Real Quantifier Elimination in Geometry[1]

Thomas Sturm[2]
FMI, Universität Passau, D-94030 Passau, Germany

December 3, 1999

[1]Im Dezember 1999 an der Fakultät für Mathematik und Informatik der Universität Passau eingereichte Dissertation

[2]`http://www.fmi.uni-passau.de/~sturm/`, `sturm@uni-passau.de`

Contents

List of Figures

Chapter 1

Introduction

In 1948, the RAND Corporation published a technical report titled *A Decision Method for Elementary Algebra and Geometry* [Tar48]. The author of this was the famous polish-born mathematician Alfred Tarski, and it had been "prepared for publication" by J.C.C. McKinsey, who was then a professor at Stanford University cooperating with the RAND. There are a lot of fascinating stories to tell around this extremely important paper, the knowledge of which will help the reader to understand the position of this maybe a little less important thesis in its scientific area.

First of all, the results published were, as stated by Tarski himself in the preface of a second edition [Tar51, Tar98], nearly 20 years old:

> "The main results were found in 1930 and first mentioned in a print one year later. It took nine years, however, before the material in its full development was prepared for publication."

This scientific accurateness cost Tarski another 10 years, because, due to the World War II, the monograph *Actualités scientifiques et industrielles* he had submitted it to never appeared.

In the title, "elementary algebra" refers to the theory of real numbers with ordering, and the term "elementary geometry" reflects Tarski's insight that there is a certain correspondence between polynomial constraint systems and geometric configurations. This thesis focuses on this correspondence. Tarski's paper actually contains much more than a decision procedure, it in fact describes a quantifier elimination procedure: One can not only decide first-order sentences, but in a certain sense parametrically decide formulas containing also free variables. As demonstrated throughout this thesis, proper quantifier elimination problems play a much more prominent role for geometry than pure decision problems. Amazingly, quantifier elimination seemed to be of minor importance for Tarski, and even more amazingly, also McKinsey did not adapt the title.

McKinsey is the link between the pure mathematician Tarski and the application oriented RAND corporation. Apart from the fact that the RAND provided with their technical report serious a generous support for the scientific community at these days, it is very likely, and explicitly claimed by Tarski in the preface cited above, that the RAND was really interested in the work:

> "Within a few months the monograph was published. As was to be expected, it reflected the specific interests which the RAND Cor-

> poration found in the results. The decision method for elementary algebra and geometry—which is one of the main results of the work—was presented in a systematic and detailed way, thus bringing to the fore the possibility of constructing an actual decision machine."

Tarski's original manuscript is still available, and comparing it to McKinsey's version, one actually gets the impression that any mathematical elegance has been sacrificed for the purpose of creating implementation instructions. The RAND does not hold any records on an actual implementation attempt[1], and it would obviously have been completely hopeless these days. Anyway, we have learned that two ideas came immediately into existence with the publication of the first real quantifier elimination procedure: "implement it," and "this is good for geometry." Both ideas also form the basis of our thesis.

1.1 Overview

Chapter 2 starts by introducing the notion of real quantifier elimination, and gives a detailed overview of the historical development of real quantifier elimination and related implementations. Various known quantifier elimination procedures are described in order to give an idea of the algorithms behind the computation techniques discussed throughout this thesis. Among the described methods, the *virtual substitution* method is of particular importance for the applications discussed here. This method has been modified by the author to a *generic quantifier elimination* procedure for solving problems of geometric nature. This thesis is going to demonstrate that generic quantifier elimination is an outstanding tool, which solves problems that are not accessible to other elimination methods.

In view of the fact that real quantifier elimination procedures amount to decision methods for variable-free formulas, it is a straightforward idea for Chapter 3 to apply them for automated theorem proving. Theorems of elementary geometry have traditionally been considered an important test case for the scope of methods in automatic theorem proving. In geometric theorem proving, not only first-order sentences have to be decided, but there have, in addition, certain non-degeneracy conditions to be derived. It will turn out that this task is in an excellent manner performed by our newly discovered generic quantifier elimination. Since (generic) quantifier elimination methods are not restricted to closed sentences, it turns out that they can even be used for geometric theorem *finding* instead of simply proving given theorems.

Chapter 4 turns to computational aspects of geometry being concerned with a number of computational problems in real 3-space or the real plane that arise from applications mainly in the area of CAD, computer vision, and motion planning. This includes parallel and central projections, offset computations, Voronoi diagrams, and collision problems. We will again use almost exclusively our *generic* variant of extended quantifier elimination, although the topics treated here are, in contrast to most geometric theorems, not inherently subject to non-degeneracy assumptions. The reason for this is that the generic quantifier elimination turns out much more efficient than the regular one. Many of our examples from computational geometry involve a significant number of

[1] Anthony C. Hearn, private communication

parameters besides the variables to be eliminated. Examples of this kind have been inaccessible to other implemented real elimination methods so far.

In Chapter 4 we have modeled the geometric objects as sets of satisfying points of quantifier-free formulas over the reals. Chapter 5 proposes to use this modeling, with certain restrictions on valid objects, also for solid modeling systems. Recent modeling systems usually operate with a hybrid representation of objects mixing a high-level CSG description on one hand and a low-level B-REP description on the other hand. The former is used for sophisticated operations on the objects, while the latter is mainly used for rendering the objects on the output device. In theory, our formula representation can be viewed to extend CSG. We show that we can perform on our formula representation all the usual operations encountered in solid modeling. In addition, our framework is well-suited for rounding and blending of solids, which has been an important research topic in the solid modeling community during the past years, and has not yet been satisfactorily solved. We also illustrate, how B-REP representations of our solids can be computed from our description by quantifier-free formulas. Finally, we demonstrate how to treat geometric feasibility problems in our framework. The background for this is that the latest generation of CAD systems is being designed to accept as input declarative descriptions of scenes instead of procedural drawing commands.

It is an important feature of our approach to computational geometry and solid modeling that we can solve all problems parametrically. We can, e.g., compute the blend of a solid with respect to a parametric radius r. The resulting description will contain case distinctions that make it correct for any substitution of a positive real number for r. Similarly, the solids themselves can be described parametrically, and the quantifier elimination work can be done once for an entire class of solids.

1.2 Main Results

We introduce the notion of generic quantifier elimination. This variant of real quantifier elimination is, in general, faster and produces smaller results than regular quantifier elimination. This is achieved by avoiding certain case distinctions on the parameter variables contained in the formula φ under consideration. Technically, instead of the case distinctions one generates in addition to a quantifier-free formula φ', a conjunction ϑ' of non-zero assumptions on polynomials in the parameters. This ϑ' is sufficient for φ' to be a correct elimination result. Formally, we have

$$\vartheta' \longrightarrow (\varphi \longleftrightarrow \varphi').$$

Applying our concept of generic quantifier elimination to automated geometric theorem proving, it turns out that we can compete with the well-known established algebraic approaches to this. Moreover, it turns out that the assumptions ϑ' generated by our procedure provide, as a rule, important non-degeneracy conditions on the geometric input configurations. That is, generic quantifier elimination is not only more efficient than the regular one, but even a superior concept for this task. Since generic quantifier elimination is a real, in contrast to complex, method, its application range exceeds that of the traditional algebraic approaches in various ways: it can handle ordering inequalities in both input and output, it can prove theorems the algebraic formulation of which holds over the reals but not over the complex numbers, and it can disprove theorems. Whenever a theorem is not generically true, our elimination result φ' provides necessary and sufficient conditions in the parameters for the theorem to hold. This extends our approach from theorem proving to theorem finding, which cannot be accomplished by any of the traditional algebraic methods.

We next treat traditional problems from computational geometry by our method. This includes parallel and central projections and corresponding reconstruction problems, offset computations, Voronoi diagrams, and collision problems. Here it again turns out that generic quantifier elimination dramatically extends the range of problems that can be handled reasonably. This time, however, the assumptions made do not provide any valuable extra information. On the other hand generic quantifier elimination does not impose any restrictions that cannot be overcome. It turns out to be a crucial feature of our approach that we can handle the considered problems parametrically, i.e., we simultaneously solve problems for an entire class of objects and configurations, which is given by all real interpretations for the parameters.

Inspired by the success of our approach in two different scientific areas with geometry as their common basis, we examine the question how solid modeling systems could benefit from real (generic) quantifier elimination. We suggest a theoretical framework extending the well-known constructive solid geometry. Our solids are certain semialgebraic sets described by quantifier-free formulas. Based on this notion of a solid, we clarify that the usual operations of solid modeling can be performed by means of quantifier elimination techniques. We then focus in particular on blendings from inside and from outside of solids, which has been extensively discussed in the solid modeling community, and which is not yet satisfactorily solved. Our framework avoids some common problems with blendings. Moreover, it can be used for analyzing these problems, and for systemizing the existing approaches. Our solids can be rendered on the usual output devices by turning them into the commonly accepted B-REP representation. We discuss algorithms involving quantifier elimination for this. Finally, we show that (generic) quantifier elimination is straightforwardly applicable to geometric constraint solving, which is of increasing importance in the scientific literature on the design of graphics systems. There it is considered to be the basic tool for the next generation of modeling systems.

Our generic quantifier elimination has been implemented within the computer logic system REDLOG by the author et al. that, as a part of the computer algebra system REDUCE, is widely accepted and used inside the scientific community. In fact, REDUCE provides with REDLOG the currently only professionally maintained and marketed implementation of real quantifier elimination.

Finally, it should be mentioned that our methods are restricted to problems where the quantified variables occur with a small total degree only. For the majority of problems discussed throughout this thesis the restriction was not relevant. There are, however, few examples where our method simply fails. We consider the selection of examples discussed here representative in this concern.

Chapter 2

Real Quantifier Elimination

Starting with some simple considerations about the zeroes of parametric polynomial functions, we naturally introduce in Section 2.1 the necessary framework for real quantifier elimination. Section 2.2 gives a detailed overview of the historical development of real quantifier elimination and related implementations. In Section 2.3, we first describe Tarski's quantifier elimination method, which—as the first such method—contains many ideas of general interest for the topic. Then we describe three further methods, which are exactly the implemented ones. Among these methods, the *virtual substitution* method is of particular importance for this thesis. It has been implemented by the author et al., and modified by the author to a *generic quantifier elimination* procedure for solving the types of geometric problems discussed throughout this thesis. This generic quantifier elimination is, besides another quantifier elimination variant, described in Section 2.4. Section 2.5 briefly sketches simplification strategies for formulas to the extent necessary for understanding the solution strategies provided within this thesis. Section 2.6 describes the design of the computer logic system REDLOG, which is a joint work by the author together with A. Dolzmann [DS97b]. For illustrating that the application area for real quantifier elimination is by no means restricted to the geometric types of problems discussed here, Section 2.7 provides a summary of other areas, where real quantifier elimination has been successfully applied, both by the author and others. Section 2.8 finally discusses the position of real quantifier elimination methods in relation to general theorem proving approaches and to logic programming.

2.1 The Notion of Quantifier Elimination

Consider a parametric affine linear function $f_{a,b}(x) = ax + b$ over the reals. For $a \neq 0$ it will have a zero $f_{a,b}(-b/a) = 0$. For $a = 0$ we find that $f_{0,b}(x) = b$ has a zero if and only if $b = 0$. Formally, we have found the equivalence

$$\exists x(ax + b = 0) \longleftrightarrow a \neq 0 \vee (a = 0 \wedge b = 0)$$

by *eliminating* both the quantifier "$\exists x$" and the corresponding bound variable x. For given parameters $a, b \in \mathbb{R}$ the question if f has a zero can now be efficiently decided by evaluating the elimination result $a \neq 0 \vee (a = 0 \wedge b = 0)$ to either "true" or "false."

Let us turn to another parametric real function $g_c(x) = x^2 + c$. This function g_c has a real zero if and only if $c \leqslant 0$:

$$\exists x(x^2 + c = 0) \longleftrightarrow c \leqslant 0.$$

For given $c \in \mathbb{R}$, the condition $c \leqslant 0$ again provides an efficient test for whether g_c has a real zero. Note that the elimination result involves not equations and negated equations but an ordering constraint, which we refer to as an *inequality*. It is not hard to see that the resulting inequality $c \leqslant 0$ cannot be restated by means of combining finitely many possibly negated equations.

Multiple quantifiers can be successively eliminated starting with the innermost one: Consider, e.g., an elimination problem of the following form:

$$\varphi(u_1, \dots, u_m) \equiv \exists x \exists y \psi(u_1, \dots, u_m, x, y).$$

Elimination of "$\exists y$" yields $\exists y \varphi(u_1, \dots, u_m, x, y) \longleftrightarrow \iota(u_1, \dots, u_m)$. It follows that $\varphi \longleftrightarrow \exists x \iota$. Thus eliminating "$\exists x$" from $\exists x \iota$ yields a quantifier-free elimination result for φ. Our example function g_c has shown that ι will in general contain ordering inequalities. Our example function $f_{a,b}$ has shown that ι can moreover contain boolean combinations of constraints. A quantifier elimination procedure thus has to permit as input arbitrary boolean combinations of constraints. The elimination of a universal quantifier "$\forall x$" can be reduced to that of an existential quantifier by means of the equivalence $\forall x \psi \longleftrightarrow \neg \exists x \neg \psi$.

2.2 The History of Real Quantifier Elimination

The first mathematical results with a certain spirit of real quantifier elimination date back to the 17th century. They are concerned with counting real zeroes of polynomials. In 1637, R. Descartes established his *rule of signs* [Des73]:

Theorem 2.1 (Descartes, 1637) *Let f be a squarefree univariate real polynomial. Denote by s the number of sign changes in the coefficients of f ignoring zeroes, and by n the number of different positive real roots of f. Then $n \leqslant s$, $s - n$ is even, and if all roots of f are real, then $n = s$.*

Note that Descartes' theorem provides rather precise information in certain cases: For odd s, we know that there must be at least one real root, and for $s = 1$ it follows that $n = 1$, i.e. there is exactly one real root. Information on the negative roots can, of course, be obtained by considering $f(-x)$ instead of $f(x)$, and $f(0)$ can be explicitly tested. This theorem unfortunately provides only an upper bound on the real roots. For the purpose of eliminating an existential quantifier one is interested in the fact whether 1 is a lower bound. That is $1 \leqslant n$ in terms of the theorem.

Budan (1807) and Fourier (1831) showed that Descartes' rule of signs is actually a special case of a more general theorem [BdB07, Fou31]:

Theorem 2.2 (Budan, 1807, Fourier, 1831) *Let f be a squarefree univariate real polynomial, and let $a < b$ be real numbers such that $f(a) \neq 0$ and $f(b) \neq 0$. Let D be the finite sequence of higher non-zero derivatives of f. Denote by s the difference between the number of sign changes in the sequence $D(a)$*

and the number of sign changes in the sequence $D(b)$, and let n be the number of different real roots of f on $[a,b]$. Then $n \leqslant s$, $s-n$ is even, and if all roots of f are real, then $n = s$.

Descartes rule of signs is obtained from this by choosing $a = 0$ and b sufficiently large. Concerning quantifier elimination, the Budan–Fourier theorem is not more helpful than Descartes' rule of signs. It was, however, an important step toward the first result on the exact number of real roots of a given polynomial by Sturm.

In 1835, Sturm derived the following procedure from the Budan–Fourier theorem by replacing the sequence D of the higher derivatives by the *Sturm sequence* [Stu35]. This yields the exact number of roots. Note that f need not be squarefree here.

Theorem 2.3 (Sturm, 1835) *Let f be a univariate real polynomial, and let $a < b$ be real numbers such that $f(a) \neq 0$ and $f(b) \neq 0$. Let S be the finite sequence of polynomials obtained from f and its derivative f' by successive division with negative remainder. Denote by s the difference between the number of sign changes in the sequence $S(a)$ and the number of sign changes in the sequence $S(b)$. Then $n = s$.*

So far, the theorems addressed a purely *algebraic*, i.e., equational problem. In 1853, Sylvester observed that Sturm's theorem can easily be extended to cover an additional inequality as a side condition [Syl53]. Problems involving also inequalities are called *semi-algebraic*.

Theorem 2.4 (Sylvester, 1853) *Let f be a squarefree univariate real polynomial, and let $a < b$ be real numbers such that $f(a) \neq 0$ and $f(b) \neq 0$. Let g be another univariate real polynomial that is relatively prime to f. Denote by T the finite sequence of polynomials obtained from f and $f'g$ by successive division with negative remainder. Denote by s the difference of the number of sign changes in the sequence $T(a)$ and the number of sign changes in the sequence $T(b)$. Let n_+ and n_- be the number of different real roots of f in $[a,b]$ at which g is positive or negative, respectively. Then $n_+ - n_- = s$.*

Note that in contrast to Sturm's original theorem, f must be squarefree here. This is no problem, since one can simply replace an arbitrary polynomial f by its squarefree part $f/\gcd(f,f')$. The sequence T is usually referred to as *Sturm–Sylvester sequence*. From the total number of real roots of f in $[a,b]$ obtained by Sturm's theorem and the difference of numbers obtained by Sturm–Sylvester one can easily obtain the number of real roots of f in the interval $[a,b]$, where g is positive or negative, respectively. By a little combinatorial trick, this procedure can be extended to count the number of real roots of f, where finitely many other real univariate polynomials have fixed signs. See [BKR86] for an efficient variant of this trick.

Such a combinatorial extension of the Sturm–Sylvester theorem is the core of the first real quantifier elimination procedure found by Tarski in the 1930's, which remained unpublished until 1948 [Tar48]. Tarski's procedure is very inefficient, more precisely it is not even elementary recursive. It is described in Section 2.3.1.

In 1975, Collins introduced a new method called *cylindrical algebraic decomposition* (CAD) [Col75], which is worst-case double exponential in the number of variables. This was the first real quantifier elimination procedure ever implemented. A full implementation of CAD within the computer algebra system SAC-2 [Col85] has been finished by Arnon in 1981 [Arn81]. In the past 20 years of research, CAD has gone through numerous improvements [McC88, Hon90b, Hon90a, Hon92], resulting in *partial* CAD [CH91] implemented in Hong's QEPCAD program based on SACLIB [Hon90b, BCE+93], a C version of SAC-2. CAD is discussed in detail in Section 2.3.2.

In 1988, it has been shown independently by Davenport and Heintz [DH88] and Weispfenning [Wei88] that real quantifier elimination is inherently hard for some problem classes. It is double exponential in the number of quantifier blocks and thus in the word length. Thus the attention turned to special procedures for restricted problem classes, where the elimination procedures can be tuned to the structure of the problem. The focus was on considering formulas in which the occurrence of quantified variables is restricted to low degrees [Wei88, LW93, Hon93a, Hon93b, GV93, Wei94, Wei97b]. This was initiated by Weispfenning in 1988 [Wei88]. In his *virtual substitution* method the number of parameters plays a minor role for the complexity. The worst-case complexity of the method is, in contrast to CAD, double exponential only in the number of the *quantifier blocks* of the input formula. This makes the method attractive for problems containing many parameters. It is implemented in the REDUCE package REDLOG [DS97b, DS99] by the author together with A. Dolzmann. The virtual substitution method is discussed in Section 2.3.3. The REDLOG package is described in Section 2.6.

In 1993, Weispfenning introduced another complete elimination procedure [Wei93, Wei98] based on comprehensive Gröbner bases [Wei92] in combination with multivariate real root counting [Wei92, BW94, PRS93]. The focus of the method is on problems containing many equations. This procedure has been implemented in the package QERRC by Dolzmann within the computer algebra system MAS [Dol94]. It is discussed in Section 2.3.4.

In 1992, Renegar published an impressing series of papers on the complexity of real quantifier elimination [Ren92]. He gave explicit asymptotically fast algorithms. In 1994, Basu *et al.* improved Renegar's complexity bounds by giving an even faster algorithm [BPR94]. There is currently no implementation of these asymptotically fast algorithms. The question to what extent these methods are of practical relevance thus cannot be answered yet.

2.3 Real Quantifier Elimination Procedures

Our aim is to find necessary and sufficient conditions in terms of the parameters $u_1, \ldots, u_m$ for the validity of the following prenex quantified first-order formula:

$$\varphi(u_1, \ldots, u_m) \equiv \mathsf{Q}_1 x_1 \ldots \mathsf{Q}_n x_n \psi(u_1, \ldots, u_m, x_1, \ldots, x_n), \quad \mathsf{Q}_i \in \{\exists, \forall\}.$$

2.3.1 The Tarski Procedure

This first known real quantifier elimination procedure has been discovered by Tarski in 1930. Nine years later Tarski submitted an article under the title

The completeness of elementary algebra and geometry to a collection called *Actualités scientifiques et industrielles*. Due to the world war, this collection never appeared. In 1948, the RAND Corporation became interested in Tarski's result and offered publication. For this purpose, the manuscript was overworked by McKinsey, a friend of Tarski cooperating with the RAND. This overworked version [Tar48] was much more algorithmic than the original article. In the introduction of the second edition [Tar51, Tar98] Tarski commented this issue as follows:

> "Within a few months the monograph was published. As was to be expected, it reflected the specific interests which the RAND Corporation found in the results. The decision method for elementary algebra and geometry—which is one of the main results of the work—was presented in a systematic and detailed way, thus bringing to the fore the possibility of constructing an actual decision machine."

So the dream of implementing real quantifier elimination came into existence as soon as people realized the theoretical possibility. Needless to say that there was no chance for a successful implementation at this time. In fact there a no records on such a project at the RAND (Anthony C. Hearn, private communication). The complexity of Tarski's procedure is not even elementary recursive.

We apply suitable CNF/DNF computations, use if necessary the equivalence $\exists x\psi \longleftrightarrow \neg\forall x \neg\psi$, and rewrite all orderings by means of "="and ">" to come to a formula of the following form:

$$\mathsf{Q}_1 x_1 \dots \mathsf{Q}_{k-1} x_{k-1} \exists x_k \dots \exists x_n \Big(\bigwedge_{f \in F} f(\mathbf{u}, \mathbf{x}) = 0 \wedge \bigwedge_{g \in G} g(\mathbf{u}, \mathbf{x}) > 0 \Big).$$

From this, we successively eliminate the quantifiers starting with the innermost existential one. For the purpose of a comprehensible notation for the description of the procedure, we rename x_n to x, and we rename the outer quantified variables $x_1, \dots, x_{n-1}$, which play the role of parameters for the procedure, to $u_{m+1}, \dots u_{m+n-1}$. Proceeding this way, we arrive at a subproblem φ^* of the following form:

$$\varphi(\mathbf{u})^* \equiv \exists x \Big(\bigwedge_{f \in F} f(\mathbf{u}, x) = 0 \wedge \bigwedge_{g \in G} g(\mathbf{u}, x) > 0 \Big).$$

Note that both F and G are possibly empty.

In case that F is empty, we construct redundant equations from the $g > 0$: Fix the parameters $u_1, \dots, u_{m+n-1}$ to real values $a_1, \dots, a_{m+n-1}$, and suppose there is $b \in \mathbb{R}$ such that $g(\mathbf{a}, b) > 0$ for all $g \in G$. Then one of the following cases must hold:

(i) $g(\mathbf{a}, x) > 0$ for all $b \leqslant x \in \mathbb{R}$ and all $g \in G$.

(ii) $g(\mathbf{a}, x) > 0$ for all $b \geqslant x \in \mathbb{R}$ and all $g \in G$.

(iii) There are $x_1, x_2 \in \mathbb{R}$ and $g_1, g_2 \in G$ such that $x_1 < x < x_2$ and $g_1(\mathbf{a}, x_1) = g_2(\mathbf{a}, x_2) = 0$. It follows that

$$\prod_{g \in G} g(\mathbf{a}, x_1) = \prod_{g \in G} g(\mathbf{a}, x_2) = 0,$$

and by Rolle's theorem there is $x_0 \in \,]x_1, x_2[$ such that $\frac{\partial \prod G}{\partial x}(\mathbf{a}, x_0) = 0$.

The first two cases are recognized by inspection of the leading coefficients of the $g \in G$, and can be described by corresponding quantifier-free formulas. In the third case Rolle's theorem provides a redundant equation. After writing the case distinction down as a first-order formula, and then reapplying DNF computation, we may wlog. assume that F is not empty.

Next, it is easy to see that $\bigwedge_{f \in F} f = 0$ is equivalent to $\sum_{f \in F} f^2 = 0$, so we can restrict our attention to a formula of the following form:

$$\varphi(\mathbf{u})^{**} \equiv \exists x \Big(f(\mathbf{u}, x) = 0 \wedge \bigwedge_{g \in G} g(\mathbf{u}, x) > 0 \Big).$$

The case that G is empty can now be handled by a parametric version of Sturm's theorem. For handling the general case by a corresponding parametric extension of Sturm–Sylvester, we have to reduce the conjunction $\bigwedge_{g \in G} g > 0$ of ordering inequalities to one such inequality. For doing so, we adopt the following crucial definition from Tarski.

Definition 2.5 (Precise Quantifier) Let $k \in \mathbb{N}$. Then $\exists_k x \varphi$ means that there are exactly k distinct real values of x that satisfy φ.

Since obviously $\exists x \varphi \longleftrightarrow \neg \exists_0 x \varphi$, we can turn our attention from φ^{**} to

$$\varphi(\mathbf{u})^{***} \equiv \exists_0 x \Big(f(\mathbf{u}, x) = 0 \wedge \bigwedge_{g \in G} g(\mathbf{u}, x) > 0 \Big).$$

Our aim is to reduce this φ^{***} to a form $\exists_k x \big(f(\mathbf{u}, x) = 0 \wedge g(\mathbf{u}, x) > 0 \big)$ for some $k \in \mathbb{N}$. This is achieved by iteratively applying the following observation by Tarski, where two left hand sides g_0 and g_1 are isolated from G:

Theorem 2.6 *The formula $\psi \equiv \exists_k x (\chi \wedge g_0 > 0 \wedge g_1 > 0)$ is equivalent to the following formula containing one less inequality in each of the three subformulas:*

$$\bigvee_{k = \frac{r_1 + r_2 - r_3}{2}} \big(\exists_{r_1} x (\chi \wedge g_0^2 g_1 > 0) \wedge \exists_{r_2} x (\chi \wedge g_0 g_1^2 > 0) \wedge \exists_{r_3} x (\chi \wedge g_0 g_1 > 0) \big).$$

Proof For fixed real parameter values consider the following numbers of satisfying points of ψ and formulas derived from ψ:

$$\begin{aligned}
n_{++} &= |\{x \in \mathbb{R} \mid \chi \wedge g_0 > 0 \wedge g_1 > 0\}| \\
n_{+-} &= |\{x \in \mathbb{R} \mid \chi \wedge g_0 > 0 \wedge g_1 < 0\}| \\
n_{-+} &= |\{x \in \mathbb{R} \mid \chi \wedge g_0 < 0 \wedge g_1 > 0\}| \\
r_1 &= |\{x \in \mathbb{R} \mid \chi \wedge g_0 g_1^2 > 0\}| \\
r_2 &= |\{x \in \mathbb{R} \mid \chi \wedge g_0^2 g_1 > 0\}| \\
r_3 &= |\{x \in \mathbb{R} \mid \chi \wedge g_0 g_1 > 0\}|.
\end{aligned}$$

Then it is easy to see that

$$r_1 = n_{++} + n_{+-}, \quad r_2 = n_{++} + n_{-+}, \quad r_3 = n_{+-} + n_{-+}.$$

Solving for n_{++} we obtain

$$n_{++} = \frac{r_1 + r_2 - r_3}{2},$$

which straightforwardly implies the equivalence stated by the theorem. □

We have finally arrived at a point, where we have to write down a quantifier-free equivalent for a formula with precise quantifier of the following form:

$$\varphi(\mathbf{u})^{****} \equiv \exists_k x\bigl(f(\mathbf{u},x)=0 \land g(\mathbf{u},x)>0\bigr).$$

This can be done by parameterizing the Sturm–Sylvester theorem, where appropriate case distinctions on the signs and on the vanishing of the parametric coefficients have to be introduced. These case distinctions finally yield the quantifier-free result.

