[x\backslash\backslash(f,(1,2,1),3)]=0\Leftrightarrow q>4a^3+2ap\wedge a^4+a^2p+r=aq\\
&g[x\backslash\backslash(f,(1,2,1),3)]<0\Leftrightarrow a<0\wedge q>4a^3+2ap\wedge a^4+a^2p+r>aq
\end{aligned}
$$

$$
\begin{aligned}
&g[x\backslash\backslash(f,(1,1,2),1)]>0\Leftrightarrow a>0\wedge a^4+a^2p+r>aq\\
&g[x\backslash\backslash(f,(1,1,2),1)]=0\Leftrightarrow a>0\wedge 6a^2+p>0\wedge a^4+a^2p+r=aq\\
&g[x\backslash\backslash(f,(1,1,2),1)]<0\Leftrightarrow q>4a^3+2ap\vee a<0\vee a^4+a^2p+r<aq\\
&g[x\backslash\backslash(f,(1,1,2),2)]>0\Leftrightarrow a^4+a^2p+r<aq\vee\big(a>0\wedge q<4a^3+2ap\big)\\
&g[x\backslash\backslash(f,(1,1,2),2)]=0\Leftrightarrow q>4a^3+2ap\wedge a^4+a^2p+r=aq\\
&g[x\backslash\backslash(f,(1,1,2),2)]<0\Leftrightarrow \big(a<0\wedge\big(6a^2+p>0\vee a^4+a^2p+r>aq\big)\big)\vee\big(q>4a^3+2ap\wedge a^4+a^2p+r>aq\big)\\
&g[x\backslash\backslash(f,(1,1,2),3)]>0\Leftrightarrow 6a^2+p<0\vee a>0\vee q<4a^3+2ap\\
&g[x\backslash\backslash(f,(1,1,2),3)]=0\Leftrightarrow a<0\wedge 6a^2+p>0\wedge 4a^3+2ap=q\\
&g[x\backslash\backslash(f,(1,1,2),3)]<0\Leftrightarrow<0\&\&q>4a^3+2ap
\end{aligned}
$$

$$
\begin{aligned}
&g[x\backslash\backslash(f,(1,1,1,1),1)]>0\Leftrightarrow a>0\wedge a^4+a^2p+r>aq\\
&g[x\backslash\backslash(f,(1,1,1,1),1)]=0\Leftrightarrow a>0\wedge 6a^2+p>0\wedge a^4+a^2p+r=aq\\
&g[x\backslash\backslash(f,(1,1,1,1),1)]<0\Leftrightarrow 6a^2+p<0\vee q>4a^3+2ap\vee a<0\vee a^4+a^2p+r<aq\\
&g[x\backslash\backslash(f,(1,1,1,1),2)]>0\Leftrightarrow a^4+a^2p+r<aq\vee\big(a>0\wedge q<4a^3+2ap\big)\\
&g[x\backslash\backslash(f,(1,1,1,1),2)]=0\Leftrightarrow q>4a^3+2ap\wedge a^4+a^2p+r=aq\\
&g[x\backslash\backslash(f,(1,1,1,1),2)]<0\Leftrightarrow \big(a<0\wedge\big(q<4a^3+2ap\vee 6a^2+p>0\big)\big)\vee\\
&\quad\big(6a^2+p<0\wedge\big(q<4a^3+2ap\vee a^4+a^2p+r>aq\big)\big)\vee\big(q>4a^3+2ap\wedge a^4+a^2p+r>aq\big)\\
&g[x\backslash\backslash(f,(1,1,1,1),3)]>0\Leftrightarrow \big(6a^2+p<0\wedge\big(q>4a^3+2ap\vee a^4+a^2p+r>aq\big)\big)\vee\\
&\quad\big(q<4a^3+2ap\wedge a^4+a^2p+r>aq\big)\vee\big(a>0\wedge\big(6a^2+p>0\vee q>4a^3+2ap\big)\big)\\
&g[x\backslash\backslash(f,(1,1,1,1),3)]=0\Leftrightarrow q<4a^3+2ap\wedge a^4+a^2p+r=aq\\
&g[x\backslash\backslash(f,(1,1,1,1),3)]<0\Leftrightarrow \big(6a^2+p<0\wedge a^4+a^2p+r<aq\big)\vee\big(a<0\wedge\big(a^4+a^2p+r<aq\vee\big(6a^2+p>0\wedge q>4a^3+2ap\big)\big)\big)\\
&g[x\backslash\backslash(f,(1,1,1,1),4)]>0\Leftrightarrow 6a^2+p<0\vee q<4a^3+2ap\vee a>0\vee a^4+a^2p+r<aq\\
&g[x\backslash\backslash(f,(1,1,1,1),4)]=0\Leftrightarrow q>4a^3+2ap\wedge a^4+a^2p+r=aq\\
&g[x\backslash\backslash(f,(1,1,1,1),4)]<0\Leftrightarrow a<0\wedge q>4a^3+2ap\wedge a^4+a^2p+r>aq
\end{aligned}
$$

The formulas for the root types (2) and $(1,1)$ could not be simplified sufficiently to sensibly be printed here.

Selbständigkeitserklärung

Ich erkläre ausdrücklich, dass es sich bei der von mir eingereichten schriftlichen Arbeit mit dem Titel

Some Aspects of Real Quantifier Elimination

um eine von mir selbst und ohne unerlaubte Beihilfe verfasste Originalarbeit handelt.
Ich bestätige überdies, dass die Arbeit als Ganze oder in Teilen nicht zur Abgeltung anderer Studienleistungen eingereicht worden ist.
Ich erkläre ausdrücklich, dass ich sämtliche in der oben genannten Arbeit enthaltenen Bezüge auf fremde Quellen (einschließlich Tabellen, Grafiken u.Ä.) als solche kenntlich gemacht habe.
Insbesondere bestätige ich, dass ich nach bestem Wissen sowohl bei wörtlich übernommenen Aussagen (Zitaten) als auch bei in eigenen Worten wiedergegebenen Aussagen anderer Autorinnen oder Autoren (Paraphrasen) die Urheberschaft angegeben habe.
Ich nehme zur Kenntnis, dass Arbeiten, welche die Grundsätze der Selbständigkeitserklärung verletzen – insbesondere solche, die Zitate oder Paraphrasen ohne Herkunftsangaben enthalten – , als Plagiat betrachtet werden können.
Ich bestätige mit meiner Unterschrift die Richtigkeit dieser Angaben.[2]

Berlin, den 21.11.2019 . . . .

Tobias Stamm

[2] Text übernommen von https://www.fu-berlin.de/sites/dse/studium/abschlussarbeiten/masterarbeit/selbsterklaerung.pdf