2.3.2 Cylindrical Algebraic Decomposition

Cylindrical algebraic decomposition (CAD) [Col75, ACM84] is the oldest and most elaborate implemented real quantifier elimination method. It was developed by Collins and his students starting in February 1973 [Col74]. Among his numerous Ph.D. students particularly Arnon [Arn81], McCallum [McC84], and Hong [Hon90a, Hon90b, Hon92] made very significant contributions. In 1991 Collins and Hong revised the method resulting in *partial* CAD [CH91]. Hong has implemented partial CAD in his program QEPCAD based on the computer algebra C-library SACLIB [Hon90b, BCE$^+$93]. Compared to earlier implementations of CAD, QEPCAD shows a dramatically improved performance. Unfortunately, the program has until now remained unpublished. In this section, we sketch the pure, i.e. pre-partial, CAD.

In CAD, "decomposition" means partitioning real k-space into disjoint cells with certain good properties.

Definition 2.7 (Sign invariant cell) Consider real k-space $\mathbb{R}^k$. A *cell* is a triplet (C, δ_C, t_C) with the following properties:

(i) $C \subseteq \mathbb{R}^k$ is connected.

(ii) $\delta_C(v_1,\dots,v_k)$ is a quantifier-free formula describing C, i.e, $\mathbf{c} \in C$ if and only if $\delta_C(\mathbf{c})$ holds.

(iii) $t_C \in C$ is a test point for C given in such a way that for any polynomial $f \in \mathbb{R}[V_1,\dots,V_k]$ the sign of $f(t_C)$ can be determined.

A cell (C, δ_C, t_C) in $\mathbb{R}^k$ is called *sign invariant* for some finite set of polynomials $F_k \subseteq \mathbb{R}[V_1,\dots,V_k]$, if each polynomial $f \in F_k$ has constant sign for all $\mathbf{c} \in C$.

Of course, not every subset of real k-space can be described by a formula. Due to the existence of the formulas δ_C the cells are called *semi-algebraic.* Note that the computational claim on t_C is by no means trivial.

Definition 2.8 (Sign invariant cell decomposition) A *cell decomposition* of real k-space $\mathbb{R}^k$ is a finite partition $\mathcal{D}_k$ into cells:

$$\mathcal{D}_k = \{(C_i, \delta_{C_i}, t_{C_i}) \mid i \in I_k\}, \quad C_i \cap C_j = \emptyset \quad \text{for} \quad i \neq j, \quad \bigcup_{i\in I_k} C_i = \mathbb{R}^k.$$

The decomposition $\mathcal{D}_k$ is *sign invariant* for some finite set $F_k \subseteq \mathbb{R}[V_1,\dots,V_k]$ of polynomials, if each cell in $\mathcal{D}_k$ is sign invariant for F_k.

Example 2.9 Consider on the real line $\mathbb{R}$ the set $F = \{x^2 - 4\}$. Then

$$\begin{aligned}\mathcal{D} \;=\; &\big\{\big(]-\infty, -2[\,,\, x < -2,\, -3\big),\, \big(\{-2\},\, x = -2,\, -2\big),\\ &\big(]-2, 2[\,,\, -2 < x \wedge x < 2,\, 0\big),\, \big(\{2\},\, x = 2,\, 2\big),\, \big(]2, \infty[\,,\, 2 < x,\, 3\big)\big\}\end{aligned}$$

is a sign invariant cell decomposition for F. □

Recall that we wish to construct necessary and sufficient conditions in $\mathbf{u}$ for the validity of $\varphi(\mathbf{u}) \equiv \mathsf{Q}_1 x_1 \dots \mathsf{Q}_n x_n \psi(\mathbf{u}, \mathbf{x})$. Let $\mathcal{D}_{m+n}$ be a cell decomposition of $\mathbb{R}^{m+n}$ that is sign invariant for the set of all polynomials occurring in ψ. Then ψ and thus φ will have constant truth value over each cell. Suppose that $\mathcal{D}_m$ is a cell decomposition of the parameter space $\mathbb{R}^m$ obtained from $\mathcal{D}_{m+n}$ by successively abstracting from the quantified variables $x_1, \dots, x_n$. Then one can use the knowledge of "true" cells for ψ and of the quantifier structure to derive from the quantifier-free descriptions δ_C of the cells in $\mathcal{D}_m$ a quantifier-free equivalent of φ. The following definition states the connection between our $\mathcal{D}_m$ and $\mathcal{D}_{m+n}$ more precisely.

Definition 2.10 (CAD) Consider the following first-order formula in n variables and m parameters over the reals:

$$\varphi(u_1, \dots, u_m) \equiv \mathsf{Q}_1 x_1 \dots \mathsf{Q}_n x_n \psi(u_1, \dots, u_m, x_1, \dots, x_n), \quad \mathsf{Q}_i \in \{\exists, \forall\}.$$

Denote by F the set of polynomials occurring in φ. A CAD for φ is a sequence $\big(\mathcal{D}_1, \dots, \mathcal{D}_m, (\mathcal{D}_{m+1}, \mathsf{Q}_1), \dots, (\mathcal{D}_{m+n}, \mathsf{Q}_n)\big)$ with the following properties:

(i) $\mathcal{D}_1, \dots, \mathcal{D}_{m+n}$ are cell decompositions of $\mathbb{R}, \dots, \mathbb{R}^{m+n}$, respectively.

(ii) $\mathcal{D}_{m+n}$ is sign invariant for F.

(iii) For $k \in \{1, \dots, m+n-1\}$ the partition $\mathcal{D}_k$ consists exactly of the projections of the cells in the partition $\mathcal{D}_{k+1}$ along the coordinate of the $k+1$-st variable. For each $(C, \delta_C, t_C) \in \mathcal{D}_k$ we can algorithmically determine the preimage $S(C, \delta_C, t_C) \subseteq \mathcal{D}_{k+1}$ under this projection.

The proof of the following theorem shows how to exploit a CAD for the purpose of quantifier elimination. Note that of the given CAD only the trailing part $\mathcal{D}_m$, $(\mathcal{D}_{m+1}, \mathsf{Q}_1)$, $\dots$, $(\mathcal{D}_{m+n}, \mathsf{Q}_n)$ is actually used.

Theorem 2.11 (Quantifier elimination by CAD) *Let φ be a first-order formula over the reals. Given a CAD for φ, we can eliminate the quantifiers from φ. That is, we can compute some quantifier-free formula φ' such that $\varphi' \longleftrightarrow \varphi$.*

Proof Let $\varphi \equiv \mathsf{Q}_1 x_1 \dots \mathsf{Q}_n x_n \psi(u_1, \dots, u_m, x_1, \dots, x_n)$, and let a CAD for φ be given as $\big(\mathcal{D}_1, \dots, \mathcal{D}_m, (\mathcal{D}_{m+1}, \mathsf{Q}_1), \dots, (\mathcal{D}_{m+n}, \mathsf{Q}_n)\big)$.

Then $\mathcal{D}_{m+n}$ is sign invariant for the polynomials in ψ, and thus truth invariant for ψ. Moreover, for each cell $(C, \delta_C, t_C) \in \mathcal{D}_{m+n}$ the truth value of ψ over C can be determined by evaluating the variable-free formula $\psi(t_C)$. A cell $(C, \delta_C, t_C) \in \mathcal{D}_{m+n}$ is called *valid* if $\psi(t_C)$ is "true."

On this basis, we recursively define the validity of cells in $\mathcal{D}_m$. The validity of a cell $(C, \delta_C, t_C) \in \mathcal{D}_k$ is determined by considering its preimage

$S(C, \delta_C, t_C) \subseteq \mathcal{D}_{k+1}$ under the corresponding projection. For $m \leqslant k < m + n$, the decomposition $\mathcal{D}_{k+1}$ occurs in the CAD as $(\mathcal{D}_{k+1}, \mathsf{Q}_{k+1})$ with $\mathsf{Q}_{k+1} \in \{\exists, \forall\}$. If $Q_{k+1} = \exists$, then (C, δ_C, t_C) is valid if at least one cell in $S(C)$ is valid. If, in contrast, $Q_{k+1} = \forall$, then (C, δ_C, t_C) is valid if all cells in $S(C)$ are valid.

It is now easy to see that a quantifier-free equivalent φ' for φ is obtained as the disjunction of all δ_C for which $(C, \delta_C, t_C) \in \mathcal{D}_m$ is valid. □

We finally sketch how the sequence $\mathcal{D}_1, \ldots, \mathcal{D}_{m+n}$ of cell decompositions is obtained from the set F of polynomials occurring in φ. This task splits into two phases:

Projection Phase Starting with $F_{m+n} := F \subseteq \mathbb{R}[u_1, \ldots, u_m, x_1, \ldots, x_n]$ one successively constructs sets of polynomials

$$\begin{aligned} F_{m+n-1} &\subseteq \mathbb{R}[u_1, \ldots, u_m, x_1, \ldots, x_{n-2}, x_{n-1}], \\ F_{m+n-2} &\subseteq \mathbb{R}[u_1, \ldots, u_m, x_1, \ldots, x_{n-2}], \\ &\vdots \\ F_1 &\subseteq \mathbb{R}[u_1]. \end{aligned}$$

The projection phase is uniform for parameters and main variables. For convenience, we denote $(u_1, \ldots, u_m, x_1, \ldots, x_n)$ also by $(V_1, \ldots, V_{m+n})$. Technically, each F_k will contain certain coefficients, discriminants, resultants, and subresultant coefficients obtained from the polynomials in F_{k+1} and their higher derivatives, regarded as univariate polynomials in their last variable V_{k+1}.

The resulting F_k satisfies the following condition: Consider $\mathbf{a}$, $\mathbf{b} \in \mathbb{R}^k$ such that for all $f' \in F_k$ the signs of both $f'(\mathbf{a})$, $f'(\mathbf{b}) \in \mathbb{R}$ are equal. Then for all $f \in F_{k+1}$ the corresponding univariate polynomials $f(\mathbf{a}, V_{k+1})$, $f(\mathbf{b}, V_{k+1}) \in \mathbb{R}[V_{k+1}]$ both have the same number of different real and complex roots.

This has the following important consequence: Let (C, δ_C, t_C) be a sign invariant cell for F_k. For each $f \in F_{k+1}$ consider the functions $\varrho_i : C \to \mathbb{R}$ assigning to $\mathbf{a} \in C$ the i-th real root of $f(\mathbf{a}, V_{k+1}) \in \mathbb{R}[V_{k+1}]$. Then all these ϱ_i are continuous. Moreover, the graphs of the various ϱ_i do not intersect. In other words, the order of the real roots does not change as they continuously change their position on the real line. In the CAD framework they are called *delineated.* The step from F_{k+1} to F_k is called a *projection.*

Construction Phase While the projection phase stepped from $\mathbb{R}^{m+n}$ down to $\mathbb{R}$, the construction phase proceeds back from $\mathbb{R}$ to $\mathbb{R}^{m+n}$. It starts with the construction of a cell decomposition $\mathcal{D}_1$ of the real line $\mathbb{R}$ that is sign invariant for F_1. For $1 \leqslant k < m + n$ the cell decompositions $\mathcal{D}_{k+1}$ of $\mathbb{R}^{k+1}$, which are sign invariant for F_{k+1} are successively constructed from $\mathcal{D}_k$ as follows.

Above each connected cell (C, δ_C, t_C) in $\mathcal{D}_k$ the roots of all polynomials in F_{k+1} regarded as univariate polynomials in their last variable V_{k+1} are delineated. They thus cut the *cylinder* above C into finitely many connected semi-algebraic segments. We take $\mathcal{D}_{k+1}$ to consist of all such segments arising from cylinders above the cells of $\mathcal{D}_k$. The required test points in the new cells are obtained by lifting the test points t_C of the corresponding base cells in $\mathcal{D}_k$ into the various cylinder segments. A quantifier-free description of each cell in

$\mathcal{D}_{k+1}$ is constructed from the corresponding δ_C together with the polynomials in F_k.

Note that this construction allows us to save the information which cells in $\mathcal{D}_{k+1}$ origin from which cell in $\mathcal{D}_k$ thus satisfying the requirement on the computability of the preimages in Definition 2.10 (iii).

2.3.3 Virtual Substitution of Test Points

The virtual substitution method dates back to a theoretical paper by Weispfenning in 1988 [Wei88]. Subsequently a lot of theoretical work has been done to improve on one hand the method itself and on the other hand the simplification strategies for quantifier-free intermediate and final results [LW93, Wei94, DS97c, Wei97b]. After promising experimental implementations by Burhenne in 1990 [Bur90] and by the author in 1992, the method has been much more efficiently reimplemented within the REDUCE package REDLOG by the author together with A. Dolzmann [DS97b, DS99]. This package is part of REDUCE 3.7. An alternative implementation in the C language based on the computer algebra system Risa/Asir [NT92] has been realized by the author in cooperation with the Fujitsu Laboratories Japan [Stu96]. Finally, there are SACLIB-based parallel implementations by the author together with A. Dolzmann and O. Gloor [DGS98].

The applicability of the method in the form described here and used throughout this thesis is restricted to formulas in which the quantified variables occur at most quadratically. Moreover, as the method successively eliminates all quantifiers, the elimination of one quantifier can increase the degree of the other quantified variables. So it is not generally clear by inspection of the quantified formula whether the implementation will succeed. All linear variables can however definitely be eliminated. From a theoretical point of view the method can be extended to arbitrary degrees [Wei94, Wei97b].

There have been various heuristic methods developed for decreasing the degrees of the contained variables during elimination. One obvious example for such methods is polynomial factorization.

For eliminating the quantifiers from an input formula

$$\varphi(u_1, \ldots, u_m) \equiv \mathsf{Q}_1 x_1 \ldots \mathsf{Q}_n x_n \psi(u_1, \ldots, u_m, x_1, \ldots, x_n), \quad \mathsf{Q}_i \in \{\exists, \forall\}$$

the elimination starts with the innermost quantifier regarding the other quantified variables within ψ as extra parameters. Universal quantifiers are handled by means of the equivalence $\forall x \psi \longleftrightarrow \neg \exists x \neg \psi$. We may thus restrict our attention to a formula of the form

$$\varphi^*(u_1, \ldots, u_k) \equiv \exists x \psi^*(u_1, \ldots, u_k, x),$$

where the $u_{m+1}, \ldots, u_k$ are actually x_i quantified from further outside.

We fix real values $a_1, \ldots, a_k$ for the parameters $u_1, \ldots, u_k$. Then all polynomials occurring in ψ^* become linear or quadratic univariate polynomials in x with real coefficients. So the set

$$M = \{\, b \in \mathbb{R} \mid \psi^*(a_1, \ldots, a_k, b) \,\}$$

of all real values b of x satisfying ψ^* is a finite union of closed, open, and half-open intervals on the real line. The endpoints of these intervals are among $\pm\infty$

together with the real zeroes of the linear and quadratic polynomials occurring in ψ^*. Candidate terms $\alpha_1, \dots, \alpha_r$ for the zeroes can be computed uniformly in $u_1, \dots, u_k$ by the solution formulas for linear and quadratic equations.

If all inequalities in ψ^* are weak, then all the intervals constituting M will, into each direction, be either unbounded or closed. In the latter case, such an interval will contain its real endpoint. Thus M is non-empty if and only if the substitution of $\pm\infty$ or of one of the candidate solutions α_j for x satisfies ψ^*. The substitution of $\pm\infty$ into a polynomial equation or inequality is evaluated in the obvious sense. The substitution of expressions in $u_1, \dots, u_k$ of the form $(a + b\sqrt{c})/d$ among the α_j can be rewritten in such a way that all denominators involving the u_i and all square-root expressions are removed from the result [Wei97b]. By disjunctively substituting all candidates into ψ^* we obtain a quantifier-free formula equivalent to $\exists x \psi^*$.

If ψ^* contains also strict inequalities, we need to add to our candidates for points in M expressions of the form $\alpha \pm \varepsilon$, where α is candidate solution for some left-hand side polynomial occurring in a strict inequality. The symbol ε stands for a positive infinitesimal number. Again, the substitution of these expressions into a polynomial equation or inequality can be rewritten in such a form that there occur neither denominators involving any of the u_i, nor any square root expressions, nor the symbol ε in the result [Wei97b]. Again, this yields a quantifier-free formula equivalent to $\exists x \psi^*$. For practical applications this method, of course, has to be refined by a careful selection of a smaller number of candidate solutions and by a combination with powerful simplification techniques for quantifier-free formulas, see [DS97c] for details. The set of all points that finally have to be substituted is called an *elimination set.*

Recall that the well-known solution formula for quadratic equations

$$ax^2 + bx + c = 0$$

requires $a \neq 0$. In our situation a is a term in $u_1, \dots, u_k$, so $a \neq 0$ can in general not be decided uniformly but depends on the interpretation of the u_i. Thus a quadratic polynomial $ax^2 + bx + c$ does not only deliver two square-root expressions α_1 and α_2 as candidate solutions but also $\alpha_3 = -c/b$, which in turn requires $b \neq 0$. Let t_1, t_2, and t_3 be the candidate points for M obtained from α_1, α_2, and α_3, respectively, by possibly adding or subtracting ε. With the substitution of all the t_i into ψ^*, it is necessary to add the conditions on the non-vanishing of a and b. Formally, we obtain

$$\bigl(a \neq 0 \land \Delta \geqslant 0 \land (\psi^*[x/t_1] \lor \psi^*[x/t_2])\bigr) \lor \bigl(a = 0 \land b \neq 0 \land \psi^*[x/t_3]\bigr), \quad (2.1)$$

where Δ denotes the discriminant of the equation $ax^2 + bx + c = 0$. If, however, a is a rational constant, then the case distinction is superfluous. In particular, if a is non-zero, the second case can be dropped.

Suppose we have eliminated an existential quantifier. Then we have in general obtained a disjunction $\psi_1' \lor \dots \lor \psi_r'$. If the next quantifier to be eliminated is also an existential one, then we make use of the equivalence

$$\exists x_{n-1}(\psi_1' \lor \dots \lor \psi_r') \longleftrightarrow \exists x_{n-1}(\psi_1') \lor \dots \lor \exists x_{n-1}(\psi_r')$$

eliminating all $\exists x_{n-1}(\psi_j')$ independently. As a consequence, no candidate solutions obtained from, say, ψ_1' are substituted into the other ψ_j. This decreases the complexity class of our procedure for single quantifier blocks from double exponential to single exponential in the number of quantifiers [Wei88].

Dramatic improvements of the general procedure sketched up to now can be obtained by reducing the number of test candidates for M depending on the structure of the formula ψ^* [LW93, Wei97b]. One simple instance for such an improvement is the following natural extension of *Gauss elimination*: Suppose ψ^* is of the form

$$bx + c = 0 \wedge \psi^{**},$$

where at least one of the coefficient terms b, c is a rational non-zero constant. Then we know that under any interpretation of the u_i the equation is *non-trivial*, i.e. different from $0 = 0$. Hence the only test candidate required is $-c/b$ substituted, of course, with the condition $b \neq 0$. No additional test candidates arising from equations or inequalities in the remainder ψ^{**} of ψ^* need be considered. This idea can easily be extended to a quadratic equation instead of a linear one, taking into account again the discriminant. Ideas very similar to our extended Gauss elimination have independently been considered by Hong within the CAD framework [Hon93b].

The notion of *virtual substitution* refers to both adding conditions and resolving non-standard subterms such as surds or infinitesimals. The complexity of this method depends on the number of quantified variables and, even more, on the number of quantifier changes. In theory, parameters play a minor role for the complexity. They turn in fact out to be very cheap in practice, too.

2.3.4 Parametric Real Root Counting

The method of parametric real root counting has been described by Weispfenning in 1993 [Wei93, Wei98]. Its basis is a theorem on root counting very much in the spirit of those discussed in Section 2.2 above. This beautiful and far-reaching extension of a univariate theorem by Hermite (1853) [Her05] was found independently by Becker and Wörmann [Bec91, BW94] on one hand and Pedersen, Roy, and Szpirglas [Ped91, PRS93] on the other hand.

Theorem 2.12 (Becker and Wörmann 1991, Pedersen et al. 1991) *Let $I \subseteq \mathbb{R}[x_1, \dots, x_n]$ be a zero-dimensional ideal. For $g \in \mathbb{R}[x_1, \dots, x_n]$ consider the symmetric quadratic form $Q_g = \left(\mathrm{trace}(m_{gb_ib_j})\right)_{1 \leqslant i,j \leqslant d}$ on the linear $\mathbb{R}$-space $S = \mathbb{R}[x_1, \dots, x_n]/I$, where $\{b_1, \dots, b_d\} \subseteq S$ is a basis, and the $m_h : S \to S$ are linear maps defined by $m_h(f + I) = (hf) + I$. Let s be the signature of Q_g, and denote by n_+ and n_- the number of real roots of I at which g is positive or negative, respectively. Then $n_+ - n_- = s$.*

The use of a Gröbner basis of the ideal I allows to obtain a basis of S, and to perform arithmetic there [Buc65], thus obtaining the matrix Q_g.

Similar to the Sturm–Sylvester theorem, this approach can be extended to obtain the exact number of roots under a side condition, and can be moreover extended to several side conditions:

Theorem 2.13 *Let F, $\{g_1, \dots, g_k\} \subseteq \mathbb{R}[x_1, \dots, x_n]$ be finite, and assume that $I = \mathrm{Id}(F)$ is zero-dimensional. Denote by N the number of real roots $a \in \mathbb{R}^n$ of F for which $g_i > 0$ for $1 \leqslant i \leqslant k$. Define*

$$\Gamma(\{g_1, \dots, g_k\}) = \{\, g_1^{e_1} \cdots g_k^{e_k} \mid (e_1, \dots, e_k) \in \{1, 2\}^k \,\}.$$

Then defining Q_g *as above we have* $2^k N = \sum_{\gamma \in \Gamma(\{g_1,\dots,g_k\})} \operatorname{sig}(Q_\gamma)$.

For real quantifier elimination, this root counting has to be further extended to multivariate polynomials with parametric coefficients in such a way that it will remain correct for *every* real specialization of the parameters including specializations to zero. This task has been carried out by Weispfenning using comprehensive Gröbner bases, [Wei92, Wei93, Wei98]. It has been improved and implemented within the package QERRC of the computer algebra system MAS by Dolzmann [Dol94].

For eliminating the quantifiers from our arbitrary input formula

$$\varphi(u_1,\dots,u_m) \equiv \mathsf{Q}_1 x_1 \dots \mathsf{Q}_n x_n \psi(u_1,\dots,u_m,x_1,\dots,x_n), \quad \mathsf{Q}_i \in \{\exists,\forall\}.$$

we proceed as with the Tarski method (see Section 2.3.1) to arrive at a an equivalent formula of the following form, where F is wlog. non-empty:

$$\varphi(\mathbf{u})^* \equiv \mathsf{Q}_1 x_1 \dots \mathsf{Q}_{k-1} x_{k-1} \exists x_k \dots \exists x_n \Big(\bigwedge_{f \in F} f(\mathbf{u},\mathbf{x}) = 0 \wedge \bigwedge_{g \in G} g(\mathbf{u},\mathbf{x}) > 0 \Big).$$

While the Tarski method focussed on the innermost quantifier, we focus here on the entire innermost existential quantifier block. The relevant subproblem to be discussed is thus the following one, where we have for convenience renamed the quantified variables $x_1, \dots, x_{k-1}$ of the outer blocks to $u_{m+1}, \dots, u_{m+k-1}$:

$$\varphi(\mathbf{u})^{**} \equiv \exists x_k \dots \exists x_n \Big(\bigwedge_{f \in F} f(\mathbf{u},\mathbf{x}) = 0 \wedge \bigwedge_{g \in G} g(\mathbf{u},\mathbf{x}) > 0 \Big).$$

As the first new step compared to Tarski's procedure, we compute a *Gröbner system* [Wei92] of F. Such a system is of the form

$$\big\{ \big(\psi_1(\mathbf{u}), F_1(\mathbf{u},\mathbf{x})\big), \dots, \big(\psi_r(\mathbf{u}), F_r(\mathbf{u},\mathbf{x})\big) \big\}$$

such that whenever $\psi_i(\mathbf{a})$ holds for $\mathbf{a} \in \mathbb{R}^m$, then $F_i(\mathbf{a},\mathbf{x})$ is a Gröbner basis of $\{f_1(\mathbf{a},\mathbf{x}),\dots,f_k(\mathbf{a},\mathbf{x})\} \subseteq \mathbb{R}[\mathbf{x}]$. In addition, there holds $F_1, \dots, F_r \subseteq \operatorname{Id}(F)$. On the formula level this corresponds to passing from φ^{**} to the equivalent formula

$$\varphi^{***}(\mathbf{u}) \equiv \bigvee_{i=1}^{r} \Big(\psi_i(\mathbf{u}) \wedge \exists x_k \dots \exists x_n \Big(\bigwedge_{f \in F_i} f(\mathbf{u},\mathbf{x}) = 0 \wedge \bigwedge_{g \in G} g(\mathbf{u},\mathbf{x}) > 0 \Big) \Big).$$

At this point, we restrict our attention to quantifier elimination for a single branch

$$\varphi_i^{***}(\mathbf{u}) \equiv \psi_i(\mathbf{u}) \wedge \exists x_k \dots \exists x_n \Big(\bigwedge_{f \in F_i} f(\mathbf{u},\mathbf{x}) = 0 \wedge \bigwedge_{g \in G} g(\mathbf{u},\mathbf{x}) > 0 \Big)$$

of the disjunction φ^{***}. Note that the condition ψ_i ensures that F_i is a Gröbner basis of $\operatorname{Id}(F)$ if φ_i^{***} holds, and that in this case we know the head terms of the polynomials in F_i, which are uniformly determined. We thus can uniformly determine the dimension of $\operatorname{Id}(F_i)$.

Let us for the moment assume that this dimension is zero. Then we can determine for $\gamma \in \Gamma(G)$ the matrices Q_γ of Theorem 2.13. In contrast to the situation of the theorem, however, the matrix elements are not real numbers but polynomials in $u_1, \ldots, u_{m+k-1}$.

For a univariate polynomial p denote by $\tau(p)$ the number of positive real roots of p minus the number of negative real roots of p counting multiplicities. For quadratic forms Q it is well-known that $\mathrm{sig}(Q) = \tau(\chi_Q)$. Moreover, we have that

$$\sum_{\gamma\in\Gamma(G_i)} \tau(\chi_{Q_\gamma}) = \tau\Big(\prod_{\gamma\in\Gamma(G_i)} \chi_{Q_\gamma}\Big)$$

reducing the computation of the sum of signatures to that of $\tau(\prod \chi_{Q_\gamma})$ for a single univariate polynomial $\prod \chi_{Q_\gamma}$ which has only real roots. We have to construct a quantifier-free formula stating that $\tau(\prod \chi_{Q_\gamma}) \neq 0$. This can be done by applying Descartes' rule of signs discussed in Section 2.2 to all possible combinations of signs of the coefficients of $\prod \chi_{Q_\gamma}$.

It remains to be clarified how to proceed in the case that $\mathrm{Id}(G_i)$ has a dimension greater than zero: We then compute, uniformly in $u_1, \ldots, u_m$, a maximally independent set $Y \subseteq \{x_k, \ldots, x_n\}$. Since the elimination ideal of $\mathrm{Id}(G_i)$ wrt. $\{x_k, \ldots, x_n\} \setminus Y$ is zero-dimensional, we can eliminate all quantifiers $\exists x_i$ for $x_i \notin Y$ considering all $x_j \in Y$ as additional parameters.

In the procedure above, it can happen that already $Y = \{x_k, \ldots, x_n\}$. That is, there is no *non-trivial* equation under the condition ψ_i. It is, however, always possible to replace φ_i^{***} by an equivalent formula

$$\varphi_i^{****}(\mathbf{u}) \equiv \bigvee_j \Big(\psi_i(\mathbf{u}) \wedge \exists x_k \ldots \exists x_n \Big(a_j(\mathbf{u}, \mathbf{x}) = 0 \wedge \bigwedge_{g\in G} g(\mathbf{u}, \mathbf{x}) > 0\Big)\Big)$$

such that restarting the whole elimination procedure on φ_i^{****}, we can eliminate at least one quantifier in the next step.

Introducing several sophisticated modifications, Dolzmann has improved the practical complexity of this method considerably. First, it is not necessary to consider all permutations of signs of coefficients of $\prod \chi_{Q_\gamma}$. Furthermore, inequalities $g \neq 0$ can be treated much more efficiently than by coding them as $g > 0 \vee -g > 0$ or $g^2 > 0$. See [Dol94] for details.

Like CAD, this approach is a general real quantifier elimination procedure. In practice a successful application requires that the system of equations in the input is *globally* zero-dimensional, i.e. zero-dimensional for all real parameter values. Moreover there should be at most one order side condition. This class of inputs is nevertheless mathematically quite interesting and comprises e.g. most implicitization problems for parametric real varieties. Similar to the virtual substitution method, the number of parameters plays a minor role for complexity compared to that of the quantified variables.

2.4 Variants of Quantifier Elimination

This section describes two variants of quantifier elimination in the framework of the virtual substitution method. *Extended quantifier elimination* differs from regular quantifier elimination by providing some extra information. *Generic quantifier elimination*, in contrast, weakens the notion of quantifier elimination by providing a result that is only valid under certain assumptions. These assumptions are, however, also explicitly provided by the generic quantifier elimination procedure.

2.4.1 Extended Quantifier Elimination

Roughly speaking, *extended quantifier elimination* differs from regular quantifier elimination by the fact that the procedure provides—besides a quantifier-free equivalent to the input formula—sample parametric answers for values of the variables in the outermost quantifier block. In the most frequent case of a purely existential input formula extended quantifier elimination can be construed as parametric constraint solving.

Recall from the description of the virtual substitution method in Section 2.3.3 that the result of the elimination for a block of existential quantifiers is obtained by disjunctively substituting certain test terms. It can be declaratively described as

$$\exists x_1 \cdots \exists x_k \psi(\mathbf{u}) \longleftrightarrow \bigvee_{(t_1,\dots t_k)\in\mathcal{T}(\mathbf{u})} \psi[x_1/t_1,\dots,x_k/t_k](\mathbf{u}),$$

where the set $\mathcal{T}$ is a set of terms only in the parameters possibly involving surds, $\pm\infty$, and $\pm\varepsilon$. This set $\mathcal{T}$ can be obtained by keeping track of the successive substitutions for the various variables x_k, ..., x_1 and then performing back-substitution.

Since the elimination of universal quantifiers is reduced to that of existential ones via double negation, the elimination result for this case looks analogously as follows:

$$\forall x_1 \cdots \forall x_k \psi(\mathbf{u}) \longleftrightarrow \bigwedge_{(t_1,\dots t_k)\in\mathcal{T}(\mathbf{u})} \psi[x_1/t_1,\dots,x_k/t_k](\mathbf{u}).$$

Extended quantifier elimination will first eliminate all inner variable blocks in the regular manner. For the outermost block, it will, in contrast, construct the set $\mathcal{T}$ and output—instead of the quantifier-free equivalent—a scheme of the following form:

$$\begin{bmatrix} \psi[x_1/t_1^{(1)},\dots,x_k/t_k^{(1)}] & \{x_1 = t_1^{(1)},\dots,x_k = t_k^{(1)}\} \\ \vdots & \vdots \\ \psi[x_1/t_1^{(|\mathcal{T}|)},\dots,x_k/t_k^{(|\mathcal{T}|)}] & \{x_1 = t_1^{(|\mathcal{T}|)},\dots,x_k = t_k^{(|\mathcal{T}|)}\} \end{bmatrix}.$$

The semantics of this scheme depends on the nature of the outermost quantifier block. For an $\exists$-block, the disjunction over the left column is obviously a quantifier-free equivalent for the input formula. Moreover, we can assert the following for fixed real values of the parameters u_1, ..., u_m: All the formulas in the left column are either "true" or "false." If any of them is "true," then the original quantified formula is "true" for our choice of parameters, and the corresponding right hand side entry in the scheme is *one* satisfying point.

The case of an $\forall$-block is dual to this: Here, the conjunction over the left column is a quantifier-free equivalent for the input formula. Whenever for fixed parameters one of the left hand side entries is "false," then the original quantified formula is "false," and the corresponding right hand side entry provides *one* counterexample against its satisfiability for the chosen parameter values.

We have already mentioned that the sample points t_i can—besides surds—include the symbols ∞ and ε. The former has to be read as "a real number x_0 such all $x \geqslant x_0$ satisfy ψ." In an analogous way, ε stands for a "small enough" positive real. Unfortunately, nothing can be said about the order between several non-standard symbols in the output. For identifying equal non-standard symbols after resubstitution, all such symbols are indexed ε_1, ε_2, ... and ∞_1, ∞_2, ..., respectively.

2.4.2 Generic Quantifier Elimination

Generic quantifier elimination is a variant of the virtual substitution method developed by the author for geometric theorem proving. It has been first published in 1996 [DSW98a]. Meanwhile, it has turned out that this variant is extremely suitable also for engineering applications [Stu99]. Generic quantifier elimination generates, besides a quantifier-free formula, additional assumptions in the form of negated equations stating that certain expressions in the parameters are non-zero.

We have seen in Section 2.3.3 that such assumptions can avoid case distinctions on the vanishing of parametric head coefficients. Consider quadratic polynomial $ax^2 + bx + c$, where the term a does involve some of the parameters u_1, ..., u_m but not any of the other quantified variables x_1, ..., x_n. Let α_1 and α_2 by solutions of the corresponding quadratic equations according to the solution formula, and let t_1 and t_2 be corresponding candidate points obtained from α_1 and α_2 by possibly adding or subtracting an infinitesimal ε. Then instead of the case distinction in Formula (2.1) on page 21, we simply produce the much simpler formula

$$\Delta \geqslant 0 \wedge (\psi^*[x/t_1] \vee \psi^*[x/t_2]),$$

which ignores the case that a might be zero for certain interpretations of the parameters, plus an external assumption $a \neq 0$.

Intermediate simplification might now equivalently replace a discriminant condition $\Delta(u_1, \ldots, u_m) \geqslant 0$ in the parameters by an equation $\Delta'(u_1, \ldots, u_m) = 0$. For instance, $-f(u_1, \ldots, u_m)^2 \geqslant 0$ is equivalent to $f(u_1, \ldots, u_m) = 0$. This equation is then treated in the same way as the coefficient conditions above. That is, the entire case is dropped in favor of an additional external assumption $\Delta(u_1, \ldots, u_m) \neq 0$ in the parameters.

Non-zeroness assumptions on the parameters can also enable the Gauss eliminations discussed in Section 2.3.3. When there is $\exists x$ to be eliminated from a formula $bx + c = 0 \wedge \psi^{**}$ but none of the coefficients b, c is a non-zero rational constant, then there is checked whether any of these coefficients is a polynomial only in the parameters u_1, ..., u_m. If so, say b does not contain any quantified variable, we add $b \neq 0$ to our assumptions, and then perform Gauss elimination. For quadratic equations we proceed analogously.

There are situations where we have to decide between a linear Gauss elimination with a non-zero assumption on a coefficient and a quadratic one without such an assumption. Consider for instance the formula

$$\exists x_2 \exists x_1 \big(ux_1 + x_2 = 0 \wedge x_1^2 + x_2 x_1 + u = 0 \wedge \psi^{**}(x_1, x_2, u)\big).$$

In spite of the necessary assumption $u \neq 0$ we then prefer the linear Gauss application for the elimination of x_1 because the quadratic equation would produce square root expressions the substitution of which into ψ_1 may increase the degree of x_2.

Consider a possible Gauss application with assumption for a variable x, where there are several coefficients that can be assumed to be non-zero. We then select the coefficient belonging to the highest power of x. It is not hard to see that this option saves conditions or cases in the corresponding substitution.

Note that all assumptions ever made have the form of negated equations. That is, our set of automatic assumptions cannot become inconsistent. Moreover we only neglect a set of parameter values of measure zero. In practice, it turns out that in most cases the assumptions made are reasonable and interpretable. For a detailed discussion of the nature of such *subsidiary conditions*, see Chapter 3. As an option, we can restrict the set of allowed assumptions to those being of the form $\prod_i u_i^{e_i} \neq 0$, which is equivalent to $\bigwedge_i u_i \neq 0$. This is typically used to avoid subsidiary conditions that cannot be interpreted. A good example for this feature is given in the following chapter by Example 3.20 on page 69.

Before making an assumption, there is checked whether the desired non-zeroness can be derived from the assumptions already made before. Theoretically this can be done by decision via quantifier elimination. In practice, there is instead a heuristic test applied, which is based on the simplifier, see [DS97c] for details. With this setting, it is straightforward to specify that generic quantifier elimination allows a collection of assumptions, which are atomic formulas but not necessarily restricted to negated equations, already as *input theory.* This input theory is then extended by the procedure. This feature should, however, be used with caution: Although we have argued that the set of automatic assumptions cannot become inconsistent, it can, of course, contradict the set of constraints in the input theory. If this happens, the elimination result becomes meaningless.

Definition 2.14 (Theory) A *theory* $\vartheta(\mathbf{u})$ is a conjunction of atomic formulas considered to hold. Given two theories $\vartheta(\mathbf{u})$ and $\vartheta'(\mathbf{u})$, we say that ϑ' *extends* ϑ if $\forall u_1 \dots \forall u_m \big(\vartheta'(u_1, \dots, u_m) \longrightarrow \vartheta(u_1, \dots, u_m)\big)$.

Definition 2.15 (Generic quantifier elimination) Generic quantifier elimination expects a quantified input formula $\varphi(\mathbf{u}) \equiv \mathsf{Q}_1 x_1 \dots \mathsf{Q}_n x_n \psi(\mathbf{u}, \mathbf{x})$ and optionally an *input theory* $\vartheta(\mathbf{u})$, where $\vartheta \equiv \mathrm{true}$ by default. The output consists of a quantifier-free formula $\varphi'(\mathbf{u})$ and an extended theory $\vartheta'(\mathbf{u})$. This ϑ' is obtained from ϑ by conjunctively adding negated equations. It implies the correctness of the elimination result φ' over the reals:

$$\forall u_1 \dots \forall u_m \big(\vartheta'(u_1, \dots, u_m) \longrightarrow (\varphi(u_1, \dots, u_m) \longleftrightarrow \varphi'(u_1, \dots, u_m))\big).$$

Note that we cannot and do not make any assumptions on the quantified variables $x_1, \dots, x_n$. For illustrating examples on generic quantifier elimination by virtual substitution see Section 3.7.

2.5 Simplification Strategies

From the discussion of the virtual substitution method in Section 2.3.3, it is obvious that powerful simplification methods are crucial for the successful application of this method. Simplification of formulas over the reals have been discussed in detail in [DS97c].

One generally has to distinguish between two types of such simplifiers:

1. A fast simplifier doing the rough work. This should be efficient enough to be frequently applied to intermediate results during the elimination process. The REDLOG system described in the following section includes such a simplifier, which has been discussed in detail in [DS97c]. It is usually referred to as the *standard simplifier.*

2. Various dedicated special-purpose simplifiers that are used to increased the quality of the final result. Here, efficiency is only a secondary issue.

One prominent example for a sophisticated simplifier of the second kind is the *Gröbner simplifier* introduced in [DS97c]. Since it plays an important role for the examples discussed throughout this thesis, we are going to sketch the principle ideas behind it by means of a simple example.

Example 2.16 (Gröbner simplification) Consider the disjunction $xy-1 \neq 0 \vee yz-1 \neq 0 \vee x-z=0$. This can be rewritten as an implication:

$$xy-1=0 \wedge yz-1=0 \longrightarrow x-z=0.$$

Reducing $x-z$ modulo a Gröbner basis $\{yz-1, x-z\}$ of the left hand side polynomials $\{xy-1, yz-1\}$, the right hand side equation and thus the entire formula can be simplified to "true." □

Gröbner bases have been introduced by Buchberger [Buc65], for details on Gröbner basis theory see e.g. [BWK93]. The exact Gröbner simplification algorithm has been described in [DS97c]. Note that it can be applied to general quantifier-free formulas over the reals by computing a CNF and then treating the contained disjunctions independently. There is also a dual variant applicable to conjunctions and thus to DNF's. The Gröbner simplifier implementation in REDLOG includes strategies for automatic decision between CNF and DNF computation.

2.6 The REDLOG Computer Logic System

Based on experimental implementations beginning 1992, the author started the realization of REDLOG in early 1994. In April 1995, the system has been redesigned by the author together with A. Dolzmann [DS97b], who from then on participated to same parts in the realization of REDLOG. REDLOG 1.0 [DS96] has been published on the Web in October 1996. This first REDLOG distribution was a great success. In October 1996 both the author and A. Dolzmann joined the REDUCE development group such that REDLOG is now developed not as a separate contributed package but as part the REDUCE system itself. The current version REDLOG 2.0 [DS99] is part of the distribution of REDUCE 3.7 issued in April 1999.

REDLOG is widely accepted inside the scientific community. For applications of REDLOG inside mathematics and computer science see e.g. [ADY+96, DYA97, HLS97, KW, LPY99], for applications in mechanics see [Ioa99].

The following description of the REDLOG system supposes that the *Portable Standard Lisp* (PSL) version of REDUCE is used. For the *Cambridge Standard Lisp* (CSL) version, some details are slightly different.

2.6.1 Overview

REDLOG stands for "REDUCE logic" system. It provides an extension of the computer algebra system REDUCE to a computer logic system implementing symbolic algorithms on first-order formulas wrt. temporarily fixed first-order languages and theories. Underlying theories currently available are

1. algebraically closed fields ("complex numbers")
2. real closed fields ("real numbers with order")
3. discretely valued fields ("p-adic numbers").

REDLOG origins from the implementation of real quantifier elimination by virtual substitution. Successfully applying such methods to both academic and real-world problems, the authors have developed a large collection of formula-manipulating tools, many of which are meanwhile interesting in their own right:

- Numerous tools for comfortably inputing, decomposing, and analyzing formulas.
- Several techniques for the logical and algebraic simplification of formulas.
- Various normal form computations.
- Quantifier elimination by virtual substitution.
- Generic quantifier elimination by virtual substitution.
- Extended (generic) quantifier elimination by virtual substitution.
- An interface to Hong's QEPCAD implementing partial cylindrical algebraic decomposition.
- An interface to Dolzmann's QERRC implementing quantifier elimination by parametric real root counting.
- Linear optimization techniques based on quantifier elimination.

2.6.2 From Computer Algebra to Computer Logic

The aim of REDLOG is to make quantifier elimination procedures available for solving practical problems in various application areas, in particular in mathematics, physics, and engineering. The decision for taking an existing computer algebra system like REDUCE as basis has the following advantages compared to a completely new implementation:

- A computer logic system user interface requires no concepts that go beyond that of computer algebra system interfaces. Hence there is no reason for spending time designing and implementing yet another interface. In addition, the large community of REDUCE users has quickly become familiar with REDLOG.
- The underlying system provides a reliable well-tested implementation of polynomials, which can serve as first-order terms in many languages. In addition, there is a large library of up-to-date algebraic algorithms available.

- There is no need for portability considerations. The system will simply run on the large variety of architectures and operation systems to which REDUCE is ported.

On the other hand, the underlying computer algebra system itself can benefit from the implementation of first-order formulas and corresponding algorithms. On example for this is the implementation of *guarded expressions* [DS97a]. Another possible application is with the widespread `solve` operator for solving systems of equations and, possibly, inequalities. These solutions to such systems can be conjunctions as for $x^2 < 25$, disjunctions as for $x^2 > 25$, or more complicated formulas. The solutions are typically given as nested lists encoding DNF's. Here, the use of formulas would be much more natural. At this point, one should mention that MATHEMATICA actually offers the option for printing the result as formula. Except for distributive expansion, there are, however, no algorithms operating on these formulas available [Wol96].

2.6.3 A User's View on the System

REDUCE know two *modes*: the *algebraic* mode, which provides the user frontend, and the *symbolic* mode, which provides the application programming interface. The symbolic mode is essentially a Lisp system within which the major part of the REDUCE functions is implemented. We first focus on the algebraic mode use of REDLOG.

2.6.3.1 Contexts

As a first step when working with REDLOG, a *context* has to be selected. This context determines a language and a theory in the sense of first-order logic. All computations are performed wrt. the selected context until a different decision is made.

When the context is changed, formulas produced in the old context can become invalid, but they need not. Certain formulas of the old context may still be meaningful in the new context. Consider for instance a formula produced in an ordered field context: If it happens to contain no ordering relation at all, then it can be reused within the algebraically closed fields context. Necessary conversions of its internal representation are applied implicitly and invisible for the user.

After fixing the context, the REDUCE functionality is extended in two ways:

1. In addition to the built-in data types such as rational functions or matrices, one can input first-order formulas over the language specified by the selected context.
2. There are new procedures available that apply to these first-order formulas.

In the sequel, we assume that the ordered field context "`ofsf`" has been selected. It knows the binary predicates "$=$," "$\neq$," "$\leqslant$," "$\geqslant$," "$<$," and "$>$."

2.6.3.2 Expressions

REDLOG extends the look-and-feel of REDUCE to first-order formulas. Invalid expressions are detected, and appropriate error messages are given.

The format for the truth values, quantifiers, and propositional connectives is specified uniformly for all contexts. Besides the reserved identifiers `true` and `false`, there are the following operators: a unary `not`, binary infix `impl`, `repl`, `equiv`, and n-ary infix `and`, `or`. Binary prefix operators `ex` and `all` serve as quantifiers. Their first argument is a variable, and their second argument is a formula.

In general, all atomic formulas are constructed with operators that are considered as predicates. Here again infix operators are possible. It is left to the context to determine which predicates are known and what the terms are. Furthermore, a context can impose some extra restrictions on the form of the atomic formulas. Consider for instance `ofsf`: The known predicates given at the end of the last subsection can all be written infix. Terms are polynomials over the integers. As an additional restriction, all right hand sides of the predicates must be zero.

The handling of the input is much more liberal than the specification of valid expressions. For the easy input of large systematic conjunctions and disjunctions there are special `for`-loops available. With the quantifiers `ex` and `all`, the first argument may be a list of variables. In `ofsf` the input may contain rational coefficients and non-zero right hand sides. Furthermore, `ofsf` allows the input of *chains* such as `a<>b<c>d=f`, which is turned into `a-b<>0 and b-c<0 and c-d>0 and d-f=0`. Here, only adjacent terms are considered to be related. With all these input facilities, the input is implicitly converted to expressions matching the specification of the current context.

2.6.3.3 Simplification vs. Evaluation

Polynomials and rational functions entered into REDUCE are automatically converted into some canonical form. Canonical means that expressions that are mathematically *equal* are converted into syntactically equal forms. We refer to this conversion as *evaluation*.

The natural extension of evaluation to first-order formulas would be converting *equivalent* formulas into syntactically equal forms. This is impossible, in general, since in non-recursive structures there is no algorithm converting sentences that hold into `true`. Even in most recursive structures it is by no means obvious, how to obtain suitable canonical forms for formulas containing free variables. Moreover, one would require for such normal forms that they are user-friendly and fast to compute.

Instead of evaluation, we use the weaker concept of simplification as discussed in Section 2.5. That is, we optionally replace formulas by equivalents that are more user-friendly though not canonical.

Similar to most other computer algebra systems, REDUCE interpreter variables are identified with the transcendental elements occurring in rational functions. With first-order formulas such an identification leads to problems, since for many contexts the terms included in the first-order formulas will be implemented as rational functions or some suitable subset. One would certainly like to as usual identify the indeterminates occurring inside these terms with the interface variables. If, however, such an indeterminate is quantified within a first-order formula, then any assignment of a non-indeterminate rational function to it violates the well-formedness of this formula. The problem of expressions becoming ill-formed due to subsequent assignments is not really new: This also happens with the rational function $1/x$ when assigning $x := 0$. To avoid confusion, we invalidate a formula in case of *any* assignment to a quantified variable even if the assigned expression is another variable.

A similar problem arises from the fact that interpreter variables interpreted as indeterminates are valid rational functions, but they are, in contrast, not valid formulas. Hence the user is not allowed to enter things like

```
f := ex(x,g); g := x>0;
```

such that afterwards `f` be evaluated to `ex(x,x>0)`. It is, however, possible to input the above statements in reverse order. Note: The expansion of interpreter variables should not be confused with *substitution*, which is, of course, implemented for arbitrary quantified formulas meeting the usual mathematical specifications.

2.6.3.4 Procedures

In order to avoid name conflicts, all REDLOG procedures and switches available in the algebraic mode are prefixed by `rl`. The procedure names are to be understood declaratively, e.g., `rlqe` stands for "apply the default procedure for quantifier elimination." The algorithm actually applied depends on the selected context. This pattern makes REDLOG easy to learn. Moreover, it allows to combine procedure calls to new algebraic mode procedures that do not depend on the context.

As usual in REDUCE we have a fairly liberal syntax including optional arguments with default values, procedures expecting either an expression of a certain type or a list of such expressions. Most of the REDLOG algorithms offer numerous options. Options are selected by setting corresponding *switches*, which are global boolean values selected by the commands `on` and `off`. The return values of the REDLOG procedures are designed to cooperate with the standard REDUCE. Extended quantifier elimination, for instance, returns the coordinates of each sample points as a list of equations because there are numerous built-in algorithms that operate on such lists.

Some procedures provide the option to protocol the progress of the computation onto the screen. We refer to this as *verbosity* output. In future versions there might be different levels of such output. It is specified, however, that there is *one* switch, namely `rlverbose`, by which all verbosity output can be turned off.

2.6.4 An Implementor's View on the System

REDLOG is implemented in the symbolic mode of REDUCE. The language there is the Lisp dialect RLISP, which is compiled to native machine code. The module structure of REDLOG is pictured in Figure 2.1.

2.6.4.1 The Algebraic Mode Interface

The algebraic mode interface implements the user frontend described in the previous Section 2.6.3. It performs all necessary conversion into the internal, i.e. symbolic mode, representation, and includes error checking. It furthermore handles optional parameters including the substitution of default values where

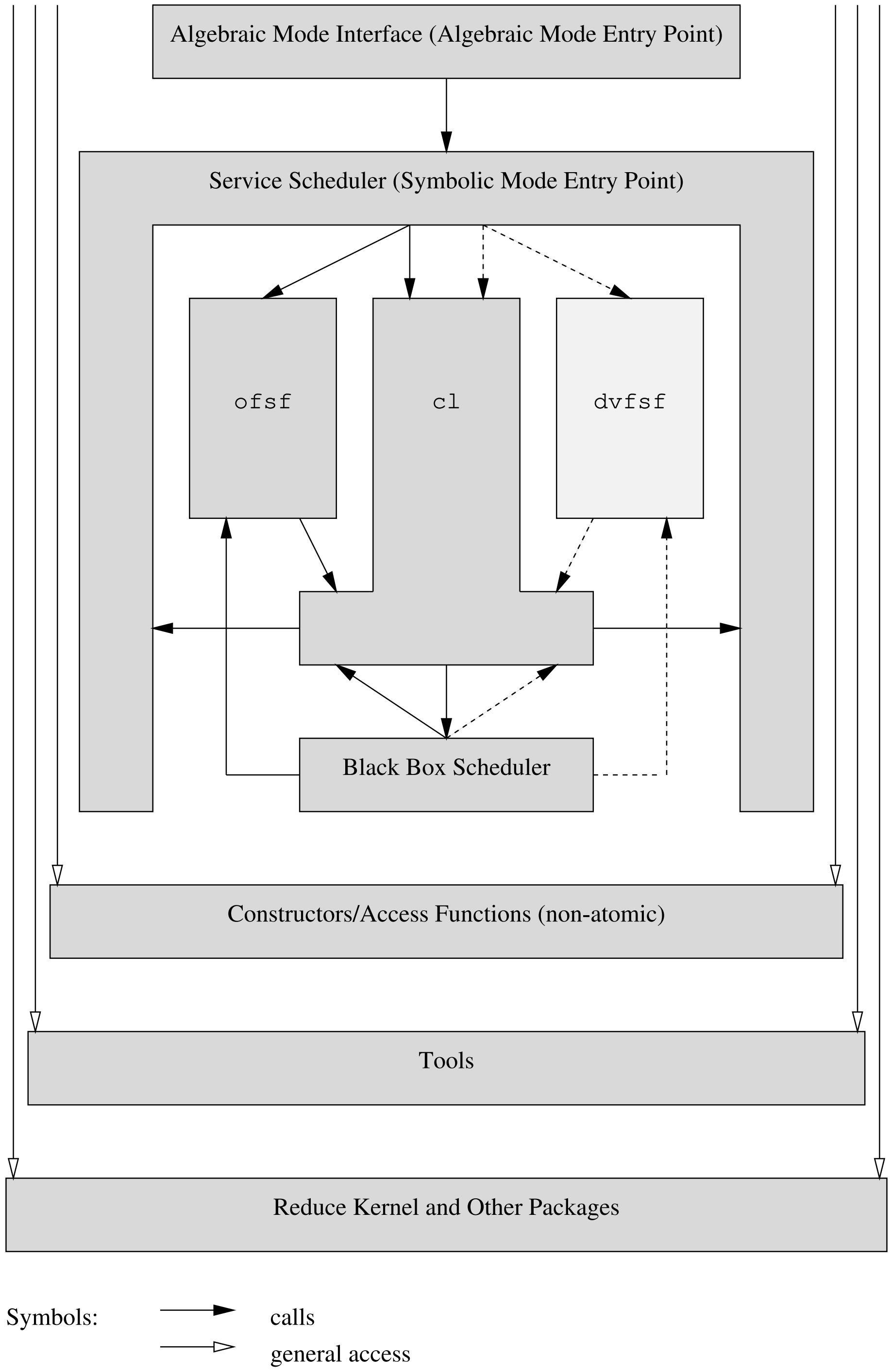

Figure 2.1: The module structure of REDLOG with the context **ofsf** active. Dotted arrows indicate alternative calls with **dvfsf** active.

necessary. Algebraic mode procedure calls are translated to calling the corresponding symbolic mode entry points with correct input parameters.

At every formula there is a *context tag* which is the context identifier of the context the formula has been created in. Comparing the context tag with the current context identifier enables the automatic conversion mentioned in the paragraph on contexts above. Besides the conversion of the user input, which appears as Lisp prefix, and the conversion of formulas from foreign contexts, there are *equations* to be converted to *logicals*, i.e. formulas, and vice versa. Due a very special role that equations play in REDUCE, the algebraic mode interface accepts and returns equations wherever possible. The symbolic mode entry points, in contrast, insist on logicals.

2.6.4.2 Formula Representation

In Lisp-based systems there is a straightforward way to represent *terms* in the sense of universal algebra: The prefix term representation $\omega^{(n)}(t_1, \dots, t_n)$ of a term with operator symbol ω of arity n and argument terms $t_1, \dots, t_n$ is mapped to a list by the *Lisp prefix representation* Λ, which is recursively defined as follows:

$$\begin{aligned} \Lambda(\omega^{(0)}) &= \omega \\ \Lambda\big(\omega^{(n)}(t_1, \dots, t_n)\big) &= \big(\omega, \Lambda(t_1), \dots, \Lambda(t_n)\big) \quad \text{for } n > 0. \end{aligned}$$

This representation corresponds in a beautiful manner to the internal representation of the Lisp language itself. It is thus extremely natural and easy to process. On the other hand, Lisp prefix has turned out to be both time and space consuming for representing terms describing arithmetic such as polynomials or rational functions. For such data, more efficient Lisp representations have evolved. In REDUCE this is namely *standard forms* for polynomials and *standard quotients* for rational functions. A multivariate polynomial $f \in \mathbb{Z}[X_1, \dots, X_n]$ is represented as a standard form $\Phi(f)$ as follows:

$$\begin{aligned} \Phi(0) &= \mathtt{nil} \\ \Phi(z) &= z \quad \text{for } z \in \mathbb{Z} \setminus \{0\} \\ \Phi\Big(\sum_{i=0}^{k} f_i X_n^i\Big) &= \Big(\big((X_n \ . \ k) \ . \ \Phi(f_k)\big) \ . \ \Phi\Big(\sum_{i=0}^{k-1} f_i X_n^i\Big)\Big), \end{aligned}$$

where $f_0, \dots, f_k \in \mathbb{Z}[X_1, \dots, X_{n-1}]$ and wlog. $f_k \neq 0$. The dot "." denotes the usual `cons` operation on Lisp cells. The standard form representation requires an internal ordering on the variables. For a fixed such ordering it is unique. The standard quotient representation of a rational function $p/q \in \mathbb{Z}(X_1, \dots, X_n)$ is defined as $\big(\Phi(p) \ . \ \Phi(q)\big)$.

The non-atomic part of the REDLOG formulas is kept as Lisp prefix. The same holds for the toplevel structure of the atomic formulas. The term representation, in contrast, depends on the context. The quantified variables occurring as first argument to the quantifiers are always represented as identifiers. In the ordered field context `ofsf` the terms are kept as standard forms, which explains its name *ordered field standard form*. Note that the representation of a term consisting of a single variable is here different from the representation of that variable as argument to a quantifier: for the variable x we have, e.g., $\big((\mathtt{x}\ .\ 1)\ .\ \mathtt{nil}\big)$ vs. `x`.

The Lisp prefix specification of the non-atomic part is hidden via *constructors* and *access functions*. Any direct access to formulas via Lisp operations is prohibited. The same applies to the atomic formulas in the various contexts.

2.6.4.3 The Common Logic Module

The common logic module `cl` contains code that operates on first order formulas without making any assumption about the current context. It contains three types of procedures. First, there are procedures that need not know the context as for instance `cl_atnum`, which given a first-order formula counts the number of atomic formulas contained in it. This works because every formula that does not have a quantifier or a known propositional connective as top-level operator is classified as an atomic formula.

Second there are procedures that split into a general and context dependent subtasks. For instance, when simplifying formulas there are general rules such as deleting `false` from disjunctions or replacing disjunctions that contain `true` by the latter. On the other hand, one would design a simplifier also to replace certain atomic formulas by simpler equivalents according to rules that depend on the context. In this situation, the general techniques are desirable for many contexts. There are several reasons for implementing them only once, and parameterizing them in some way with context dependent subalgorithms:

- New contexts can be implemented faster and safer, thus saving time for redeveloping and debugging the first-order parts of the algorithms.
- All contexts will immediately profit from future algorithmic extensions and tuning of the general parts.

We say that `cl` uses *black boxes* for the context dependent tasks, and, accordingly, we denote the code addressed by a black box as *black box implementation*. Note that there can be several generic implementations for the same task corresponding to several sets of contexts.

Local operations on atomic formulas are not the only reason for introducing black boxes. Consider, for instance, the following simplification for ordered fields: $x > 0 \wedge x > 1$ becomes $x > 1$. Though obviously context dependent, this applies to a non-atomic formula.

Some black box implementations of this latter kind are simply smarter variants of context independent counterparts, such as recognizing syntactically equal atomic formulas in our particular example. All contexts for which there is no smarter version use the context independent variant implemented within `cl`. This leads to the existence of context independent black box implementations as a third group of `cl` procedures. Whenever a black box implementation is required, `cl` calls the *black box scheduler*.

2.6.4.4 The Black Box Scheduler

The black box scheduler knows the current context. It performs the mapping between black boxes and black box implementations. The latter are either implemented within the active *context module* or within `cl` itself as described above. The black box scheduler is accessed only by `cl` and hence can be viewed as a part of it. In practice, the black box scheduler is a collection of procedures which branch into the code valid for the current context.

2.6.4.5 The Context Modules

In the Figure 2.1 on page 33, there are two modules, `ofsf` and `dvfsf`, which serve as examples for *context modules*. We have defined a context as the specification of a language, a theory, and a term representation.

There is no separation between these aspects, which has some restricting consequences: First, there is in particular no parameterization wrt. the term representation. This is reasonable, because the standard form representation is quite canonical in REDUCE if one decides to keep the terms as polynomials. Keeping the terms multiplicatively, i.e., as lists of irreducible factors leads to completely different algorithmic ideas, which cannot be covered by simply redefining constructors and access functions for terms. Second, there is also no hierarchy of context modules, i.e., there is no ordered field context based on a field context.

For a given context, the corresponding context module contains all context dependent code, which splits into the following three parts:

1. Context dependent algebraic mode interface code
2. Black box implementations for `cl` procedures.
3. Complete algorithm implementations, i.e., context specific implementations that cannot be split into a general part and context dependent part in a reasonable economic way.

Of course, the completely implemented algorithms are not necessarily disjoint from the algorithms for which there is a `cl` implementation. Which implementations are actually used is hidden from the user who calls the `rl`-procedures from a declarative point of view (cf. Section 2.6.3.4). Mapping calls to `rl`-prefixed procedures from both the algebraic mode and the symbolic mode to actual `cl` or context module implementations is the job of the *service scheduler*.

2.6.4.6 The Service Scheduler

The service scheduler implements the map between declarative computation requests on one hand and the corresponding implementations valid in the current context on the other hand. Like the black box scheduler, it consists of a collection of procedures. These procedures establish the symbolic mode interface to REDLOG. They are also accessed by the algebraic mode interface which is implemented as a front-end to the symbolic mode interface.

Within REDLOG, the service scheduler is called from inside the `cl` module for accessing services, which may be implemented either in a context module, or in turn within `cl` itself. Consider, for instance, intermediate result simplification within a `cl` implementation of quantifier elimination. Though there is a simplifier implemented within `cl`, there is no guarantee that this simplifier is valid for all contexts in which the `cl` elimination code is used.

On the other hand, there are cases where `cl` should call `cl` procedures immediately. An obvious example is a recursive procedure. A more general one is a hierarchy of subprocedures built for the implementation of one or several services. In the second case, it is already possible that there is a better context dependent alternative for one of the subprocedures. It is left to the implementor to decide how to proceed in particular cases. When calling `cl` code immediately, one must, however, ensure not to address code for which there are black box implementations missing, since not all the `cl` procedures are applicable in all contexts.

The context module procedures call all services immediately. There is no reason for them to call the service scheduler because they know the current context. For clarity, we have thus specified that they are not allowed to do so.

2.6.4.7 Tools

The tool modules contain functions required for the implementation which do not apply to logicals, but are of a more common nature. They are also used for implementing low-level code shared by different context modules. Recall that various context modules certainly cannot call one another.

2.6.5 Scheduler Internals

We have already discussed that, at any time, there is one context fixed. All input processing and all computations are performed for this context. This allows the automatic formula conversion between contexts mentioned in Subsection 2.6.3.1. The schedulers are designed on the assumption that there is always some global information on the current context available.

2.6.5.1 Black Box Scheduling

We first focus on black box scheduling within the `cl` module. There are essentially three different ways to perform this task:

1. A data-driven model using tagged forms.

2. A static approach using a global environment.

3. A functional model using parameter functions.

A tag is a certain identifier encoding a data type. Data type specific operations are stored in the property list of the tag. In Lisp based implementations it is a usual technique to have such a tag combined with every data object. A typical tag position is the first entry in the list representation of each data object. In our case, this would amount to keeping the context tag at the beginning of every formula. Informally spoken, the formula "knows" which black box implementations are appropriate for it. The scheduler looks at the tag, calls the implementation found there, and then returns the result.

Keeping the context information at every formula is not necessary for our purposes because all computations are performed wrt. the current context. In particular, there are never formulas combined that belong to different contexts. Such redundancy of type information would in fact even cause problems: Many of the implemented algorithms are inherently recursive. When entering a formula recursively, one would have to copy tags to subformulas for recursion, and in turn drop tags when recombining the recursion results. One way to overcome this is saving the tag globally at the beginning, do the recursion with tag-free formulas, and retag the final result at the end.

This naturally leads us to the second approach, which is actually used in REDLOG. All information stored in the property list of the tag in the first approach is instead stored in the property list of the context identifier. At the formulas there is no context information at all. Accordingly, the black box scheduler accesses the context identifier instead of the tag.

Since searching the property list of the context identifier has turned out time consuming, we use a *cache* technique: The information, which is basically an *associative list* mapping black boxes to implementations, is copied into a set of global variables, i.e., one variable for each black box. The functions that make up the scheduler *apply* the values of their corresponding variables to their parameters. The cache is updated at every context selection.

A third technique, which has been used in very early versions of REDLOG, uses *parameter functions*: The names of the required black box implementations are passed to the `cl` procedures as extra parameters. The `cl` procedures address the black box implementations by *applying* the parameters. Then there is no black box scheduler module at all. This approach had several disadvantages: First, due to the extra parameters the code is less readable. Second, the extra parameters have to be passed down during recursion. Third, when new black box algorithms and hence procedure parameters are added, all corresponding procedure calls in the code have to be adapted.

2.6.5.2 Service Scheduling

The service scheduler works in the same way as the parameter scheduler does. Here, the cache technique is also adequate: Recall from Subsection 2.6.4.6 that the service scheduler does not only provide a user interface, but may also be called by `cl` procedures, possibly within loops.

In the previous paragraph we have pointed out that with parameter functions there is no black box scheduler. Using parameter functions actually means leaving the black box scheduling to the service scheduler. The latter would select the appropriate service function name (service scheduling), and call it with appropriate functional parameters if it is a `cl` function (black box scheduling).

2.6.6 Documentation

The REDLOG user manual [DS99] is written in the GNU Texinfo format from which both a printed TeX document and an HTML online documentation are created.

The program documentation is another Texinfo file generated automatically from the source code: Every procedure starts with a comment block. This is an English text consisting of complete sentences. The first few sentences are specified to have specific contents. In particular, the first sentence explains the name of the procedure, e.g., in `cl_qe`, a `cl` implementation of quantifier elimination it would be: "Common logic quantifier elimination." An alphabetic index containing the words occurring in the first sentences has shown out to be very useful with the computer algebra systems SAC and MAS. Further sentences are reserved for format descriptions of parameters, return values, and access to global variables. The comment ends with a *declarative* description of what the procedure does.

2.7 Application Areas

The application areas of quantifier elimination in general and of REDLOG in particular are by no means restricted to the geometric applications, on which this thesis focuses. For the sake of a more complete picture of applied quantifier elimination, we are going to summarize some of the non-geometric applications so far. For a more detailed overview on applied quantifier elimination see [DSW98b].

2.7.1 Constraint Solving

Constraint solving classically involves a system

$$f_1(x_1,\dots,x_n) \geqslant 0, \quad \dots, \quad f_k(x_1,\dots,x_n) \geqslant 0$$

of parameter-free non-strict polynomial inequalities. One wishes to test whether such a system is *feasible*, i.e., whether it has a solution. Moreover one often wishes to determine at least one solution if there exists one. Most classical methods, such as e.g. the *simplex method* [Dan51] or the *Fourier–Motzkin method* [Mot36], require the variables $x_1, \dots, x_n$ to occur only linearly.

In this setting, quantifier elimination applied to the formula

$$\exists x_1 \dots \exists x_n \big(f_1(x_1,\dots,x_n) \geqslant 0 \land \dots \land f_k(x_1,\dots,x_n) \geqslant 0\big)$$

will yield "true" if and only if the system is feasible. Extended quantifier elimination will in addition yield a sample solution.

Moreover quantifier elimination can be straightforwardly applied to solve the following generalizations of the problem:

- The constraint system is an arbitrary boolean combination of constraints instead of simply a conjunction.
- The coefficients of the x_i are polynomials in parameters $u_1, \dots, u_m$.
- The constraints include strict inequalities such as "$>$" and "$\neq$."
- The x_i occur with arbitrary degree. There are also arbitrary products of different x_i and x_j. Here, one possibly has to obey restrictions imposed by the quantifier elimination method used.

Recall that with strict inequalities the sample solutions provided by extended quantifier elimination by virtual substitution can involve non-standard terms containing "ε." This can in principle be overcome for the price of higher complexity [LW93]. With higher degrees, solutions can in general not be determined as closed terms but in some algebraic form.

2.7.2 Optimization

Optimization differs from constraint solving in the extra issue that a sample solution point is required for which an objective function $F(x_1,\dots,x_n)$ is minimal. The simplex algorithm [Dan51] is designed for this task with a parameter-free linear function. With the Fourier–Motzkin [Mot36] method linear objective functions are introduced via an additional constraint $z \geqslant F(x_1, \ldots, x_n)$, where z is a new variable.

This coding of objective functions by constraints can be used also with quantifier elimination methods. For instance, one can do *hyperbolic optimization*, i.e., optimization wrt. an objective function $F(\mathbf{u}, \mathbf{x}) = f(\mathbf{u}, \mathbf{x})/g(\mathbf{u}, \mathbf{x})$ for polynomials f and g and a system of polynomial constraints $\psi(\mathbf{u}, \mathbf{x})$ by eliminating

$$\exists x_1 \ldots \exists x_n \big(\psi \wedge \big((g > 0 \wedge f \leqslant zg) \vee (g < 0 \wedge f \geqslant zg)\big)\big).$$

2.7.3 Scheduling

The classical scheduling problem is as follows: *Tasks* 1, ..., n have to be performed on machines 1, ..., m. Each task i is assigned to a machine m_i and requires a processing time p_i. Moreover, there is a partial ordering "$\prec$" defined on the tasks. The problem is to determine starting times t_i for each task i such that none of the ordering constraints on the tasks is violated, no machine is occupied by more than one task at the same time, and the last task is finished at the earliest time possible.

This classical setting can be restated as a quantifier elimination problem as follows:

$$\exists t_1 \ldots \exists t_n \Big(\bigwedge_{i=1}^{n} t_i \geqslant 0 \wedge$$
$$\bigwedge_{i \prec j} t_i + p_i \leqslant t_j \wedge \bigwedge_{\substack{1 \leqslant i < j \leqslant n \\ m_i = m_j}} (t_i + p_i \leqslant t_j \vee t_j + p_j \leqslant t_i) \wedge \bigwedge_{i=1}^{n} z \geqslant t_i + p_i \Big).$$

The last conjunction of constraints involving z corresponds to the treatment of objective functions discussed with optimization.

Quantifier elimination allows to tackle the following more general types of scheduling problems:

- For each job there is given an interval of time within which it has to be performed.
- Certain machines are allowed to process several tasks in parallel.
- Certain jobs require more than one machine in parallel.
- The cost function to be minimized is an arbitrary piecewise rational function of starting times of the tasks.
- The extreme freedom in the form of constraints allows to do *hierarchical* scheduling: In two elimination steps, a second objective function is optimized under the assumption of an optimal solution wrt. a first objective function.

This includes e.g. *job shop* problems on several machines, problems requiring several units of different resources arising, e.g., in the construction of buildings, and transportation problems involving penalties for earliness or tardiness of certain tasks. Here the constraint formula will usually involve disjunctions resulting from resource restrictions.

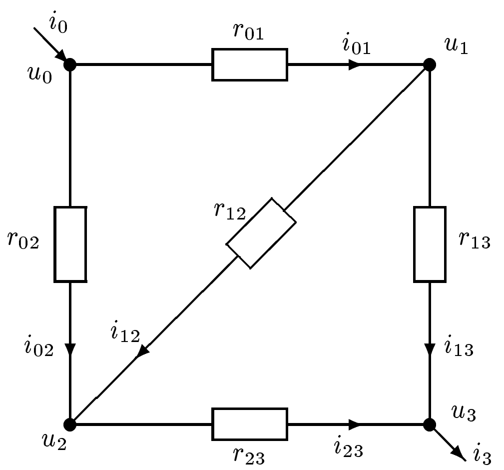

Figure 2.2: A simple example network consisting of resistors

2.7.4 Simulation, Sizing, and Diagnosis

We consider simulation, design, and error diagnosis of electrical, mechanical, or hydraulic networks.

Figure 2.2 shows a very simple electrical sample network [Wei97c]. It consists of resistances r_{jk} subject to wattage restrictions p_{jk}. For fixed resistances r_{jk} and input voltage $u_3 - u_0$, all amperages i_{jk} and voltages u_k are uniquely determined by physical laws. A description of this circuit as a formula is obtained as follows: By Ohm's law we have

$$\omega \equiv \bigwedge_{(j,k)\in N} (u_k - u_j) = r_{jk} i_{jk}, \quad N = \{(0,1), (0,2), (1,2), (1,3), (2,3)\}.$$

Next, Kirchhoff's laws state that

$$\kappa \equiv -i_0 + i_{01} + i_{02} = 0 \wedge -i_{01} + i_{12} + i_{13} = 0 \wedge$$
$$-i_{02} - i_{12} + i_{23} = 0 \wedge -i_{13} - i_{23} + i_3 = 0.$$

Furthermore, we write down the wattage restrictions for the resistors:

$$\rho \equiv \bigwedge_{(j,k)\in N} \big((u_k - u_j) i_{jk}\big)^2 \leqslant p_{jk}^2.$$

This yields an algebraic translation $\nu = \omega \wedge \kappa \wedge \rho$. Finally, we may normalize $u_0 = 0$, and we may certainly assume that $i_3 = i_0$. Since our network is intended to be a 12 V circuit, we also set $u_3 = 12$ yielding ν_0.

The easiest thing to do now is *simulation* of the circuit, i.e., we plug in values for the r_{jk}, and then eliminate all the amperages and voltages. This will yield either "true" or "false." In the latter case some wattage restriction is violated. Extended quantifier elimination would yield in addition to "true" sample values for the amperages and voltages, which are in fact the uniquely determined values assumed. More sophisticated applications of quantifier elimination in this area include the following [Stu99]:

Sizing In the example network, determine resistances such that certain constraints on the amperages and voltages are satisfied.

Error diagnosis Imagine a concrete network in which all resistances, amperages, and voltages are known. Suppose r_{12} is actually an ammeter for monitoring the network. If the amperage i_{12} changes, one determines from its new value which part of the circuit is damaged in which way.

2.7.5 Control Theory and Stability

Hong, Liska, and Steinberg have shown that certain stability problems for ODE's and PDE's can be rephrased as real quantifier elimination problems. While many of these problems are still out of the range of present day elimination methods, some non-trivial and interesting instances of such problems have been solved by QEPCAD and REDLOG; see [HLS97] for details.

Abdallah, Dorato, and Yang have also worked in this area. They have solved an open problem in control theory using QEPCAD in combination with REDLOG [ADY+96, DYA97].

Jirstrand has examined stationary, stability, and following of a polynomially parameterized curve for dynamical systems using QEPCAD [Jir97].

2.7.6 Real Implicitization

For $n < m$ let $f : \mathbb{R}^n \to \mathbb{R}^m$ be a rational map with component functions p_i/q, where p_i, $q \in \mathbb{R}[x_1, \dots, x_n]$ for $1 \leqslant i \leqslant m$. The image $f(\mathbb{R}^n)$ of f is a definable and hence semi-algebraic subset of $\mathbb{R}^m$ described by the formula

$$\exists x_1 \dots \exists x_n \big(q(x_1, \dots, x_n) \neq 0 \land \bigwedge_{i=1}^{m} p_i(x_1, \dots, x_n) = u_i q(x_1, \dots, x_n)\big).$$

The aim is then to obtain a quantifier-free description of $f(\mathbb{R}^n)$ in the variables $u_1, \dots, u_m$, preferably a single equation, which would provide an implicit definition of f.

2.7.7 Automatic Theorem Proving

Quantifier elimination can obviously be used as a decision procedure for first-order sentences. Application of generic quantifier elimination to geometric theorem proving is discussed in detail in the following chapter. The automatic proof of the following "theorem" is a nice non-geometric application, which has been introduced by Brown [Bro83] and proved using PROLOG III by Colmerauer [Col90].

Example 2.17 (Period 9) Consider the infinite sequence $(x_i)_{i \in \mathbb{N}}$ of real numbers defined by $x_{i+2} := |x_{i+1}| - x_i$, where x_1 and x_2 are arbitrary. This sequence is always periodic, and the period length is less than or equal to 9. □

A straightforward translation of this theorem to a first-order formula over the reals reads as follows:

$$\forall x_1 \forall x_2 \forall x_3 \forall x_4 \forall x_5 \forall x_6 \forall x_7 \forall x_8 \forall x_9 \forall x_{10} \forall x_{11} \Big(\bigwedge_{i=1}^{9} \big((x_{i+1} \geqslant 0 \land x_{i+2} = x_{i+1} - x_i) \lor$$

$$(x_{i+1} < 0 \wedge x_{i+2} = -x_{i+1} - x_i)\big) \longrightarrow x_1 = x_{10} \wedge x_2 = x_{11}\Big).$$

2.8 Related Work

There are two major research areas that are related to the field of real quantifier elimination in the sense that they address apparently similar problems. The first one is general approaches to automated theorem proving, and the second one is logic programming. We want to sketch the conceptual differences between these and our own work.

Automated theorem proving splits into two major approaches: resolution methods and tableau methods. In either case, the methods do not immediately make use of the semantics of the predicates contained in the sentences to be proved. More precisely, all the algebraic knowledge on, say, real closed fields, is coded by logic formulas and can then within the proof process not be distinguished from formulas that stem from the target problem. Apart from simplification issues, quantifier elimination methods operate, in contrast, nearly completely on the algebraic side. The theorem provers simply implement a logical engine, and all algebraic knowledge required for solving the input problem has to derived by this engine using the implemented calculus. Quantifier elimination procedures, in contrast implement all the necessary algebraic knowledge within the engine. This implemented knowledge has been proved in advance by humans or machines using various calculi.

Logic programming techniques as implemented in the programming language PROLOG were originally restricted variants of the resolution proof method mentioned above. More recent versions such as PROLOG III include constraint solving capabilities, which gives then indeed a certain spirit of quantifier elimination. The periodicity theorem discussed above as Example 2.17 has actually been also solved using PROLOG III. The major difference to our method is here, that solving it with PROLOG III has the flavor of a "programming job:" In order to be successful, one has to think about how PROLOG works. Quantifier elimination, in contrast, just requires a formally correct description of the problem. Interestingly, the initial motivation for logic programming languages was *declarative* problem specification as opposed to *procedural* programming. So in a certain sense, quantifier elimination is closer to the original goals of logic programming than the corresponding programming languages. Note, however, that the applications discussed throughout in this thesis suggest that there is still a certain expertise in the proper formulation of problems necessary to efficiently make use of quantifier elimination techniques.

2.9 Conclusions

We have given a detailed overview over the historical development of real quantifier elimination, which dates back to the 17th century. Wherever possible, we have figured out and cited the original sources of the algebraic methods. We have given detailed descriptions of all available implemented quantifier elimination algorithms. Due to its historical importance, we have also sketched the first published real quantifier elimination algorithm by Tarski.

Our focus is naturally on quantifier elimination by virtual substitution, which has been implemented by the author et al. within the computer logic system REDLOG that, as a part of the computer algebra system REDUCE, is widely accepted and used inside the scientific community. In fact, REDUCE provides with REDLOG the currently only professionally maintained and marketed implementation of real quantifier elimination.

We have introduced two extensions of quantifier elimination by virtual substitution, namely extended quantifier elimination and generic quantifier elimination. Generic quantifier elimination has been developed by the author for the purpose of geometric theorem proving. This thesis is going to demonstrate that the introduction of generic quantifier elimination is actually an important progress in geometric theorem proving and in other geometric application areas.

With the description of generic quantifier elimination, we have also discussed the importance of sophisticated variable selection strategies for elimination. This is crucial for avoiding failures due to the degree restrictions to the most possible extent. Although the degree restriction can be overcome in theory, it will for efficiency reasons always be advantageous to keep low all degrees of quantified variables as long as possible. The discussed strategies are thus not a temporary technique to overcome limitations of the current implementation. Instead they are of general theoretical interest. The generic quantifier elimination is closely related to the concept of a theory in the parameters, which can be passed as an optional argument to both regular and extended generic quantifier elimination by virtual substitution.

We have described the design of the REDLOG system, which will be our major computation tool throughout this thesis. A summary of a variety of application areas has given an idea of the scope of the methods used here. They provide an extremely general tool that is by no means restricted to geometric problems, which we focus on here.

Finally, we have clarified the difference between real quantifier elimination and apparently closely related areas such as automatic theorem proving and constraint logic programming.

Chapter 3

Geometric Theorem Proving

In view of the fact that real quantifier elimination procedures amount to decision methods for variable-free formulas, it is a straightforward idea to apply them for automated theorem proving. Theorems of elementary geometry have traditionally been considered an important test case for the scope of methods in automatic theorem proving.

In Section 3.1, we summarize the conventional algebraic methods for automatic geometry proving discussed in the literature so far. In Section 3.2, we introduce the important notion of geometric non-degeneracy conditions, which have to be detected by automatic provers. With our approach, the assumptions made by generic quantifier elimination will yield these conditions. Section 3.3 illustrates the common method of algebraically modeling geometric configurations and conjectures, the basic ideas of which are common to all the proving methods we discuss including our own one. Section 3.4 points on the limitations present with the conventional algebraic methods. These limitations are on one hand due to their being *complex* in contrast to *real*. On the other hand, they are restricted to algebraic translations that are universal Horn clauses. In Sections 3.5, we suggest to overcome all these limitations by using (generic) real quantifier elimination as a geometric theorem prover. Section 3.6 exhibits the important advantage of our generic quantifier elimination to be not only a theorem prover but even a *theorem finder* in case that the input hypothesis is not a theorem even under certain non-degeneracy assumptions. Section 3.7 illustrates our method in detail by two examples computed by hand. Section 3.8 finally provides a collection of automatic proofs accomplished by our implementation of the generic virtual substitution method within the REDLOG system, which has been described in Section 2.6 of the previous chapter.

3.1 Conventional Algebraic Methods

Geometric problems have stimulated a variety of algebraic techniques for their solution. Around 1977, Wu discovered the *Wu–Ritt method* adapting a method by Ritt [Rit32, Rit50] for differential algebra [Wu84, Wu86a]. Chou made important contributions to the development and application of the Wu–Ritt method.

Most of Chou's work has been summarized in an impressing monograph including more than 500 geometric theorems automatically proved by this method [Cho88].

Kapur [Kap86] developed a prover based on a radical membership test via Gröbner Basis [Buc65] computation.

Kutzler and Stifter [KS86] developed another prover also based on Gröbner techniques. They modified the notion of reduction to a *pseudo reduction*, which is slightly similar to the notion of reduction used in the Wu–Ritt method.

Wang [Wan93b, Wan95a] developed complex elimination methods based on ideas by Seidenberg [Sei56a, Sei56b, Sei69] but also inspired by Wu–Ritt.

All these methods have proved to be quite successful. Their common basis can be characterized as follows: The first step is a translation of the geometric assertion $\mathfrak{G}$ via a suitably positioned coordinate system into an algebraic statement φ involving multivariate polynomial equations.

Subsequently, the corresponding algebraic method itself is applied in an attempt to prove φ as a statement about *complex numbers*. Since φ is generally a universally quantified assertion, the validity of φ over the complex numbers entails the validity of φ over the reals and thus an automatic proof of the original geometric assertion. If, in contrast, φ turns out to be false over the complex numbers, no decision on the validity of $\mathfrak{G}$ can be made.

3.2 Subsidiary Conditions

In general, geometric theorems are only *generically* true, or in other words, true only under some *non-degeneracy conditions*. Such conditions mostly state that the involved geometric configurations are proper. That is, triangles are not degenerated to lines or points, or several points do not coincide, etc. Beyond this, non-degeneracy conditions might exclude certain well-behaved configurations like parallel lines or rectangular triangles. Since such cases are not really degenerate, non-degeneracy conditions have also been called *subsidiary conditions*.

It is in general by no means easy to find and add all the non-degeneracy conditions for the validity of a theorem. Illustrating examples for hidden conditions have been given by Chou (cf. [Cho88], pp. 43–44).

The answers produced by the conventional proving methods also happen to be correct only under certain assumptions in the shape of negated equations, which are, of course, explicitly provided to the user. It is a most amazing fact that these side conditions produced by the proof methods mostly coincide with the subsidiary conditions necessary for the validity of the target theorem. That is, an apparent weakness of the algebraic procedures turns out as an adequate tool for actively *discovering* additional "trivial" hypothesises.

The presence of this phenomenon with various different automatic proof strategies suggests that the approach to translate geometry to algebra is extremely natural.

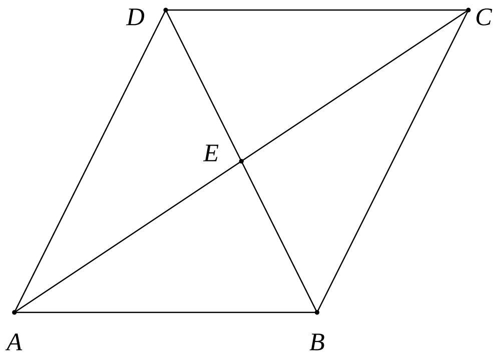

Figure 3.1: Parallelogram diagonal cut (Example 3.1 and Example 3.8)

3.3 Translating Geometry to Algebra

The style of algebraic modeling with parameters and dependent variables commonly used and introduced in this section can be found already in Hilbert's *Grundlagen der Geometrie*[1] [Hil87], originally published in 1899; see in particular the proof of Theorem 62 there.

We illustrate the approach by means of an example modeling the following simple theorem taken from [Cho88], which we will also prove automatically as Example 3.8 on page 55.

Example 3.1 (Chou) Let $ABCD$ be a parallelogram, and let E be the intersection point of its diagonals. Then E is the midpoint of both diagonals (Figure 3.1). □

In the translation, we distinguish between two types of variables: On one hand we have *parameters* describing coordinates that can be chosen more or less arbitrarily. On the other hand there are *dependent variables* describing coordinates that are uniquely determined by the parameters.

We denote parameters by $u_1, \dots, u_m$, and we denote dependent variables by $x_1, \dots, x_n$. In the example, we introduce wlog. the following parameters:

$$A = (0,0), \quad B = (u_1, 0), \quad C = (u_2, u_3).$$

For any choice of u_1, u_2, u_3, the coordinates of D and E are now uniquely determined. We use dependent variables for their coordinates:

$$D = (x_1, x_2), \quad E = (x_3, x_4).$$

Using these coordinates, we can now write down the hypothesis: For $ABCD$ to be a parallelogram, we require that $x_1 = u_2 - u_1$ and $x_2 = u_3$. The point E is the intersection of the diagonals AC and BD if and only if it lies on both these diagonals. This is in turn the case if and only if the lines AE and AC are identical, and the lines BE and BD are identical. The identity of lines with a common point can be expressed in terms of their gradients:

$$\frac{x_4}{x_3} = \frac{u_3}{u_2}, \quad \frac{x_4}{x_3 - u_1} = \frac{x_2}{x_1 - u_1}.$$

[1] Foundations of Geometry

Clearing denominators this amounts to the polynomial equations $x_4 u_2 = u_3 x_3$ and $x_4(x_1 - u_1) = x_2(x_3 - u_1)$, respectively.

The conclusion of the theorem is that E is the midpoint of both AC and BD. That is,

$$\sqrt{x_3^2 + x_4^2} = \frac{\sqrt{u_2^2 + u_3^2}}{2} \quad \text{and} \quad \sqrt{(x_3 - u_1)^2 + x_4^2} = \frac{\sqrt{(x_1 - u_1)^2 + x_2^2}}{2},$$

respectively. This yields restated with polynomials $4x_3^2 + 4x_4^2 = u_2^2 + u_3^2$ and $4(x_3 - u_1)^2 + 4x_4^2 = (x_1 - u_1)^2 + x_2^2$, respectively.

In all the polynomial equations obtained so far, we subtract the right hand sides to the corresponding left hand sides obtaining the following left hand side polynomials for the hypothesises and the conclusions:

$$\begin{aligned}
h_1(\mathbf{u}, \mathbf{x}) &= x_1 + u_1 - u_2 \\
h_2(\mathbf{u}, \mathbf{x}) &= x_2 - u_3 \\
h_3(\mathbf{u}, \mathbf{x}) &= u_2 x_4 - u_3 x_3 \\
h_4(\mathbf{u}, \mathbf{x}) &= x_4(x_1 - u_1) - x_2(x_3 - u_1) \\
c_1(\mathbf{u}, \mathbf{x}) &= 4x_3^2 + 4x_4^2 - u_2^2 - u_3^2 \\
c_2(\mathbf{u}, \mathbf{x}) &= 4(x_3 - u_1)^2 + 4x_4^2 - (x_1 - u_1)^2 - x_2^2.
\end{aligned}$$

Automatic provers will now try to prove the *generic validity* of the following assertion over either $\mathbb{R}$ or $\mathbb{C}$:

$$\forall \mathbf{u} \forall \mathbf{x} \Big(\big(\bigwedge_{i=1}^{4} h_i(\mathbf{u}, \mathbf{x}) = 0 \big) \longrightarrow \big(c_1(\mathbf{u}, \mathbf{x}) = 0 \wedge c_2(\mathbf{u}, \mathbf{x}) = 0 \big) \Big).$$

Generic validity means validity under certain subsidiary conditions to be discovered by the automatic prover as discussed in Section 3.2. It should be clear from our modeling that well-expressed non-degeneracy conditions will involve the parameters $u_1, \ldots, u_3$ but not the dependent variables $x_1, \ldots, x_4$.

Among the conventional algebraic methods, only the approach of Kapur does not distinguish between parameters and dependent variables.

Note that one enters a certain amount of geometric knowledge at the modeling stage: For instance, our formulation of the Euclidean line lengths require knowledge of Pythagoras' theorem. Trying to prove Pythagoras' theorem with a such a modeling would certainly result in a trivial algebraic formulation. Generally, one has to be very careful about performing relevant parts of the proof already at the translation stage. Kutzler has addressed this problem in his doctoral dissertation [Kut88]. Following Hilbert, he introduces the basic types *point* and *line*, and then defines a restricted set of predicates such as *parallel* or *midpoint* on these types. First-order formulas over the obtained two-sorted relational language are then mechanically translated to corresponding formulas over the language of rings.

3.4 Limitations of the Algebraic Methods

It is an amazing fact, which does not appear to have a sufficient theoretical explanation up to now, that for the overwhelming majority of theorems in the

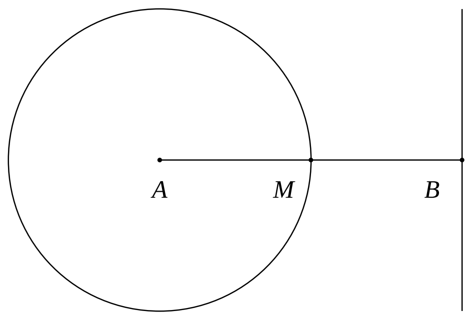

Figure 3.2: The line L would intersect the circle in the complex plane (Example 3.2 and Example 3.10)

plane geometry of points, lines, circles, and cones the algebraic translation φ does hold in the field of complex numbers. Anyway, there are certain limitations on the applicability of algebraic methods to geometry proving.

First of all, as already indicated in the previous section, it is impossible to *disprove* theorems. Note that this can in particular not be done by proving the negation of the target theorem since this negation is not a universal but an existential sentence.

We now turn to illustrating the various problems that can arise when trying to reduce the proof of some theorem over the reals to a proof over the complex numbers. For this we give four examples for the types of problems that can arise.

The procedures fail in quite a trivial way if the theorem asserts properties of points that do not exist in the real plane but exist in the complex plane. This is the case for the point C in the following example taken from [Cho88], pp. 56–57:

Example 3.2 (Chou) Let A and B be two distinct points, let M be the midpoint of A and B. Let C be one of the intersections of the circle $\big(A, (AM)\big)$ with the line L passing through B and perpendicular to AB, then $AC \equiv AB$ (Figure 3.2). $\square$

For an automatic proof of this conjecture by real quantifier elimination see Section 3.8.1, page 57, Example 3.10.

Among the few non-trivial examples for the failure of algebraic methods in real geometry, the following is a famous one. It is originally due to MacLane, 1936, [Mac36]:

Example 3.3 (MacLane) Consider eight points $A, \dots, H$ such that the following eight triples are collinear (A, B, D), (B, C, E), (C, D, F), (D, E, G), (E, F, H), (F, G, A), (G, H, B), (H, A, C). Then all eight points lie on a line. $\square$

MacLane's example has been discussed by Kutzler, who also provides additional examples for real theorems that fail over the complex numbers [Kut88]. Conti and Traverso have also discussed the example in detail [CT95]. We will automatically prove the conjecture in Section 3.8.2, page 69, Example 3.20.

There are numerous reasonable geometric theorems that involve ordering inequalities in their formulation. One prominent example is Pedoe's inequality:

Example 3.4 (Pedoe) Given two arbitrary triangles ABC and $A'B'C'$ with sides a, b, c and a', b', c', respectively, the areas Δ and Δ' of these triangles satisfy the following inequality:

$$a'^2(b^2 + c^2 - a^2) + b'^2(c^2 + a^2 - b^2) + c'^2(a^2 + b^2 - c^2) \geqslant 16\Delta'\Delta. \quad \square$$

Such a theorem can, of course, not straightforwardly be formulated in a purely algebraic setting. It should be mentioned, however, that both the Wu–Ritt method and Wang's method have "automatically" proved Pedoe's inequality by considering local minima [Wan95a]. We return to Pedoe's inequality in Section 3.8.2, page 67, Example 3.18.

The following theorem cannot be tackled by such tricks, but requires a genuinely real method:

Example 3.5 (Median bisector theorem) Consider a non-isosceles triangle ABC. The median over the side AB is always greater than the interior bisector on the same side. See Figure 3.12 on page 66. $\square$

Such problems, which inherently require ordering inequalities, are subject to current research. The median bisector theorem has been addressed by Wu using an incomplete extension of Wu–Ritt reduction that can cope with ordering inequalities [Wu92]. We will automatically prove the median bisector theorem in Section 3.8.2, page 66, Example 3.17.

3.5 Proving by Real Quantifier Elimination

Our idea is now to use real quantifier elimination methods for automatic geometry proving. This overcomes all the limitations discussed in Section 3.4.

Compared to the conventional algebraic methods, real quantifier elimination is generally believed to be not able to compete: In 1988, Kutzler compared several provers. The SAC-2 implementation of Collins' CAD then failed for the largest part of his benchmarks [Kut88]. The author of this thesis is not aware of any publication on geometry proving by pure quantifier elimination.

From these well-known efficiency problems, and from the discussion on subsidiary conditions in Section 3.2 it follows that for most cases it is not appropriate to simply use real quantifier elimination as a decision procedure for the universal geometric sentences to be proved. Due to missing non-degeneracy hypothesises it would in general find that the input sentence is not a theorem. Instead, we could universally quantify only the dependent variables $x_1, \dots, x_n$, but leave the parameters $u_1, \dots, u_m$ unquantified. The elimination result with regular quantifier elimination would then be a quantifier-free formula in the parameters providing *necessary and sufficient* conditions for the theorem to hold.

In order to obtain instead subsidiary conditions of the usual kind, we use the much more efficient generic quantifier elimination introduced in Section 2.4.2, which has actually been developed for geometric theorem proving [DSW98a]. Recall that this procedure returns on input of a quantified formula

$$\varphi(u_1, \dots, u_m) \equiv \mathsf{Q}_1 x_1 \dots \mathsf{Q}_n x_n \psi(u_1, \dots, u_m, x_1, \dots, x_n), \quad \mathsf{Q}_i \in \{\exists, \forall\}$$

a quantifier-free formula $\varphi'(u_1, \dots, u_m)$ together with a conjunction of assumptions $\vartheta'(u_1, \dots, u_m)$ such that

$$\forall u_1 \dots \forall u_m \big(\vartheta'(u_1, \dots, u_m) \longrightarrow (\varphi(u_1, \dots, u_m) \longleftrightarrow \varphi'(u_1, \dots, u_m))\big).$$

We have already indicated in Section 2.4.2 that as with the conventional algebraic methods, the assumptions ϑ' correspond mostly to geometric subsidiary conditions.

If φ' equals "true," then we have proved φ and thus the geometric assertion $\mathfrak{G}$ modeled by φ under the non-degeneracy conditions contained in ϑ'. Otherwise, it is still possible that φ' holds wrt. ϑ'; in other words

$$\forall u_1 \dots \forall u_m \big(\vartheta'(u_1, \dots, u_m) \longrightarrow \varphi'(u_1, \dots, u_m)\big).$$

This can be checked automatically using some unmodified quantifier elimination procedure such as the virtual substitution method, if the degrees allow to do so, or partial CAD. If we succeed in eliminating all the quantifiers, we definitely know whether the input conjecture φ is a theorem wrt. ϑ' or not.

In the latter case there are two possibilities. Either our method has not found all necessary non-degeneracy conditions, i.e., ϑ' is not sufficient for φ to be an appropriate algebraic translation of the geometric conjecture $\mathfrak{G}$, or $\mathfrak{G}$ itself does not hold in the claimed generality.

If we suspect that there are simply non-degeneracy conditions missing, we can apply a *theory generator*. This is a procedure that enlarges ϑ' by further negated equations until it, hopefully, implies φ'. The technique is based on adding negated equations that occur literally in φ'. This procedure can certainly fail, which happens in particular when there are only few negated equations in φ' or none at all.

Note, however, that in either case φ' specifies additional constraints on the parameters u_1, …, u_m that are necessary and sufficient for the validity of $\mathfrak{G}$ under ϑ'. Both φ' and ϑ' can thus be used in order to turn a not generally valid geometric conjecture $\mathfrak{G}$ into a true geometric theorem, provided that one succeeds in a geometric back-translation of the algebraic assertions ϑ' and φ'. This leads us from theorem proving to theorem finding.

3.6 From Theorem Proving to Theorem Finding

The non-degeneracy conditions obtained by the conventional algebraic methods are sufficient but, in general, not necessary for the conjecture under consideration to hold. The same holds for the non-degeneracy conditions obtained by generic quantifier elimination. The conditions on the parameters obtained by regular real quantifier elimination are, in contrast, both necessary and sufficient. In other words, regular quantifier elimination results are of better quality, since they do under no circumstances suggest superfluous assumptions on the parameters.

Furthermore, quantifier elimination results are arbitrary boolean combinations of equations, negated equations, and ordering inequalities, while conventional subsidiary conditions are restricted to a collection or conjunction of negated equations. Such conjunctions can only except a set of measure zero. So on one hand, quantifier elimination can find much more precise conditions, e.g. that a theorem holds for isosceles triangles only; on the other hand, elimination results are certainly harder to translate back to geometry. Moreover, regular quantifier elimination causes efficiency problems as discussed in the previous section.

Most interestingly, generic quantifier elimination combines both a finder for sufficient simple non-degeneracy conditions, and a finder for more sophisticated hypothesises that the user missed to add: If the proof of the input conjecture φ fails in spite of all appropriate non-degeneracy conditions, then the obtained result yields necessary and sufficient conditions for the conjecture to hold. These can be added as additional premises. In other words, we do not only prove theorems but even *find* theorems. For examples see Example 3.14, 3.15, or 3.19. Note the most amazing fact that the "usual" non-degeneracy conditions, as a rule, actually happen to get into the theory but not into the quantifier elimination result, although this is well possible in theory.

3.7 Generic Virtual Substitution Sample Proofs

We are going to illustrate by two simple examples how the generic variant of quantifier elimination by virtual substitution works. Our first example shows how non-degeneracy conditions are introduced, and how we reduce universal quantifiers to existential quantifiers. It is, however, rather trivial concerning the elimination set construction: the elimination of both quantifiers can be reduced to our extended notion of Gauss elimination (see Section 2.3.3).

Example 3.6 Two lines intersect in one and only one point. □

The first line is modeled by the X-axis, the second one by an affine linear function $mx + b$:

$$\exists x\bigl(mx + b = 0 \land \forall y(y \neq x \longrightarrow my + b \neq 0)\bigr).$$

We start with turning the formula into a prenex *positive* formula, i.e. the matrix of the formula becomes an $\land$-$\lor$-combination of atomic formulas.

$$\exists x\forall y\bigl(mx + b = 0 \land (y = x \lor my + b \neq 0)\bigr).$$

The elimination of $\forall y$ is reduced to that of

$$\neg\exists y\bigl(mx + b \neq 0 \lor (y \neq x \land my = -b)\bigr).$$

Since $mx + b \neq 0$ is independent of y, we may restrict our attention to the nested conjunction. We add $m \neq 0$ to our theory. Under this assumption the equation $my = -b$ is non-trivial, and it suffices to substitute $\frac{-b}{m}$ as a test term. The result is the following:

$$\neg\bigl(0 \neq 0 \lor (-b \neq mx \land -b = -b)\bigr).$$

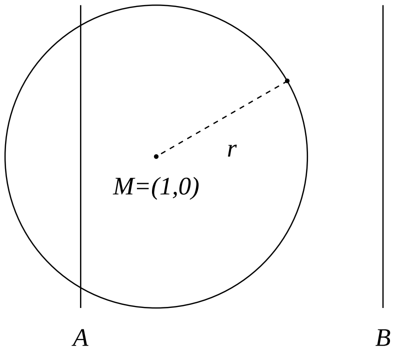

Figure 3.3: Intersection between a strip and a disk (Example 3.7)

After encoding the negation "$\neg$" into the atomic formulas, simplifying, and renormalizing wrt. the next quantifier "$\exists x$" we have:

$$mx = -b.$$

We have already made the assumption $m \neq 0$, hence the remaining equation is nontrivial. We substitute $\frac{-b}{m}$ for x obtaining "true" as the final elimination result. The non-degeneracy condition $m \neq 0$ states that our two lines be not parallel. The non-generic virtual substitution method would, instead of making the assumption, have introduced case distinctions on the vanishing of m within the resulting formula.

Our second example illustrates how the virtual substitution method deals with inequalities and with quadratic variables.

Example 3.7 Consider in 2-space a disk with center M and radius r, and two parallel lines A, B. Find necessary and sufficient conditions for the strip between A and B to have a nonempty intersection with the disk (see Figure 3.3). □

In our algebraic translation, we denote by (x, y) a point which is both inside the strip and inside the disk. We locate the center M of the disk at $(-1, 0)$, and the lines A and B are defined as $\{(a, y) \mid y \in \mathbb{R}\}$ and $\{(b, y) \mid y \in \mathbb{R}\}$, respectively. This yields the following input formula:

$$\exists x \exists y \big((x+1)^2 + y^2 \leqslant r^2 \wedge a \leqslant x \wedge x \leqslant b\big).$$

We start with the elimination of $\exists y$. The purely quadratic occurrence of y allows a shift $h = y^2$:

$$\exists h \big(h \leqslant r^2 - (x+1)^2 \wedge h \geqslant 0 \wedge a \leqslant x \wedge x \leqslant b\big).$$

There is both a lower bound 0 and an upper bound $r^2 - (x+1)^2$ for h. In this situation, we know that it suffices to substitute either all upper bounds together with the formal symbol $-\infty$ or all lower bounds together with ∞. We decide for the elimination set $\{\infty, r^2 - (x+1)^2\}$ yielding

$$\begin{aligned}&(\mathrm{false} \wedge \mathrm{true} \wedge a \leqslant x \wedge x \leqslant b) \vee \\ &\quad \big(r^2 - (x+1)^2 \leqslant r^2 - (x+1)^2 \wedge r^2 - (x+1)^2 \geqslant 0 \wedge a \leqslant x \wedge x \leqslant b\big).\end{aligned}$$

Note that we know the direction of the bounds because h does not occur with a parametric coefficient in the corresponding atomic formulas. In connection with strict inequalities there would also occur terms including infinitesimal formal symbols $\pm\varepsilon$. The substitution of terms with formal symbols into atomic formulas is described in [Wei97b].

After simplification and renormalization wrt. the next quantifier $\exists x$, we have

$$(x+1)^2 \leqslant r^2 \wedge x \geqslant a \wedge x \leqslant b$$

including linear bounds a, b, and quadratic bounds $r-1$ and $-r-1$ for x. We do not have to substitute the quadratic bounds because we are in an important special case: There is only one quadratic inequality. This can be treated by substituting the zero -1 of the derivative

$$\frac{\partial\big((x+1)^2 - r^2\big)}{\partial x} = 2x - 2.$$

This will not increase the degree in the parameters, which are possibly quantified from outside. Proceeding this way, however, we cannot decide between linear lower or upper bounds. Disjunctive substitution of the elimination set $\{a, b, -1, -\infty, \infty\}$ yields

$$\begin{aligned}&\big((a+1)^2 \leqslant r^2 \wedge a \geqslant a \wedge a \leqslant b\big) \vee \big((b+1)^2 \leqslant r^2 \wedge b \geqslant a \wedge b \leqslant b\big) \vee \\ &\quad (0 \leqslant r^2 \wedge -1 \geqslant a \wedge -1 \leqslant b) \vee (\text{false} \wedge \text{false} \wedge \text{true}) \vee \\ &\quad (\text{false} \wedge \text{true} \wedge \text{false}).\end{aligned}$$

In general, one has to substitute all the roots of the quadratic constraints possibly with $\pm\varepsilon$. Note that these roots can contain surds. In [Wei97b] there is described how to substitute the roots in such a way that the substitution result does not contain any surds.

Our sample elimination result is automatically simplified to the following quantifier-free formula:

$$\big((a+1)^2 \leqslant r^2 \wedge a \leqslant b\big) \vee \big((b+1)^2 \leqslant r^2 \wedge b \geqslant a\big) \vee (a \leqslant -1 \leqslant b).$$

There have not been made any assumptions with this elimination.

3.8 A Collection of Automatic Proofs

As mentioned in the previous section REDLOG provides an interface to Hoon Hong's QEPCAD: it can spawn a QEPCAD process, communicate formulas to it, receive the results, and convert them back to its own internal formula format. We have actually tried to compute all examples discussed in this section with QEPCAD, which failed in most cases probably due to the large number of variables. On the other hand, QEPCAD does a good job in checking whether $\vartheta' \longrightarrow \varphi'$ after application of our method since both ϑ' and φ' contain only few variables, namely at most the parameters u_i of the input problem φ. Such applications of QEPCAD can be found in Example 3.17 on page 66 and Example 3.22 on page 72.

We start with summarizing the timings and solutions of several examples taken from Chou [Cho88]. Most of the latter can also be proved by complex methods. Then we discuss the automatic proofs of some examples of real geometry taken from Wu [Wu92], Wang [Wan95b], Kutzler [Kut88], and McPhee, Chou, and Gao [MCG94].

Throughout this section, *generic quantifier elimination* refers to the REDLOG implementation of generic quantifier elimination by virtual substitution as described in Section 3.7. By *regular quantifier elimination*, we denote the REDLOG implementation of quantifier elimination by virtual substitution as described in Section 2.3.3.

All computations have been performed on a SUN Ultra 1 Model 140 workstation using 32 MB of memory.

3.8.1 Examples Discussed by Chou

The following examples are taken from the famous monograph of Chou [Cho88] on geometry proving using the Wu–Ritt method. The theorems, with the exception of Example 3.10 and our modification of Example 3.13, hold also over the complex numbers. Except where explicitly mentioned, we use the algebraic translations by Chou.

The theorems are taken from an example section illustrating the various features of the Wu–Ritt method and of Chou's prover. This choice provides us a representative sample set. Since Chou's attention is mainly on the methodology, he does, unfortunately, not give his non-degeneracy conditions for all the examples. For the cases, where the conditions are given, we compare them with the conditions obtained by our generic quantifier elimination.

Example 3.8 (Parallelogram diagonals) Let $ABCD$ be a parallelogram, and let E be the intersection point of its diagonals. Then E is the midpoint of both of the diagonals. See Figure 3.1 on page 47. □

Chou uses the following algebraic translation of this conjecture: The points take coordinates

$$A = (0,0), \quad B = (u_1, 0), \quad C = (u_2, u_3), \quad D = (x_2, x_1), \quad E = (x_4, x_3),$$

and the hypothesises $h_1, \ldots, h_4$ and the conclusion g read as follows:

$$\begin{array}{rcll} h_1 & \equiv & u_1x_1 - u_1u_3 = 0 & AB \parallel DC \\ h_2 & \equiv & u_3x_2 - (u_2 - u_1)x_1 = 0 & DA \parallel CB \\ h_3 & \equiv & x_1x_4 - (x_2 - u_1)x_3 - u_1x_1 = 0 & E \in BD \\ h_4 & \equiv & u_3x_4 - u_2x_3 = 0 & E \in AC \\ g & \equiv & 2u_2x_4 + 2u_3x_3 - u_3^2 - u_2^2 = 0 & AE \equiv CE. \end{array}$$

On input of $\varphi \equiv \forall x_1 \forall x_2 \forall x_3 \forall x_4 (h_1 \land h_2 \land h_3 \land h_4 \longrightarrow g)$, we obtain by generic quantifier elimination after 40 ms the elimination result $\varphi' \equiv \mathrm{true}$ plus the non-degeneracy conditions $\vartheta' \equiv u_1 \neq 0 \land u_3 \neq 0$, which state that $ABCD$ is a proper parallelogram.

Chou obtains in addition a superfluous condition $u_3x_2 - u_2x_1 - u_1u_3 \neq 0$. This states that the diagonals have different gradients, which trivially holds in any parallelogram; more precisely, one always has $\nabla AC = -\nabla BD$.

Precise information on the non-degeneracy conditions that are actually necessary is obtained for the price of a significantly higher computation time by

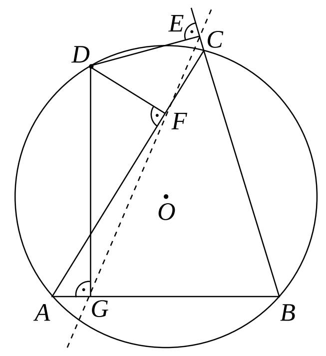

Figure 3.4: Simson's theorem (Example 3.9)

applying regular quantifier elimination by virtual substitution to φ. REDLOG obtains after 200 ms for quantifier elimination and subsequent Gröbner simplification the following equivalent quantifier-free formula:

$$\varphi' \equiv (u_2 = 0 \vee u_3 \neq 0) \wedge (u_1 \neq 0 \vee u_3 = 0).$$

Using the algebraic translation derived for the conjecture in Section 3.3, one yields essentially the same though slightly more complicated results.

Example 3.9 (Simson's theorem) Consider a triangle ABC, and select a point D on its circumcircle. From D draw perpendiculars to each of the sides BC, CA, and AB of the triangle. Then the feet E, F, and G of these perpendiculars are collinear. See Figure 3.4. □

In Chou's algebraic translation, the corner points of the triangle and the point D on its circumcircle take the following coordinates:

$$A = (0, 0), \quad B = (u_1, 0), \quad C = (u_2, u_3), \quad D = (x_3, u_4).$$

Note that due to that fact that D may be freely chosen on the circumcircle, one of its coordinates is modeled by a parameter u_4. Since the choice of D is restricted to lying on the circle, its other coordinate is uniquely determined and thus modeled by a dependent variable x_3. Either are the center O of the circumcircle and the footpoints of the perpendiculars:

$$O = (x_2, x_1), \quad E = (x_5, x_4), \quad F = (x_7, x_6), \quad G = (x_3, 0).$$

Using these coordinates, Chou derives hypothesises $h_1, \ldots, h_7$ and the conclusion g as follows:

$$\begin{array}{rcll}
h_1 & \equiv & 2u_2x_2 + 2u_3x_1 - u_3^2 - u_2^2 = 0 & OA \equiv OC \\
h_2 & \equiv & 2u_1x_2 - u_1^2 = 0 & OA \equiv OB \\
h_3 & \equiv & -x_3^2 + 2x_2x_3 + 2u_4x_1 - u_4^2 = 0 & OA \equiv OD \\
h_4 & \equiv & u_3x_5 + (-u_2 + u_1)x_4 - u_1u_3 = 0 & \text{collinear}(E, B, C) \\
h_5 & \equiv & (u_2 - u_1)x_5 + u_3x_4 + (-u_2 + u_1)x_3 - u_3u_4 = 0 & DE \perp BC \\
h_6 & \equiv & u_3x_7 - u_2x_6 = 0 & \text{collinear}(F, A, C) \\
h_7 & \equiv & u_2x_7 + u_3x_6 - u_2x_3 - u_3u_4 = 0 & DF \perp AC \\
g & \equiv & x_4x_7 + (-x_5 + x_3)x_6 - x_3x_4 = 0 & \text{collinear}(E, F, G).
\end{array}$$

For the input $\varphi \equiv \forall x_1 \dots \forall x_7(h_1 \wedge \dots \wedge h_7 \longrightarrow g)$, the generic quantifier elimination of REDLOG obtains after 180 ms elimination result $\varphi' \equiv$ true valid under the subsidiary conditions

$$\vartheta' \equiv u_1^2 - 2u_1u_2 + u_2^2 + u_3^2 \neq 0 \wedge u_1 \neq 0 \wedge u_2 \neq 0 \wedge u_3 \neq 0.$$

The condition $u_1 \neq 0 \wedge u_3 \neq 0$ states that ABC is a proper triangle. The first condition $(u_1 - u_2)^2 + u_3^2 \neq 0$ is implied by $u_3 \neq 0$ and can thus be dropped. Finally, the condition $u_2 \neq 0$ states that $\angle BAC$ is not a right angle.

Regular quantifier elimination by virtual substitution fails on φ due to lack of memory.

Chou obtains up to numerical factors our subsidiary conditions ϑ', and additionally $4u_1u_3 \neq 0$ and $-u_3^2 - u_2^2 \neq 0$. The first one is obviously redundant. The second one states that $\angle BAC \neq 45°$.

Example 3.10 (Bounded circle) Let A and B be two distinct points, let M be the midpoint of A and B. Let C be one of the intersections of the circle $(A, |AM|)$ with the line L passing through B and perpendicular to AB, then $AC \equiv AB$. See Figure 3.2 on page 49. □

This conjecture does not hold in the complex plane, since there the circle $x_1^2 + x_2^2 = r^2$ with center 0 and radius r is unbounded.

Chou does not give an algebraic translation but only mentions the unapplicability of his method. We choose the following coordinates for the points:

$$A = (0, 0), \quad B = (u_1, 0), \quad M = (x_1, 0), \quad C = (x_2, x_3).$$

The hypothesises h_1, h_2, h_3 and the conclusion g then are as follows:

$$\begin{array}{rcll} h_1 & \equiv & 2x_1 = u_1 & AM \equiv AB/2 \\ h_2 & \equiv & x_2^2 + x_3^2 = x_1^2 & C \in (A, |AM|) \\ h_3 & \equiv & (x_2 - u_1)x_1 = 0 & BC \perp AM \\ g & \equiv & x_3 = u_1 & AC \equiv AB. \end{array}$$

We apply REDLOG to the input $\varphi \equiv \forall x_1 \forall x_2 \forall x_3(h_1 \wedge h_2 \wedge h_3 \longrightarrow g)$. Generic quantifier elimination yields after 10 ms the quantifier-free formula $\varphi' \equiv$ true plus the subsidiary condition $\vartheta' \equiv u_1 \neq 0$ stating that A and B are actually distinct points. Regular quantifier elimination yields after 20 ms $\varphi' \equiv$ true. So ϑ' is superfluous but still reasonable.

This quantifier elimination can also be performed using QEPCAD. After less than 10 ms one obtains $\varphi' \equiv$ true.

Example 3.11 (Distances in square figure) Let $ABCD$ be a square, let CG be a parallel to the diagonal BD, and let E be a point on CG with $BE \equiv BD$. Denote the intersection of BE and DC by F. Then $DF \equiv DE$. See Figure 3.5. □

The coordinates for the corner points of the square are given as $A = (0, 0)$, $B = (u_1, 0)$, $C = (u_1, u_1)$, and $D = (0, u_1)$. The constructed points E and F take coordinates $E = (x_1, x_2)$ and $F = (x_3, u_1)$. The hypothesises h_1, h_2, h_3

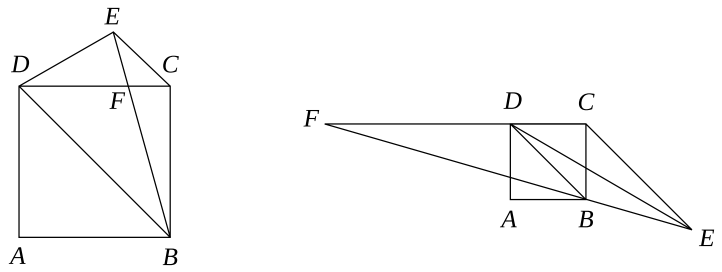

Figure 3.5: Two types of configurations for Example 3.11

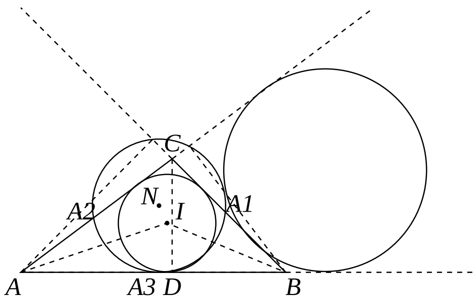

Figure 3.6: Feuerbach's theorem (Example 3.12)

and the conclusion g read as follows:

$$\begin{array}{rcll} h_1 & \equiv & x_2^2 + x_1^2 - 2u_1x_1 - u_1^2 = 0 & BE \equiv BD \\ h_2 & \equiv & -u_1x_2 - u_1x_1 + 2u_1^2 = 0 & CE \parallel BD \\ h_3 & \equiv & -x_2x_3 + u_1x_2 + u_1x_1 - u_1^2 = 0 & F \in BE \\ g & \equiv & x_3^2 - x_2^2 + 2u_1x_2 - x_1^2 - u_1^2 = 0 & DF \equiv DE. \end{array}$$

Generic quantifier elimination yields after 60 ms the non-degeneracy condition $\vartheta' \equiv u_1 \neq 0$ stating that $ABCD$ is a proper square, and the elimination result $\varphi' \equiv$ true.

Regular quantifier elimination by virtual substitution cannot be applied since this example violates the degree restrictions of the REDLOG implementation. QEPCAD computes $u_1 \neq 0$ as quantifier-free equivalent, which takes 1 s with 10^6 cells. This shows that the non-degeneracy condition is actually necessary for the conjecture to hold.

Note that the algebraic translation does not distinguish between the two possible types of configurations sketched in Figure 3.5. We have thus proved the conjecture for both of them.

Example 3.12 (Feuerbach's theorem) The nine point circle of a triangle is tangent to the incircle and to each of the excircles of the triangle. See Figure 3.6. □

The incircle of a triangle is the circle that is tangent to the three sides; its center is the intersection of the bisectors. An excircle, or escribed circle, of a triangle is a circle tangent to one side of the triangle and to the extensions of the other sides. The nine point circle of a triangle is the circle that passes through the midpoints of the sides. It also contains the feet of the three altitudes.

Chou gives two alternatives for the algebraic translation of this conjecture. From the first one, we can by generic quantifier elimination eliminate 8 out of 9 quantifiers. Then the procedure fails due to the degree blowup with x_1 being of degree 4.

Generic quantifier elimination succeeds, however, with the second translation variant, in which the triangle ABI is freely chosen:

$$D = (0,0), \quad A = (u_1, 0), \quad I = (0, u_2), \quad B = (u_3, 0).$$

The remaining points are constructed from this triangle:

$$C = (x_1, x_2), \quad A_1 = (x_3, x_4), \quad A_2 = (x_5, x_6), \quad A_3 = (x_7, 0), \quad N = (x_8, x_9).$$

With these coordinates, the hypothesises are as follows:

$$\begin{array}{lcll}
h_1 & \equiv & (u_2^2 - u_1^2)u_3 - u_1u_2^2 + u_1^3)x_2 & \\
 & & + (-2u_1u_2u_3 + 2u_1^2u_2)x_1 & \\
 & & + 2u_1^2u_2u_3 - 2u_1^3u_2 = 0 & \tan(\angle CBI) = \tan(\angle IBA) \\
h_2 & \equiv & (u_3^3 - u_1u_3^2 - u_2^2u_3 + u_1u_2^2)x_2 & \\
 & & + (2u_2u_3^2 - 2u_1u_2u_3)x_1 & \\
 & & - 2u_2u_3^3 + 2u_1u_2u_3^2 = 0 & \tan(\angle CAI) = \tan(\angle IAB) \\
h_3 & \equiv & 2x_3 - x_1 - u_3 = 0 & \\
h_4 & \equiv & 2x_4 - x_2 = 0 & A_1 = \mathrm{midpoint}(B, C) \\
h_5 & \equiv & 2x_5 - x_1 - u_1 = 0 & \\
h_6 & \equiv & 2x_6 - x_2 = 0 & A_2 = \mathrm{midpoint}(A, C) \\
h_7 & \equiv & 2x_7 - u_3 - u_1 = 0 & A_3 = \mathrm{midpoint}(A, B) \\
h_8 & \equiv & 2x_6x_9 + (-2x_7 + 2x_5)x_8 & \\
 & & +x_7^2 - x_6^2 - x_5^2 = 0 & NA_3 \equiv NA_2 \\
h_9 & \equiv & 2x_4x_9 + (-2x_7 + 2x_3)x_8 & \\
 & & +x_7^2 - x_4^2 - x_3^2 = 0 & NA_3 \equiv NA_1.
\end{array}$$

For this choice of coordinates, Chou leaves out the conclusion. We use the following conclusion g expressing the fact that the distance between the center of the 9-point circle and the center of the incircle or an excircle equals the difference or the sum of the respective radii:

$$(2x_8^2 + 2x_9^2 + x_7^2 - 2x_7x_8 - 2x_9u_2)^2 = 4(x_8^2 - 2x_7x_8 + x_7^2 + x_9^2)(x_8^2 + x_9^2 - 2x_9u_2 + u_2^2).$$

On input of $\forall x_1 \dots \forall x_9(h_1 \wedge \cdots \wedge h_9 \longrightarrow g)$ we obtain the following non-degeneracy conditions:

$$\begin{aligned}
\vartheta' \quad \equiv \quad & u_1u_3 + u_2^2 \neq 0 \wedge u_1 + u_2 \neq 0 \wedge \\
& u_1 - u_2 \neq 0 \wedge u_1 - u_3 \neq 0 \wedge u_1 \neq 0 \wedge u_2 \neq 0 \wedge u_3 \neq 0.
\end{aligned}$$

The condition $u_1u_3 + u_2^2 \neq 0$ states that $\angle ACB \neq 0$, the conditions $u_1 + u_2 \neq 0$ and $u_1 - u_2 \neq 0$ say that $\angle BAC$ and $\angle CAB$ are not right angles, respectively. Finally $u_1 \neq 0 \wedge u_2 \neq 0 \wedge u_3 \neq 0$ states that ABC is a proper triangle. The elimination result is $\varphi' \equiv \mathrm{true}$. This computation takes 1.4 s.

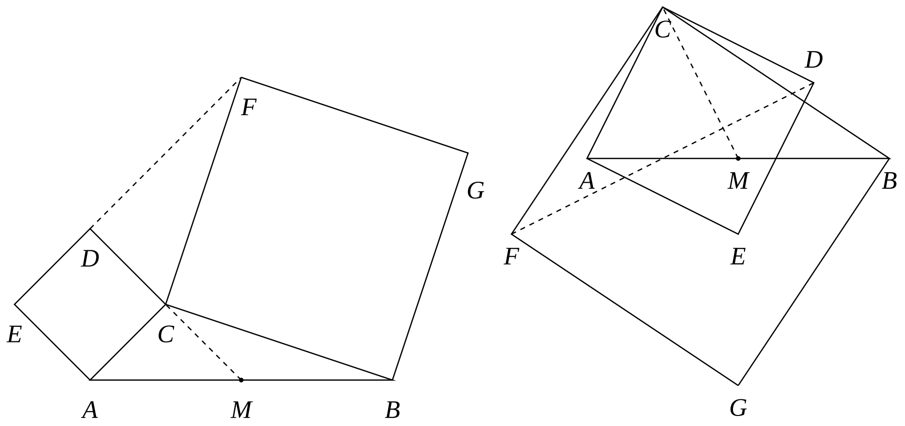

Figure 3.7: The two types of valid configurations for Example 3.13

Example 3.13 (Squares on a triangle) On the two sides AC and BC of a triangle ABC, there are two squares $ACDE$ and $BCFG$ drawn. Denote by M the midpoint of AB. Then $DF \equiv 2CM$. See Figure 3.7. □

After choosing coordinates $A = (u_1, 0)$, $B = (u_2, u_3)$, $C = (0,0)$, $D = (0, u_1)$, $F = (x_1, x_2)$, and $M = (x_3, x_4)$, Chou's hypothesises $h_1, \dots, h_4$ and conclusion g read as follows:

$$\begin{array}{rcll}
h_1 & \equiv & x_2^2 + x_1^2 - u_3^2 - u_2^2 = 0 & CF \equiv BC \\
h_2 & \equiv & u_3 x_2 + u_2 x_1 = 0 & CF \perp BC \\
h_3 & \equiv & 2x_3 - u_2 - u_1 = 0 & \\
h_4 & \equiv & 2x_4 - u_3 = 0 & M = \text{midpoint}(A, B) \\
g & \equiv & 4x_4^2 + 4x_3^2 - x_2^2 + 2u_1 x_2 - x_1^2 - u_1^2 = 0 & DF \equiv 2CM.
\end{array}$$

On input of $\varphi \equiv \forall x_1 \dots \forall x_4 (h_1 \wedge \cdots \wedge h_4 \longrightarrow g)$, generic quantifier elimination yields after 30 ms the surprising results $\vartheta' \equiv u_3 \neq 0$ and

$$\varphi' \equiv u_1 = 0 \vee u_2 = 0.$$

The condition φ' obviously claims a degenerate triangle ABC. Regular quantifier elimination plus final Gröbner simplification shows after 30 ms that φ' is equivalent to φ already without the side condition ϑ'.

Chou has observed that the conjecture of Example 3.13 in fact does not hold in the generality assumed by the algebraic translation φ. For the conjecture to hold, it is crucial that either both squares are outside the triangle, or both squares intersect the triangle. These two types of valid configurations are pictured in Figure 3.7. Figure 3.8 shows, in contrast, an invalid geometric configuration satisfying φ. According to Chou, the valid configurations in the above sense are exactly those satisfying the condition

$$\delta := \frac{-u_1^2 u_3^2}{u_1^2 u_2 x_2 - u_1^2 u_3 x_1} > 0.$$

Quantifier elimination could not derive this condition because it involves the dependent variables x_1 and x_2. While Chou's algebraic prover cannot prove the

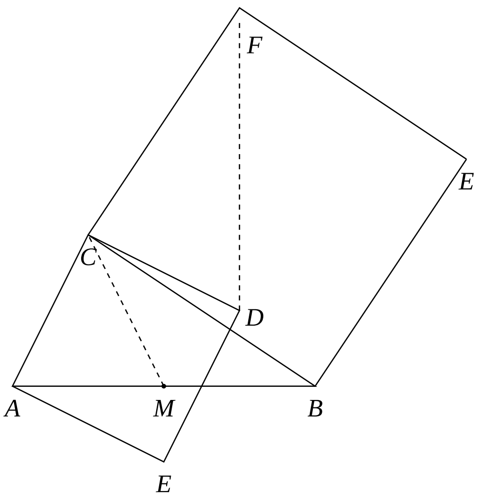

Figure 3.8: An invalid configuration for Example 3.13

conjecture for these configurations fully automatically, our real generic quantifier elimination allows to add $\delta > 0$ to the hypothesises as follows: Since the numerator of δ cannot become positive, we may simply add

$$u_1 \neq 0 \wedge u_3 \neq 0 \wedge u_2 x_2 - u_3 x_1 < 0.$$

We allow ourselves to drop the first two constituents of the conjunction since these are non-degeneracy conditions, which we may expect our prover to detect automatically. On input of

$$\varphi \equiv \forall x_1 \ldots \forall x_4 (h_1 \wedge \cdots \wedge h_4 \wedge u_2 x_2 - u_3 x_1 < 0 \longrightarrow g),$$

generic quantifier elimination yields after 30 ms the result $\varphi' \equiv$ true subject to the non-degeneracy condition $\vartheta' \equiv u_3 \neq 0$. Regular quantifier elimination yields $\varphi' \equiv$ true after 310 ms. This shows that the subsidiary condition ϑ', which states that ABC does not degenerate to a line, is not really necessary for the conjecture to hold.

Example 3.14 (M. Paterson's problem) Consider a triangle ABC. Erect three similar isosceles triangles A_1BC, AB_1C, and ABC_1 on the sides BC, CA, and AB, respectively. Show that AA_1, BB_1, and CC_1 are concurrent. How does the point of concurrency move as the areas of the three similar triangles are varied between 0 and ∞? See Figure 3.9. $\square$

For proving the concurrency, let $A = (0, 0)$, $B = (u_1, 0)$, $C = (u_2, u_3)$, $C_1 = (x_2, x_1)$, $A_1 = (x_4, x_3)$, $B_1 = (x_6, x_5)$, and $O = (x_8, x_7)$, where O denotes the point of concurrency. With these coordinates, Chou models hypothesises

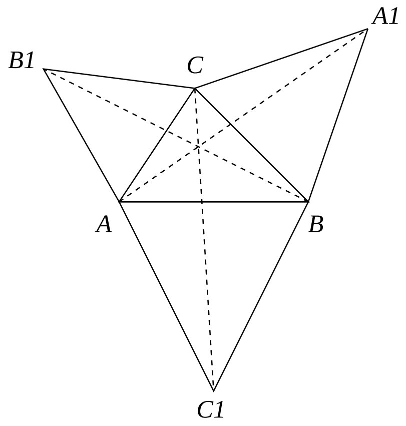

Figure 3.9: M. Paterson's problem (Example 3.14)

$h_1, \ldots, h_7$ and conclusion g as follows:

$$\begin{array}{rcll}
h_1 & \equiv & 2u_1x_1 - u_1^2 = 0 & C_1A \equiv C_1B \\
h_2 & \equiv & (u_1u_3x_1 + (u_1u_2 - u_1^2)u_4)x_3 + & \\
 & & ((-u_1u_2 + u_1^2)x_1 + u_1u_3u_4)x_2 - & \\
 & & u_1^2u_3x_1 + (-u_1^2u_2 + u_1^3)u_4 = 0 & \tan(\angle BAC_1) = \tan(\angle CBA_1) \\
h_3 & \equiv & (2u_2 - 2u_1)x_3 + & \\
 & & 2u_3x_2 - u_3^2 - u_2^2 + u_1^2 = 0 & A_1B \equiv A_1C \\
h_4 & \equiv & (u_1u_3x_1 + u_1u_2u_4)x_5 + & \\
 & & (-u_1u_2x_1 + u_1u_3u_4)x_4 + & \\
 & & (-u_1u_3^2 - u_1u_2^2)u_4 = 0 & \tan(\angle BAC_1) = \tan(\angle ACB_1) \\
h_5 & \equiv & 2u_2x_5 + 2u_3x_4 - u_3^2 - u_2^2 = 0 & B_1A \equiv B_1C \\
h_6 & \equiv & x_4x_7 + (-x_5 + u_1)x_6 - u_1x_4 = 0 & O \in BB_1 \\
h_7 & \equiv & x_2x_7 - x_3x_6 = 0 & O \in AA_1 \\
g & \equiv & (u_4 - u_3)x_7 + (-x_1 + u_2)x_6 + & \\
 & & u_3x_1 - u_2u_4 = 0 & \mathrm{collinear}(O, C, C_1).
\end{array}$$

After 610 ms, generic quantifier elimination yields $\varphi' \equiv$ true and the following non-degeneracy conditions:

$$\begin{array}{rcl}
\vartheta' & \equiv & u_1^2 - 2u_1u_2 + u_2^2 + u_3^2 \neq 0 \wedge u_1u_3 - 2u_1u_4 + 2u_2u_4 \neq 0 \wedge \\
 & & u_1u_3 + 2u_2u_4 \neq 0 \wedge u_1u_3 - 2u_2u_4 \neq 0 \wedge u_1 \neq 0 \wedge u_2 \neq 0.
\end{array}$$

The first condition $(u_1 - u_2)^2 + u_3^2 \neq 0$ is equivalent to $u_1 \neq u_2 \wedge u_3 \neq 0$. This together with $u_1 \neq 0 \wedge u_2 \neq 0$ states that ABC be a proper triangle. The remaining negated equations are non-collinearity conditions.

The next question is on the locus of the point O of concurrency as the area of the three similar triangles is varied. This is actually rather theorem finding than just theorem proving. We choose new coordinates as follows: $A = (0,0)$, $B = (u_1, 0)$, $C = (u_2, u_3)$, $O = (x, y)$, $C_1 = (x_2, x_1)$, $B_1 = (x_4, x_3)$. The coordinates x and y of O are extra parameters. The hypothesises $h_1, \ldots, h_4$

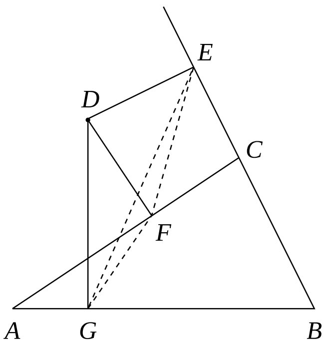

Figure 3.10: Gergonne's theorem (Example 3.15)

and the conclusion g are as follows:

$$\begin{array}{rcll} h_1 & \equiv & (y-u_3)x_2+(-x+u_2)x_1- & \\ & & u_2y+u_3x=0 & C_1 \in OC \\ h_2 & \equiv & 2u_1x_2-u_1^2=0 & C_1A \equiv C_1B \\ h_3 & \equiv & yx_4+(-x+u_1)x_3-u_1y=0 & B_1 \in OB \\ h_4 & \equiv & 2u_2x_4+2u_3x_3-u_3^2-u_2^2=0 & B_1A \equiv B_1C \\ g & \equiv & (u_1u_3x_2+u_1u_2x_1)x_4+ & \\ & & (-u_1u_2x_2+u_1u_3x_1)x_3+ & \\ & & (-u_1u_3^2-u_1u_2^2)x_1=0 & \tan(\angle BAC_1)=\tan(\angle ACB_1). \end{array}$$

On input of $\forall x_1 \dots \forall x_4(h_1 \wedge \cdots \wedge h_4 \longrightarrow g)$ we obtain by generic quantifier elimination the elimination result

$$\begin{array}{rcl} \varphi' & \equiv & u_1^2u_2y+u_1^2u_3x-2u_1^2xy+u_1u_2^2y-2u_1u_2u_3x+2u_1u_2xy-u_1u_3^2y- \\ & & u_1u_3x^2+u_1u_3y^2-2u_2^2xy+2u_2u_3x^2-2u_2u_3y^2+2u_3^2xy=0 \vee \\ & & u_1=0 \vee (u_2=0 \wedge u_3=0) \end{array}$$

under the subsidiary conditions

$$\vartheta' \equiv u_1u_2-u_2x-u_3y \neq 0 \wedge u_2-x \neq 0 \wedge y \neq 0.$$

Chou gives the following solution, which seems not compatible with ours:

$$R_0 = u_1^2(u_3^2+u_2^2)\big(ay^2+2bxy+cu_3x^2+(u_1u_3^2-u_1u_2^2-u_1^2u_2)y+(2u_1u_2-u_1^2)u_3x\big).$$

We suspect that there is a typo in R_0, such that it should actually read as follows:

$$R_0 = u_1^2(u_3^2+u_2^2)\big(ay^2+2bxy+\underline{cx^2}+(u_1u_3^2-u_1u_2^2-u_1^2u_2)y+(2u_1u_2-u_1^2)u_3x\big).$$

If this is the case, then our φ' provides a factorization of Chou's R_0, where the conditions $-u_1^2=0$ and $u_3^2+u_2^2=0$ are equivalently replaced by $u_1=0$ and $u_2=0 \wedge u_3=0$, respectively. This example takes 110 ms.

Example 3.15 (Gergonne's theorem) Consider a triangle ABC and a moving point D. From D draw three perpendicular lines with feet E, F, and G to the three sides of the triangle ABC. What is the locus of the point D for the area of the triangle EFG to remain constant? See Figure 3.10. □

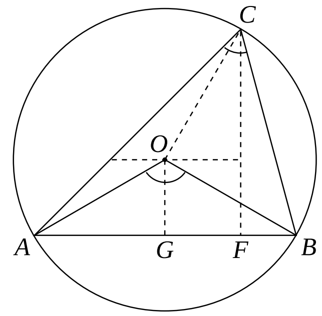

Figure 3.11: Angle at circumference vs. angle at center (Example 3.16)

Similar to the previous one, this final example taken from Chou is also concerned with finding theorems rather than proving theorems. It is a generalization of Simson's theorem discussed as Example 3.9. We let the area of EFG be $a/2$, and choose coordinates $A = (0,0)$, $B = (u_1, 0)$, $C = (u_2, u_3)$, $D = (x, y)$, $E = (x_2, x_1)$, $F = (x_4, x_3)$, and $G = (x, 0)$. The hypothesises $h_1, \dots, h_4$ and the conclusion g read as follows:

$$\begin{array}{rcll} h_1 & \equiv & u_3x_2 + (-u_2 + u_1)x_1 - u_1u_3 = 0 & E \in \mathrm{BC} \\ h_2 & \equiv & (u_2 - u_1)x_2 + u_3x_1 - u_3y + (-u_2 + u_1)x = 0 & ED \perp BC \\ h_3 & \equiv & u_3x_4 - u_2x_3 = 0 & F \in AC \\ h_4 & \equiv & u_2x_4 + u_3x_3 - u_3y - u_2x = 0 & FD \perp AC \\ g & \equiv & x_1x_4 + (-x_2 + x)x_3 - xx_1 + a = 0 & A(EFG) = a/2. \end{array}$$

After 160 ms of computation time, we get by generic quantifier elimination the following elimination result:

$$\begin{array}{rcl} \varphi' & \equiv & au_1^2u_2^2 + au_1^2u_3^2 - 2au_1u_2^3 - 2au_1u_2u_3^2 + au_2^4 + 2au_2^2u_3^2 + au_3^4 + \\ & & u_1^2u_2u_3^2y - u_1^2u_3^3x - u_1u_2^2u_3^2y - u_1u_3^4y + u_1u_3^3x^2 + u_1u_3^3y^2 = 0. \end{array}$$

Chou obtains exactly the same equation for the locus of point D.

We furthermore obtain the non-degeneracy conditions

$$\vartheta' \equiv u_1^2 - 2u_1u_2 + u_2^2 + u_3^2 \neq 0 \wedge u_2 \neq 0 \wedge u_3 \neq 0.$$

The first condition $(u_1 + u_2)^2 + u_3^2 \neq 0$ is obviously implied by $u_3 \neq 0$ and can thus be dropped. The condition $u_2 \neq 0$ states that $\angle BAC$ is not a right angle, and the condition $u_3 \neq 0$ states that ABC is a proper triangle.

3.8.2 Further Examples Requiring a Real Prover

Example 3.16 (Angle at circumference) Let O be the center of the circumcircle of a triangle ABC. If O does not lie outside of ABC, then $\angle ACB = \angle AOB/2$. See Figure 3.11. □

At the end of the discussion of this example, we will generalize the conjecture dropping the restriction that O must be inside ABC .

We choose coordinates $A = (-u_1, 0)$, $B = (u_1, 0)$, and $C = (u_2, u_3)$ for the triangle. The center $O = (0, x_1)$ of the circumcircle is then determined via its radius $r^2 = u_1^2 + x_1^2 = u_2^2 + (u_3 - x_1)^2$.

Denoting by $F = (0, u_3)$ the foot of the perpendicular from C onto AB, we split the angle at circumference: $\angle ACB = \angle ACF + \angle FCB$. Its two components are encoded into tangents with

$$\tan(\angle ACF) = \frac{u_1 + u_2}{u_3} \quad \text{and} \quad \tan(\angle FCB) = \frac{u_1 - u_2}{u_3}.$$

By an addition theorem for tangents we know that

$$\tan(\angle ACB) = \tan(\angle ACF + \angle FCB) = \frac{\tan(\angle ACF) + \tan(\angle FCB)}{1 - \tan(\angle ACF)\tan(\angle FCB)}.$$

Denote by $G = (0, x_1)$ the foot of the perpendicular from O onto AB. Then $\angle AOG$ is obviously half the angle $\angle AOB$ at center. By encoding into tangents we obtain

$$\tan\Bigl(\frac{\angle AOB}{2}\Bigr) = \tan(\angle AOG) = \frac{u_1}{x_1}.$$

The conclusion of our conjecture states that $\angle ACB = \angle AOB/2$, which we describe by $\tan(\angle ACB) = \tan(\angle AOB/2)$.

We finally arrive at the following algebraic translation in which the various tangents occur as real variables t_1, t_2, t, and t':

$$\begin{aligned} \varphi \quad \equiv \quad & \forall r \forall x_1 \forall t_1 \forall t_2 \forall t \forall t' \bigl(r^2 = u_1^2 + x_1^2 \wedge r^2 = u_2^2 + (u_3 - x_1)^2 \wedge \\ & u_3 t_1 = u_1 + u_2 \wedge u_3 t_2 = u_1 - u_2 \wedge (1 - t_1 t_2) t = t_1 + t_2 \wedge \\ & x_1 t' = u_1 \longrightarrow t = t' \bigr). \end{aligned}$$

After 70 ms of computation time, we obtain by generic quantifier elimination $\varphi' \equiv \text{true}$ and the non-degeneracy conditions $\vartheta' \equiv u_1 \neq 0 \wedge u_3 \neq 0$ stating that ABC is a non-degenerate triangle.

Note that there is no condition prohibiting that AOB degenerates to a line. Our proof thus covers the important special case that $\angle AOB = 180°$ and $\angle ACB = 90°$: *Thales' theorem.*

Regular quantifier elimination and successive Gröbner simplification yields after 150 ms an exact quantifier-free equivalent of our algebraic translation: $\varphi' \equiv u_1 \neq 0 \vee u_2 \neq 0 \vee u_3 \neq 0$.

Note that we have not added to our hypothesises the constraint that the center O of the circumcircle must lie inside the triangle ABC. Suppose now that O lies outside of ABC. That is, in Figure 3.11 we choose a point C below the line AB. Then the angle at circumcircle is obviously larger than with C chosen above AB. In fact, any geometry textbook will state that then the angle at circumference is $\angle BCA = 180° - \angle AOB$ for the smaller segment of the circumcircle. Let us, however, not switch to $\angle BCA$, but continuously consider

$$\angle ACB = 360° - (180° - \angle AOB) = 180° + \angle AOB.$$

Since certainly $\tan(\alpha) = \tan(180° + \alpha)$ for any angle α, our translation turns out to be completely uniform. In other words, we have already proved the generalized theorem.

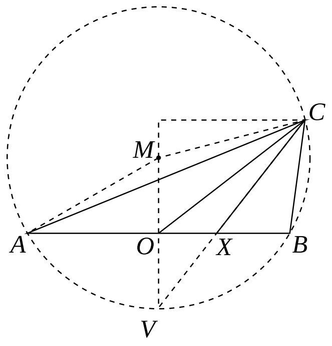

Figure 3.12: The median bisector theorem (Example 3.5 and Example 3.17)

Example 3.17 (Median bisector theorem) Consider a non-isosceles triangle ABC. The median over the side AB is always greater than the interior bisector on the same side. See Figure 3.12. □

This example and some ideas for its algebraic translation are taken from [Wu92], pp. 7–8. To prove the theorem, we take coordinates such that

$$A = (-1, 0), \quad B = (1, 0), \quad C = (u_1, u_2).$$

We may wlog. assume that $u_2 > 0$. Since the origin $O = (0, 0)$ is the mid point of AB, we have that CO is the median on AB.

We construct the bisector using the geometric theorem of the previous Example 3.16. The center $M = (0, x_1)$ of the circumcircle of the triangle ABC is determined by its radius $r > 0$:

$$r^2 = 1 + x_1^2 = u_1^2 + (u_2 - x_1)^2.$$

We know that $\angle ACB = \angle AMB/2$. Let now $V = (0, x_1 - r)$ be the lower extremity of the circumcircle. Since our circumcircle is certainly also the circumcircle of the triangle AVC and $\angle AMV = \angle AMB/2$, it follows that $\angle ACV = \angle ACB/2$. If we thus let $X = (x_2, 0)$ be the intersection between CV and AB,

$$\frac{u_2}{u_1 - x_2} x_2 + (x_1 - r) = 0,$$

then CX is the interior bisector on the side AB.

We come to the following algebraic translation:

$$\begin{aligned}\varphi \quad \equiv \quad & \forall x_1 \forall r \forall x_2 \big(u_2 > 0 \land r > 0 \land r^2 = 1 + x_1^2 \land r^2 = u_1^2 + (u_2 - x_1)^2 \land \\ & u_2 x_2 + (x_1 - r)(u_1 - x_2) = 0 \longrightarrow u_1^2 + u_2^2 > (u_1 - x_2)^2 + u_2^2\big).\end{aligned}$$

After 380 ms elimination time, we obtain the non-degeneracy conditions $\vartheta' \equiv u_1 \neq 0 \land u_2 \neq 0$, stating that ABC is non-degenerate, plus a quantifier-free equivalent φ' containing 13 atomic formulas:

$$\begin{aligned}\varphi' \quad \equiv \quad & u_1^4 + 2u_1^2 u_2^2 - 2u_1^2 + u_2^4 + 2u_2^2 + 1 < 0 \lor \\ & \big((u_1^4 u_2 - 2u_1^2 u_2 + u_2^5 + 2u_2^3 + u_2 \geqslant 0 \lor u_1^2 + 2u_1 + u_2^2 + 1 = 0 \lor\end{aligned}$$

$$u_1^2 - 2u_1 + u_2^2 + 1 = 0 \vee u_2 \leqslant 0 \vee u_1^4 + 2u_1^2 - u_2^4 - 2u_2^2 - 1 > 0 \wedge$$
$$2u_1^2 - 2u_2^2 - 1 > 0) \wedge (u_1^4 u_2 - 2u_1^2 u_2 + u_2^5 + 2u_2^3 + u_2 \leqslant 0 \vee$$
$$u_1^4 + 2u_1^2 - u_2^4 - 2u_2^2 - 1 > 0 \vee 2u_1^2 - 2u_2^2 - 1 < 0 \vee$$
$$u_1^2 + 2u_1 + u_2^2 + 1 = 0 \vee u_1^2 - 2u_1 + u_2^2 + 1 = 0 \vee u_2 \leqslant 0)\big).$$

This result φ' can hardly be interpreted geometrically. Luckily, it turns out to be equivalent to "true" under the assumptions ϑ'. QEPCAD yields within 1 s that

$$\forall u_1 \forall u_2 \big(\vartheta'(u_1, u_2) \longrightarrow \varphi'(u_1, u_2)\big),$$

which finally proves the conjecture φ wrt. ϑ'. Due to the degree restrictions, this final step cannot be performed via regular quantifier elimination by virtual substitution.

The result φ' of the generic quantifier elimination obviously suffered from insufficient simplification. This weakness could be overcome by another quantifier elimination step, which shows that quantifier elimination is an extremely universal, flexible, and powerful tool.

In case that the final proving step $\vartheta' \longrightarrow \varphi'$ does not succeed, one has to carefully analyze the situation. Assume, for instance, that we model Example 3.17 slightly more general by not normalizing the side AB to 2 but to $2u_3$ by taking coordinates $A = (-u_3, 0)$, $B = (u_3, 0)$, and the rest as above. The algebraic translation φ then becomes

$$\begin{aligned}\overline{\varphi} \equiv\ & \forall x_1 \forall r \forall x_2 \big(u_2 > 0 \wedge r > 0 \wedge r^2 = u_3^2 + x_1^2 \wedge r^2 = u_1^2 + (u_2 - x_1)^2 \wedge \\ & u_2 x_2 + (x_1 - r)(u_1 - x_2) = 0 \longrightarrow u_1^2 + u_2^2 > (u_1 - x_2)^2 + u_2^2\big).\end{aligned}$$

Generic quantifier elimination on this new $\overline{\varphi}$ yields after 530 ms $\overline{\vartheta}' \equiv \vartheta'$, and a quantifier-free formula $\overline{\varphi}'$ with 21 atomic formulas similar to φ' above. This time, however, QEPCAD obtains after 3 s the result "false" on input of

$$\forall u_1 \forall u_2 \forall u_3 \big(\overline{\vartheta}'(u_1, u_2, u_3) \longrightarrow \overline{\varphi}'(u_1, u_2, u_3)\big).$$

This means that there are non-degeneracy conditions missing, which certainly cannot be detected during this elimination step, since all the remaining parameters have become quantified variables. We happen to know that the remaining non-degeneracy condition will most probably involve u_3, and we thus apply QEPCAD to

$$\forall u_1 \forall u_2 \big(\overline{\vartheta}'(u_1, u_2, u_3) \longrightarrow \overline{\varphi}'(u_1, u_2, u_3)\big)$$

leaving u_3 as a parameter. After 3 s we obtain the expected condition $u_3 \neq 0$, which states that that triangle must not degenerate on the line AB. Generally, one can use such eliminations on the final result to remove at least some "uncritical" variables, and hope to come to an interpretable result.

Example 3.18 (Pedoe's inequality) Consider two arbitrary triangles ABC and $A'B'C'$ with sides a, b, c and a', b', c' respectively. The areas Δ and Δ' of this triangles satisfy the following inequality:

$$a'^2(b^2 + c^2 - a^2) + b'^2(c^2 + a^2 - b^2) + c'^2(a^2 + b^2 - c^2) \geqslant 16\Delta'\Delta. \quad \square$$

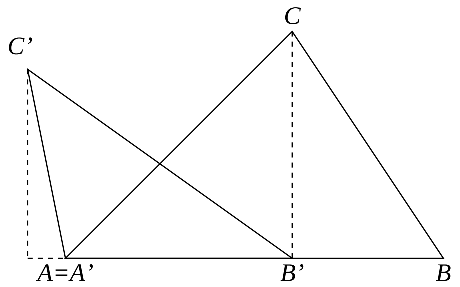

Figure 3.13: Modeling Pedoe's inequality (Example 3.18)

This example is taken from Wang [Wan95b], pp. 158–160. We choose coordinates $A = A' = (0,0)$, $B = (u_1, 0)$, $B' = (u_2, 0)$, $C = (u_3, u_4)$, $C' = (u_5, u_6)$. Then the triangle side lengths a, b, and c are determined by

$$a^2 = (u_1 - u_3)^2 + u_4^2, \quad b^2 = u_3^2 + u_4^2, \quad c^2 = u_1^2.$$

The sides a', b', and c' are obtained analogously. The areas are determined by $\Delta = cu_4/2$ and $\Delta' = c'u_6/2$, respectively. With these naming conventions the conclusion can be stated exactly as in the conjecture in Example 3.4. Our algebraic translation reads as follows:

$$\begin{aligned}\varphi \quad \equiv \quad & \forall a \forall b \forall c \forall a' \forall b' \forall c' \forall \Delta \forall \Delta' \big(a^2 = (u_1 - u_3)^2 + u_4^2 \wedge b^2 = u_3^2 + u_4^2 \wedge \\ & c^2 = u_1^2 \wedge a'^2 = (u_2 - u_5)^2 + u_6^2 \wedge b'^2 = u_5^2 + u_6^2 \wedge c'^2 = u_2^2 \wedge \\ & \Delta = cu_4/2 \wedge \Delta' = c'u_6/2 \longrightarrow \\ & a'^2(b^2 + c^2 - a^2) + b'^2(c^2 + a^2 - b^2) + c'^2(a^2 + b^2 - c^2) \geqslant 16\Delta'\Delta \big).\end{aligned}$$

Applying generic quantifier elimination to φ yields after 150 ms no non-degeneracy conditions at all, formally $\vartheta' \equiv \mathrm{true}$, and the following quantifier-free formula:

$$\begin{aligned}\varphi' \quad \equiv \quad & (u_1^2u_5^2 + u_1^2u_6^2 - 2u_1u_2u_3u_5 + 2u_1u_2u_4u_6 + u_2^2u_3^2 + u_2^2u_4^2 \geqslant 0 \vee \\ & u_1^2 - 2u_1u_3 + u_3^2 + u_4^2 < 0 \vee u_2^2 - 2u_2u_5 + u_5^2 + u_6^2 < 0) \wedge \\ & (u_1^2u_5^2 + u_1^2u_6^2 - 2u_1u_2u_3u_5 - 2u_1u_2u_4u_6 + u_2^2u_3^2 + u_2^2u_4^2 \geqslant 0 \vee \\ & u_1^2 - 2u_1u_3 + u_3^2 + u_4^2 < 0 \vee u_2^2 - 2u_2u_5 + u_5^2 + u_6^2 < 0).\end{aligned}$$

Regular quantifier elimination checks within 1.3 s that

$$\forall u_1 \dots \forall u_6 \varphi'(u_1, \dots, u_6) \longleftrightarrow \mathrm{true}.$$

The entire procedure can certainly be performed also in one step by applying regular quantifier elimination to

$$\forall u_1 \dots \forall u_6 \forall a \forall b \forall c \forall a' \forall b' \forall c' \forall \Delta \forall \Delta' \varphi(u_1, \dots, u_6, a, b, c, a', b', c', \Delta, \Delta').$$

Proceeding this way, however, we would loose the chance to detect subsidiary conditions during the first generic step.

Wang [Wan95b] has proved Pedoe's inequality putting more human intelligence into the algebraic translation. His translation is, up to different names for the coordinates, the inequality

$$u_5^2 + u_1^2 u_6^2 - 2u_1u_2u_3u_5 - 2u_1u_2u_4u_6 + u_2^2u_3^2 + u_2^2u_4^2 \geqslant 0$$

contained in our φ'. Wu [Wu87] has automatically proved Pedoe's inequality in a purely algebraic setting by determining the extremal values of the areas Δ and Δ'.

Example 3.19 (Qin–Heron's formula) Determine the area Δ of a triangle ABC in terms of its three sides. □

Here we do once more not only prove but actually find a theorem. The Qin–Heron formula has been mechanically derived also by Wang [Wan95b] and by Wu [Wu86b].

We locate $A = (-u_1, 0)$, $B = (u_1, 0)$, and $C = (u_2, u_3)$. Then the side lengths are determined as follows:

$$a^2 = (u_1 - u_2)^2 + u_3^2, \quad b^2 = (-u_1 - u_2)^2 + u_3^2, \quad \text{and} \quad c = 2u_1.$$

On the other hand we know $\Delta = cu_3/2 = u_1u_3$. This yields our translation φ with parameters a, b, and c:

$$\begin{aligned} \varphi \quad \equiv \quad & \exists u_1 \exists u_2 \exists u_3 \big(\Delta = u_1u_3 \wedge \\ & a^2 = (u_1-u_2)^2 + u_3^2 \wedge b^2 = (-u_1-u_2)^2 + u_3^2 \wedge c = 2u_1\big). \end{aligned}$$

After 136 ms, generic quantifier elimination obtains $\vartheta' \equiv c \neq 0$, i.e., the points A and B are distinct. The quantifier-free result looks after Gröbner simplification (60 ms) as follows:

$$\varphi' \equiv a^4 - 2a^2b^2 - 2a^2c^2 + b^4 - 2b^2c^2 + c^4 + 16\Delta^2 = 0 \wedge a^2c^2 - 4\Delta^2 \geqslant 0.$$

The inequality follows from the equation as regular quantifier elimination on

$$\forall a \forall b \forall c \forall \Delta (a^4 - 2a^2b^2 - 2a^2c^2 + b^4 - 2b^2c^2 + c^4 + 16\Delta^2 = 0 \longrightarrow a^2c^2 - 4\Delta^2 \geqslant 0)$$

proves within less than 10 ms. Setting $s = (a+b+c)/2$, the equation can be rewritten as Heron's formula $\Delta^2 = s(s-a)(s-b)(s-c)$.

Example 3.20 (MacLane) Consider eight points $A, \ldots, H$ such that the following eight triples are collinear ABD, BCE, CDF, DEG, EFH, FGA, GHB, HAC. Then all eight points lie on a line. □

This example is originally due to MacLane [Mac36]. It holds in the real plane but amazingly fails in the complex plane.

We adopt a quite sophisticated translation proposed by Kutzler [Kut88]. The coordinates for the points are $A = (0, 0)$, $B = (x_b, 0)$, $C = (x_c, y_c)$, $D = (x_d, 0)$, $E = (x_e, y_e)$, $\ldots$, $H = (x_h, y_h)$. Note that three points (x_1, y_1), (x_2, y_2), and (x_3, y_3) are collinear if and only if

$$(y_2 - y_1)x_3 + (x_1 - x_2)y_3 + (x_2y_1 - x_1y_2) = 0.$$

The entire algebraic translation reads as follows:

$$\begin{aligned}\varphi \quad \equiv \quad & \forall y_h \forall x_e \forall y_e \forall x_f \forall y_f \forall x_g \forall y_g (x_h y_c - x_c y_h = 0 \land x_g y_f - x_f y_g = 0 \land \\ & x_b y_e - x_c y_e + x_e y_c - x_b y_c = 0 \land x_b y_h - x_g y_h + x_h y_g - x_b y_g = 0 \land \\ & x_c y_f - x_d y_f - x_f y_c + x_d y_c = 0 \land x_d y_g - x_e y_g + x_g y_e - x_d y_e = 0 \land \\ & x_e y_h - x_f y_h + x_h y_f - x_e y_f - x_h y_e + x_f y_e = 0 \longrightarrow x_b y_c = 0).\end{aligned}$$

Generic quantifier elimination yields after 4.3 s the elimination result $\varphi' \equiv$ true. The subsidiary conditions are the following:

$$\begin{aligned}\vartheta' \quad \equiv \quad & x_b^2 x_h^2 - x_b x_c x_d x_h - x_b x_d x_h^2 + x_c^2 x_d^2 - x_c x_d^2 x_h + x_d^2 x_h^2 \neq 0 \land \\ & x_b^2 x_h - x_b x_c x_d - x_b x_c x_h + x_c x_d x_h \neq 0 \land \\ & x_b^2 x_h - x_b x_c x_d - x_b x_d x_h + x_d^2 x_h \neq 0 \land \\ & x_b - x_c \neq 0 \land x_b - x_d \neq 0 \land x_b - x_h \neq 0 \land x_c - x_d \neq 0 \land \\ & x_c - x_h \neq 0 \land x_c \neq 0 \land x_d \neq 0 \land x_h \neq 0.\end{aligned}$$

Except for the first three inequalities, all these conditions are obvious geometric non-degeneracy conditions. The first inequality follows from the geometric ones as regular quantifier elimination shows in 330 ms, and is thus redundant.

We rerun our method with the option permitting only monomial assumptions: After 7.8 s we obtain a quantifier-free formula φ' with 204 atomic formulas and the following non-degeneracy conditions ϑ':

$$x_c \neq 0 \land x_h \neq 0 \land y_c \neq 0.$$

Within 15.6 s, our theory generator obtains ϑ'' by adding to ϑ' assumptions, which make it sufficient for φ' and thus for φ:

$$\begin{aligned}\vartheta'' \quad \equiv \quad & x_b^2 x_h^2 - x_b x_c x_d x_h - x_b x_d x_h^2 + x_c^2 x_d^2 - x_c x_d^2 x_h + x_d^2 x_h^2 \neq 0 \land \\ & x_b - x_d \neq 0 \land x_c - x_h \neq 0 \land x_c \neq 0 \land x_d \neq 0 \land x_h \neq 0 \land y_c \neq 0.\end{aligned}$$

Again, our general method proves that the first negated equation follows from the remaining non-degeneracy conditions (140 ms).

We have finally proved that MacLane's conjecture holds under the following non-degeneracy conditions, which all prohibit coincidence of certain points:

$$x_b - x_d \neq 0 \land x_c - x_h \neq 0 \land x_c \neq 0 \land x_d \neq 0 \land x_h \neq 0 \land y_c \neq 0.$$

Example 3.21 (Pompeiu's theorem) Let ABC be a equilateral triangle, and let P be some point not on the circumcircle of ABC, then the segments AP, BP, CP can be used to form a triangle. That is, the sum of the lengths of any two of them exceeds the length of the third one. See Figure 3.14. □

This example is taken from McPhee, Chou, and Gao [MCG94]. We first literally adopt the algebraic formulation given there.

$$\begin{aligned}\varphi \quad \equiv \quad & \forall x_0 \forall x_1 \forall x_2 \forall x_3 \forall x_4 \forall u_0 \forall u_1 (4x_0^2 - 3 = 0 \land 2x_1 + 1 = 0 \land \\ & x_4^2 - 2x_4 + x_3^2 - u_0^2 + 1 = 0 \land u_0 > 0 \land \\ & x_4^2 - 2x_1 x_4 + x_3^2 + 2x_0 x_3 + x_1^2 + x_0^2 - u_1^2 = 0 \land u_1 > 0 \land \\ & x_4^2 - 2x_1 x_4 + x_3^2 - 2x_0 x_3 - x_2^2 + x_1^2 + x_0^2 = 0 \land x_2 > 0 \land \\ & x_4^2 + x_3^2 - 1 \neq 0 \longrightarrow x_2 - u_0 - u_1 < 0).\end{aligned}$$

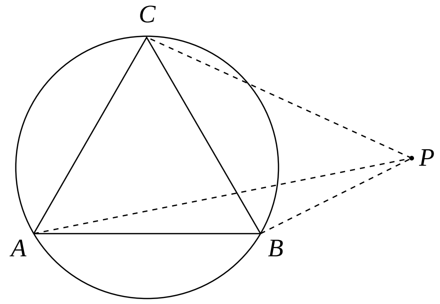

Figure 3.14: Pompeiu's theorem (Example 3.21)

Note that there are no parameters; all the variables are quantified. The non-degeneracy condition $\gamma \equiv x_4^2 + x_3^2 - 1 \neq 0$ is explicitly added as a hypothesis. Eliminating all the quantifiers by regular quantifier elimination yields $\varphi \equiv \mathrm{true}$ after 180 ms.

Next, we delete γ from the hypotheses, and eliminate only the dependent variables u_0, u_1, and x_2:

$$\begin{aligned}\overline{\varphi} \quad \equiv \quad & \forall x_1 \forall x_2 \forall u_0 (4x_0^2 - 3 = 0 \wedge 2x_1 + 1 = 0 \wedge x_4^2 - 2x_4 + x_3^2 - u_0^2 + 1 = 0 \wedge \\ & u_0 > 0 \wedge x_4^2 - 2x_1x_4 + x_3^2 + 2x_0x_3 + x_1^2 + x_0^2 - u_1^2 = 0 \wedge u_1 > 0 \wedge \\ & x_4^2 - 2x_1x_4 + x_3^2 - 2x_0x_3 - x_2^2 + x_1^2 + x_0^2 = 0 \wedge x_2 > 0 \longrightarrow \\ & x_2 - u_0 - u_1 < 0).\end{aligned}$$

Generic quantifier elimination yields after 80 ms a formula $\overline{\varphi}'$ containing 7 atomic formulas:

$$\begin{aligned}\overline{\varphi}' \quad \equiv \quad & 12x_0^2x_3^2 - 4x_0^2x_4^2 + 8x_0^2x_4 - 4x_0^2 - 4x_1^2x_3^2 - 4x_1^2x_4^2 + 8x_1^2x_4 - 4x_1^2 + \\ & 8x_1x_3^2x_4 + 8x_1x_4^3 - 16x_1x_4^2 + 8x_1x_4 - 3x_3^4 - 6x_3^2x_4^2 + 4x_3^2x_4 - 2x_3^2 - \\ & 3x_4^4 + 4x_4^3 + 2x_4^2 - 4x_4 + 1 < 0 \vee \\ & 4x_0^2 - 3 \neq 0 \vee x_0^2 + 2x_0x_3 + x_1^2 - 2x_1x_4 + x_3^2 + x_4^2 \leqslant 0 \vee \\ & x_0^2 - 2x_0x_3 + x_1^2 - 2x_1x_4 + x_3^2 + x_4^2 \leqslant 0 \vee \\ & 4x_0x_3 + x_3^2 + x_4^2 - 2x_4 + 1 > 0 \vee 2x_1 + 1 \neq 0 \vee x_3^2 + x_4^2 - 2x_4 + 1 \leqslant 0.\end{aligned}$$

Unfortunately, there are no assumptions made: $\overline{\vartheta}' \equiv \mathrm{true}$.

We proceed by eliminating also x_0 and x_1 in order to find a necessary and sufficient condition in terms of the point (x_3, x_4) for the theorem to hold. This elimination yields 12 atomic formulas after 100 ms, again without any assumptions. The Gröbner simplifier reduces the result within 90 ms to the following necessary and sufficient condition $\overline{\varphi}''(x_3, x_4)$ for $\overline{\varphi}$ to hold.

$$\overline{\varphi}'' \equiv x_3^2 + x_4^2 - 1 \neq 0 \vee 2x_4^2 - x_4 - 1 > 0 \vee 2x_4 + 1 = 0 \vee x_4 + 2 \leqslant 0 \vee x_4 - 1 \geqslant 0.$$

An improved standard simplifier, which is not part of REDLOG 2.0, simplifies $\overline{\varphi}''$ in less than 10 ms by factorizing the second condition $2x_4^2 - x_4 - 1 > 0$ of $\overline{\varphi}''$ into $(x_4 - 1 > 0 \wedge 2x_4 + 1 > 0) \vee (x_4 - 1 < 0 \wedge 2x_4 + 1 < 0)$, and then applying the ordinary standard simplification. This yields our final result:

$$\overline{\varphi}''' \equiv x_3^2 + x_4^2 - 1 \neq 0 \vee 2x_4 + 1 \leqslant 0 \vee x_4 - 1 \geqslant 0.$$

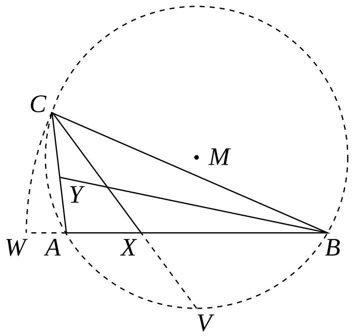

Figure 3.15: A variant of the Steiner–Lehmus theorem (Example 3.22)

In $\overline{\varphi}'''$, the inequality $x_3^2 + x_4^2 - 1 \neq 0$ states that (x_3, x_4) be not on the circumcircle. Surprisingly, the condition $2x_4 + 1 \leqslant 0$ allows (x_3, x_4) to be on the circumcircle provided that it lies below or on the basis BC. Similarly the condition $x_4 - 1 \geqslant 0$ allows (x_3, x_4) to coincide with the vertex A. The reason for this is that the formulation $x_2 - u_0 - u_1 < 0$ of the conclusion is asymmetric leaving out the cases $u_0 - x_2 - u_1 < 0$ and $u_1 - u_0 - x_2 < 0$. That is, the vertices A, B, and C are excluded already by the hypothesis.

Example 3.22 (Steiner–Lehmus theorem, variant) Assume that ABC is a triangle such that $AB > AC$. Then the angle bisector from B to AC is longer than the angle bisector from C to AB. That is, the longer bisector goes to the shorter side. See Figure 3.15. □

This example is taken from McPhee, Chou, and Gao [MCG94]. In its original form, the Steiner–Lehmus theorem states that *any triangle with two equal internal bisectors is isosceles.* Its contrapositive follows immediately from the variant discussed here.

In our algebraic translation, we take coordinates $A = (-1, 0)$, $B = (1, 0)$, and $C = (u_1, u_2)$ for the triangle. We assume wlog. that both C and the center $M = (0, x_1)$ of the circumcircle of ABC are above AB:

$$h_1 \equiv u_2 \geqslant 0 \wedge x_1 \geqslant 0.$$

The center $M = (0, x_1)$ of the circumcircle is determined by its radius:

$$h_2 \equiv r^2 = 1 + x_1^2 = u_1^2 + (u_2 - x_1)^2.$$

We can now construct the bisector of $\angle ACB$ as discussed with the median bisector theorem in Example 3.17: The point $V = (0, x_2)$ is the lower extremity of the circumcircle:

$$h_3 \equiv x_2 \leqslant 0 \wedge r^2 = (x_2 - x_1)^2,$$

and $X = (x_3, 0)$ is the intersection between CV and AB. That is, the points CXV are collinear:

$$h_4 \equiv u_1 x_2 + u_2 x_3 - x_2 x_3 = 0.$$

See Example 3.20 for details on formulating collinearity. The line segment CX is the bisector on the side AB.

For constructing the bisector on AC using the above technique, we would have to guarantee that both M and C lie on the same side of AC within the circumcircle, which would cause tedious case distinctions. We thus construct the bisector on AC more straightforwardly: Let $W = (x_4, 0)$ be the point on the line AB that lies left of B with distance BC:

$$h_5 \equiv x_4 \leqslant 1 \wedge (x_4 - 1)^2 = (u_1 - 1)^2 + u_2^2.$$

Then the foot $Y = (x_5, x_6)$ of the bisector on AC is the unique point with equal distance to both W and C and with AYC collinear:

$$h_6 \equiv (x_4 - x_5)^2 + x_6^2 = (u_1 - x_5)^2 + (u_2 - x_6)^2 \wedge u_1 x_6 - u_2 x_5 - u_2 + x_6 = 0.$$

Finally we add to our hypothesises the fact that AC is shorter than AB:

$$h_7 \equiv (-1 - u_1)^2 + u_2^2 < 2^2,$$

and state the conclusion that CX is shorter than BY:

$$g \equiv (u_1 - x_3)^2 + u_2^2 < (x_5 - 1)^2 + x_6^2.$$

On input of $\varphi \equiv \forall x_6 \ldots \forall x_1 \forall r (h_1 \wedge \cdots \wedge h_7 \longrightarrow g)$, generic quantifier elimination yields after 106 s a quantifier-free formula φ' containing 231 atomic formulas plus the subsidiary conditions

$$\vartheta' \equiv u_1^2 - 2u_1 + u_2^2 - 3 \neq 0 \wedge u_1 \neq 0 \wedge u_2 \neq 0.$$

The first condition can be rewritten as $(u_1 - 1)^2 + u_2^2 \neq 4$, i.e., the distance between B and C is different from 2. In other words, the triangle ABC is not isosceles.

QEPCAD proves within 111 s that $\forall u_1 \forall u_2 (\vartheta' \longrightarrow \varphi')$, while regular quantifier elimination by virtual substitution fails in doing so due to the degree restrictions.

3.9 Conclusions

We have applied our generic quantifier elimination by virtual substitution introduced in the previous chapter to automated theorem proving in geometry.

By automatically proving some well-known benchmark examples with our REDLOG implementation, we have demonstrated that our method can compete with the algebraic methods discussed in the literature so far. Since our method is a genuinely real—not complex—proof method, its application range exceeds that of the traditional methods:

- Generic quantifier elimination can handle not only polynomial equations but also ordering inequalities. The Steiner–Lehmus theorem 3.22 is one such example.
- Among the theorems that do not require any inequalities in their formulation, generic quantifier elimination can also prove those which are theorems in real geometry but which do not hold in the complex plane. In particular, there are theorems such as Example 3.20 by MacLane, which do not obviously fail over the complex numbers.

- With the conventional methods, one infers the universal validity over the reals from that over the complex numbers. If, however, some universal sentence does *not* hold over the complex numbers, one does not know anything about the situation over the reals. Our real method can, in contrast, not only prove but also disprove theorems. Compare the discussion of Example 3.13.

After specification of the independent parameters in the given problem, the assumptions made by the generic quantifier elimination for reducing the complexity of the elimination algorithm turn out to specify mostly the non-degeneracy conditions that are actually necessary for the input conjecture to hold. The apparent restriction of generic quantifier elimination in contrast to regular quantifier elimination has thus turned out as a valuable tool. In fact, generic quantifier elimination is, at least in the application area considered here, superior to regular quantifier elimination, which cannot straightforwardly applied in such a way that non-degeneracy conditions are detected. With examples of the type usually discussed for the traditional methods, the quantifier elimination result we obtain will be "true."

There are geometric assertions that do not generally hold without an additional assumption that in turn cannot be characterized as non-degeneracy condition. Such an assumption might require, e.g., that a triangle be isosceles, which is rather a "degeneracy condition" than a non-degeneracy condition. Note that this condition cannot be described by a negated equation. More generally, such an assumption can describe the locus of a certain points to fulfill the geometrical requirements of a theorem (Example 3.14 and Example 3.15). Moreover one can even derive information on the ordering between certain values. The Qin–Heron formula derived as Example 3.19 is a typical example for this. For all these examples, our method supplies by means of the actual elimination result, which is an arbitrary boolean combination of equations and inequalities, exactly the necessary and sufficient conditions for the theorem to hold wrt. the classical non-degeneracy conditions obtained independently. This extends the application range of our method from pure theorem proving to the much more sophisticated idea of *theorem finding*.

The present limitation of the elimination method to quadratic variables was not a problem at all for the large number of examples discussed here. Recall anyway that this limitation can be pushed up to higher degrees.

Chapter 4

Computational Geometry

After successfully using generic quantifier elimination for verifying geometric facts in the previous chapter, we now turn to computational aspects of geometry. This chapter is concerned with a number of computational problems in real 3-space or the real plane that arise from applications mainly in the area of CAD, computer vision, and motion planning. The problems are of the following types:

Parallel projection Given an object A and a parametric light ray r, compute the parallel projection of A in direction r onto a plane perpendicular to r. Compute the shaded and lighted parts of A. Compute the shadow cast by A onto some other object B. Reconstruct object A, or determine features of A from its image under parallel projection.

Central projection We treat the same problems as with parallel projection, but wrt. central projection arising from a punctual light source.

Offsets Given a closed object A and a positive real r. Compute the r-offset of A: This is the set of points having distance exactly r from A. Here distance may be Euclidean distance or distance in some other norm.

Voronoi diagrams Given finitely many pairwise disjoint closed objects A_1, ..., A_n, compute their Voronoi diagram: This is the set of all points that have equal minimal distance to at least two different objects A_i and A_j. Again distances may be measured according to various norms.

Collision problems Given two moving objects A and B, a starting position and velocity vector for A, and a parametric starting position and velocity vector for B. Determine the conditions on the parameters under which the objects A and B will eventually collide, and under these conditions the time and place, where the collision will happen.

Among these problems, projections have obvious applications in the area of CAD and computer vision. Offsets are important special cases of Minkowski sums required for robot motion planning [BKOS97]. There has also been considerable research on using offset computations for rounding and blending of solids in CSG-based CAD systems [Ros85, RR86, SW97]. The next chapter will focus on this issue. Voronoi diagrams have numerous applications in various fields such as social geography, physics, astronomy, and again robotics [BKOS97]. The relevance of collision problems, e.g. for robot motion planning, is again obvious.

We will show that all these problem types can be modeled as regular or extended quantifier elimination problems in real algebra. Roughly speaking, extended quantifier elimination differs from ordinary quantifier elimination by the fact that the procedure provides—besides a quantifier-free equivalent to the input formula—sample parametric answers for values of the variables in the outermost existential quantifier block. See Section 2.4.1 for a detailed discussion on extended quantifier elimination by virtual substitution as it is implemented in REDLOG. We will again use almost exclusively our generic variants of regular and extended quantifier elimination. Since the topics treated here are, in contrast to most geometric theorems, not inherently subject to non-degeneracy assumptions, it will be interesting to observe and analyze the role of the assumptions made by the generic quantifier elimination for these problems. From now on, we will simply call these assumptions ND-*conditions*, which is derived from the phrase *non-degeneracy condition* but should not be understood literally.

Many of the examples computed below involve a significant number of parameters besides the variables to be eliminated. Examples of this kind have been inaccessible to other implemented real elimination methods so far.

In the following section, we discuss the restrictions imposed by modeling geometric objects as semialgebraic sets. The remaining sections of this chapter are then devoted to the various types of geometric problems mentioned above. For each problem class we describe its modeling by first-order formulas, and discuss specific instances in this problem class, and their automatic solution using the REDLOG package. All computations have been performed on a SUN Ultra 1 Model 140 workstation using 32 MB of memory.

4.1 Geometric Objects as Semialgebraic Sets

Throughout this chapter, we will assume that all inputs to our computations are objects in the real plane or in real 3-space given by a "low-degree" semi-algebraic description, i.e. a boolean combination of polynomial equations and inequalities in two or three variables, respectively. These equations and inequalities will be quadratic in most cases. Frequently we will allow unspecified real parameters in such a description; then we obtain a solution for a whole parametric family of problems. Our methods do not impose any restrictions on the degree of parameters. Similarly, the computed output objects will also be given in such a description, where, however, the polynomial degrees may and will exceed 2 as a rule.

We consider semi-algebraic objects A, B, ... in real 3-space $S = \mathbb{R}^3$ given by corresponding defining quantifier-free formulas $\alpha(u_1, u_2, u_3)$, $\beta(u_1, u_2, u_3)$, ... We will allow ourselves to identify formulas with the corresponding semialgebraic sets of points when discussing the various operations.

4.2 Parallel Projection

Let $\mathbf{r} = (r_1, r_2, r_3)$ be a non-zero vector in S. Interpreting $\mathbf{r}$ as the direction of light coming in parallel rays from an infinitely far light source, we can model various concepts of image and shadow: A point $(u_1, u_2, u_3) \in S$ is the image of a point $(x_1, x_2, x_3) \in S$ under parallel projection along $\mathbf{r}$ if and only if

$$\exists t\Big(t > 0 \wedge \bigwedge_{i=1}^{3} u_i = x_i + tr_i\Big).$$

Hence the region $\mathrm{SR}(\alpha, \mathbf{r})$ in space S that is shaded by object $\alpha(u_1, u_2, u_3)$ can be described by the following formula:

$$\mathrm{SR}(\alpha, \mathbf{r})(\mathbf{u}) \equiv \exists x_1 \exists x_2 \exists x_3 \exists t\Big(\alpha[\mathbf{u}/\mathbf{x}] \wedge t > 0 \wedge \bigwedge_{i=1}^{3} u_i = x_i + tr_i\Big).$$

By determining a quantifier-free equivalent $\mathrm{SR}(\alpha, \mathbf{r})'(\mathbf{u})$ to the formula $\mathrm{SR}(\alpha, \mathbf{r})$ above, we get a semi-algebraic description of the corresponding region.

The *boundary* of $\alpha(\mathbf{u})$ can be described by a quantifier-free formula $\partial(\alpha)'$ equivalent to

$$\partial(\alpha)(\mathbf{u}) \equiv \alpha \wedge \forall \varepsilon\Big(\varepsilon > 0 \longrightarrow \exists x_1 \exists x_2 \exists x_3 \Big(\neg\alpha[\mathbf{u}/\mathbf{x}] \wedge \bigwedge_{i=1}^{3} (-\varepsilon < x_i - u_i < \varepsilon)\Big)\Big).$$

This description of the boundary together with that of the shaded region above can be used to describe the shaded part $\mathrm{Sh}(\alpha, \mathbf{r})$ and the lighted part $\mathrm{Li}(\alpha, \mathbf{r})$ of the boundary of an object α:

$$\mathrm{Sh}(\alpha, \mathbf{r})(\mathbf{u}) \equiv \partial(\alpha) \wedge \mathrm{SR}(\alpha, \mathbf{r}), \quad \mathrm{Li}(\alpha, \mathbf{r})(\mathbf{u}) \equiv \partial(\alpha) \wedge \neg\mathrm{SR}(\alpha, \mathbf{r}).$$

For strictly convex objects $\gamma(\mathbf{u})$, we can describe the boundary $\mathrm{SL}(\gamma, \mathbf{r})$ between $\mathrm{Sh}(\gamma, \mathbf{r})$ and $\mathrm{Li}(\gamma, \mathbf{r})$ as follows:

$$\mathrm{SL}(\gamma, \mathbf{r})(\mathbf{u}) \equiv \partial(\gamma) \wedge \neg\exists x_1 \exists x_2 \exists x_3 \exists t\Big(\gamma[\mathbf{u}/\mathbf{x}] \wedge t \neq 0 \wedge \bigwedge_{i=1}^{3} u_i = x_i + tr_i\Big).$$

This definition works even for convex but not strictly convex objects provided that the direction $\mathbf{r}$ of light is not degenerate.

Consider now two objects $\alpha(\mathbf{u})$ and $\beta(\mathbf{u})$. Then the shadow $\mathrm{CS}(\alpha, \beta, \mathbf{r})$ cast by object α onto the boundary of β can be described by the following formula:

$$\mathrm{CS}(\alpha, \beta, \mathbf{r})(\mathbf{u}) \equiv \partial(\beta) \wedge \exists x_1 \exists x_2 \exists x_3 \exists t\Big(\alpha[\mathbf{u}/\mathbf{x}] \wedge t \geqslant 0 \wedge \bigwedge_{i=1}^{3} u_i = x_i + tr_i\Big).$$

This applies in particular to the case when β is a plane perpendicular to the light ray $\mathbf{r}$. In this case the shadow cast onto β by α is identical to the image of α viewed in direction $\mathbf{r}$. This image is called the *aspect* of α. For simplicity, we may take β as a plane through the origin and drop the condition $t \geqslant 0$. Then we get the following formula for the aspect of α in direction $\mathbf{r}$:

$$\mathrm{Asp}(\alpha, \mathbf{r})(\mathbf{u}) \equiv \sum_{i=1}^{3} r_i u_i = 0 \wedge \exists x_1 \exists x_2 \exists x_3 \exists t\Big(\alpha[\mathbf{u}/\mathbf{x}] \wedge \bigwedge_{i=1}^{3} u_i = x_i + tr_i\Big).$$

All the formulations discussed so far can obviously also be applied to parametric objects $\alpha(u_1, u_2, u_3, a_1, \ldots, a_n)$. The parameters $a_1, \ldots, a_n$ will then in general also occur in the quantifier-free descriptions resulting from the operations. The same holds for parametric directions of light.

Example 4.1 (Parametric quadric) For our first set of examples, we consider a parametric quadric in space of the form

$$\eta(\mathbf{u},\mathbf{a}) \equiv a_1u_1^2 + a_2u_2^2 + a_3u_3^2 \leqslant 1$$

and a parametric light ray $\mathbf{r} = (r_1, r_2, r_3)$. Depending on the signs of the parameters a_1, a_2, a_3, the quadric η is an ellipsoid, a cylinder, or a hyperboloid.

$$\begin{aligned} \mathrm{SR}(\eta,\mathbf{r})(\mathbf{u},\mathbf{a},\mathbf{r}) \quad \equiv \quad & \exists x_1 \exists x_2 \exists x_3 \exists t (t > 0 \wedge a_1x_1^2 + a_2x_2^2 + a_3x_3^2 \leqslant 1 \wedge \\ & u_1 = x_1 + tr_1 \wedge u_2 = x_2 + tr_2 \wedge u_3 = x_3 + tr_3). \end{aligned}$$

We obtain after 442 ms a quantifier-free equivalent $\mathrm{SR}(\eta,\mathbf{r})'$ containing 6 atomic formulas, and the ND-condition $ar_1^2 + br_2^2 + cr_3^2 \neq 0$.

$$\begin{aligned} \mathrm{Sh}(\eta,\mathbf{r})(\mathbf{u},\mathbf{a},\mathbf{r}) \quad \equiv \quad & a_1u_1^2 + a_2u_2^2 + a_3u_3^2 = 1 \wedge \\ & \exists x_1 \exists x_2 \exists x_3 \exists t (t > 0 \wedge a_1x_1^2 + a_2x_2^2 + a_3x_3^2 \leqslant 1 \wedge \\ & u_1 = x_1 + tr_1 \wedge u_2 = x_2 + tr_2 \wedge u_3 = x_3 + tr_3). \end{aligned}$$

We obtain after 629 ms a quantifier-free description $\mathrm{Sh}(\eta,\mathbf{r})'$ containing 9 atomic formulas, valid under the ND-condition $ar_1^2 + br_2^2 + cr_3^2 \neq 0$.

$$\begin{aligned} \mathrm{Li}(\eta,\mathbf{r})(\mathbf{u},\mathbf{a},\mathbf{r}) \quad \equiv \quad & a_1u_1^2 + a_2u_2^2 + a_3u_3^2 = 1 \wedge \\ & \neg\exists x_1 \exists x_2 \exists x_3 \exists t (t > 0 \wedge a_1x_1^2 + a_2x_2^2 + a_3x_3^2 \leqslant 1 \wedge \\ & u_1 = x_1 + tr_1 \wedge u_2 = x_2 + tr_2 \wedge u_3 = x_3 + tr_3). \end{aligned}$$

We obtain after 561 ms a quantifier-free description $\mathrm{Li}(\eta,\mathbf{r})'$ in 5 atomic formulas valid under the ND-condition $ar_1^2 + br_2^2 + cr_3^2 \neq 0$.

$$\begin{aligned} \mathrm{SL}(\eta,\mathbf{r})(\mathbf{u},\mathbf{a},\mathbf{r}) \quad \equiv \quad & a_1u_1^2 + a_2u_2^2 + a_3u_3^2 = 1 \wedge \\ & \neg\exists x_1 \exists x_2 \exists x_3 \exists t (t \neq 0 \wedge a_1x_1^2 + a_2x_2^2 + a_3x_3^2 \leqslant 1 \wedge \\ & u_1 = x_1 + tr_1 \wedge u_2 = x_2 + tr_2 \wedge u_3 = x_3 + tr_3). \end{aligned}$$

We obtain within 476 ms the quantifier-free equivalent

$$\mathrm{SL}(\eta,\mathbf{r})' \equiv \sum_{i=1}^{3} a_ir_i^2 \geqslant 0 \wedge \sum_{i=1}^{3} a_ir_iu_i = 0 \wedge \sum_{i=1}^{3}(a_iu_i^2) - 1 = 0$$

valid under the simultaneously obtained ND-condition $a_1r_1^2 + a_2r_2^2 + a_3r_3^2 \neq 0$.

$$\begin{aligned} \mathrm{Asp}(\eta,\mathbf{r})(\mathbf{u},\mathbf{a},\mathbf{r}) \quad \equiv \quad & \exists x_1 \exists x_2 \exists x_3 \exists t (a_1x_1^2 + a_2x_2^2 + a_3x_3^2 \leqslant 1 \wedge \\ & u_1 = x_1 + tr_1 \wedge u_2 = x_2 + tr_2 \wedge u_3 = x_3 + tr_3 \wedge \\ & r_1u_1 + r_2u_2 + r_3u_3 = 0). \end{aligned}$$

We obtain after 442 ms a quantifier-free equivalent with 6 atomic formulas valid under the ND-condition $a_1r_1^2 + a_2r_2^2 + a_3r_3^2 \neq 0$. □

Example 4.2 (Aspect of intersection) Consider two parametric tubes that cross perpendicularly but not necessarily centrally:

$$\begin{aligned}\tau_1(\mathbf{u},\mathbf{a},d_1) &\equiv (u_1-a_1)^2+(u_2-a_2)^2=d_1^2\\ \tau_2(\mathbf{u},d_2) &\equiv u_1^2+u_3^2=d_2^2.\end{aligned}$$

We compute the aspect of their intersection line $\iota=\tau_1\wedge\tau_2$ from a fully parametric direction $\mathbf{r}=(r_1,r_2,r_3)$:

$$\begin{aligned}\mathrm{Asp}(\iota,\mathbf{r})(\mathbf{u},\mathbf{a},d_1,d_2,\mathbf{r}) \equiv\ & \exists x_1\exists x_2\exists x_3\exists t\big(\iota[\mathbf{u}/\mathbf{x}]\wedge r_1u_1+r_2u_2+r_3u_3=0\ \wedge\\ & u_1=x_1+tr_1\wedge u_2=x_2+tr_2\wedge u_3=x_3+tr_3\big).\end{aligned}$$

The result obtained after 357 ms is the following quantifier-free description valid under the ND-condition $r_3\neq 0\wedge u_3\neq 0$:

$$\begin{aligned}a_1^2r_3^2u_3^2+2a_1r_1^2r_3u_1u_3+2a_1r_1r_2r_3u_2u_3+2a_1r_1r_3u_3^3\ -\\ 2a_1r_3^2u_1u_3^2+a_2^2r_3^2u_3^2+2a_2r_1r_2r_3u_1u_3+2a_2r_2^2r_3u_2u_3+2a_2r_2r_3u_3^3\ -\\ 2a_2r_3^2u_2u_3^2-d_1^2r_3^2u_3^2+r_1^4u_1^2+2r_1^3r_2u_1u_2+2r_1^3u_1u_3^2+r_1^2r_2^2u_1^2\ +\\ r_1^2r_2^2u_2^2+2r_1^2r_2u_2u_3^2-2r_1^2r_3u_1^2u_3+r_1^2u_3^4+2r_1r_2^3u_1u_2+2r_1r_2^2u_1u_3^2\ -\\ 4r_1r_2r_3u_1u_2u_3-2r_1r_3u_1u_3^3+r_2^4u_2^2+2r_2^3u_2u_3^2\ -\\ 2r_2^2r_3u_2^2u_3+r_2^2u_3^4-2r_2r_3u_2u_3^3+r_3^2u_1^2u_3^2+r_3^2u_2^2u_3^2 &= 0\\ \wedge\ d_2^2r_3^2u_3^2-r_1^4u_1^2-2r_1^3r_2u_1u_2-2r_1^3u_1u_3^2-r_1^2r_2^2u_2^2-2r_1^2r_2u_2u_3^2\ -\\ r_1^2r_3^2u_1^2+2r_1^2r_3u_1^2u_3-r_1^2u_3^4-2r_1r_2r_3^2u_1u_2\ +\\ 2r_1r_2r_3u_1u_2u_3+2r_1r_3u_1u_3^3-r_2^2r_3^2u_2^2-r_3^2u_1^2u_3^2 &= 0.\end{aligned}$$

Note that such conjunctions $f_1=0\wedge f_2=0$ can be restated algebraically as $f_1^2+f_2^2=0$. □

We now turn to the inverse problem to computing the aspect of an object: We wish to reconstruct features of an object α and its orientation relative to the projection plane from its aspect. To this end we assume that our object belongs to a parametric class of objects, e.g. cuboids, balls, or ellipsoids, described by a quantifier-free formula $\alpha(u_1,u_2,u_3,a_1,\dots,a_n)$. The real parameters $a_1,\dots,a_n$ describe the dimensions and the orientation of the object.

In general, we can put our coordinate system into such a position that the light ray projects the object along the Z-axis onto the X-Y-plane. Since the image of the object is invariant under translation of the object along the light ray, we may assume in addition that some reference point of the object is placed in the origin of the coordinate system. Suppose now that we have a quantifier-free formula $\alpha_0(u_1,u_2)$ which we assume to describe $\mathrm{Asp}\big(\alpha,(0,0,1)\big)$. Then application of extended quantifier elimination to the formula

$$\mathrm{Rec}(\alpha,\alpha_0)\equiv\exists u_3\exists a_1\dots\exists a_n\forall u_1\forall u_2\big(\alpha(u_1,u_2,u_3,a_1,\dots,a_n)\longleftrightarrow\alpha_0(u_1,u_2)\big)$$

will yield the result "false" if α_0 does not describe the aspect of an object in the given class; otherwise it will yield "true" together with a list of possible real values for u_3, $a_1,\dots,a_n$, where the values for u_3 are of no interest.

Among the problems described in this section, feature reconstruction is by far the hardest one. In order to obtain suitable results within a tolerable amount of time, one has to put more intelligence into the coding of the problem than with the general formulation $\mathrm{Rec}(\alpha, \alpha_0)$ above. To give an impression of possible formulations, we describe how one would try to recover the dimensions of a rectangular solid κ and its orientation in space from its aspect. This approach is applied to a concrete aspect in Example 4.3 below.

Consider the target cuboid κ to be generated by pairwise perpendicular vectors

$$\mathbf{e_1} = (e_{11}, e_{12}, e_{13}), \quad \mathbf{e_2} = (e_{21}, e_{22}, e_{23}), \quad \mathbf{e_3} = (e_{31}, e_{32}, e_{33})$$

with one corner located at the origin. Its corners $\mathbf{x} = (x_1, x_2, x_3)$ are described by the formula

$$\mathbf{e_1e_2} = \mathbf{e_1e_3} = \mathbf{e_2e_3} = 0 \wedge \exists\lambda_1\exists\lambda_2\exists\lambda_3\big(\gamma(\lambda_1, \lambda_2, \lambda_3, \mathbf{x}, \mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3})\big), \quad \text{where}$$

$$\gamma(\lambda_1, \lambda_2, \lambda_3, \mathbf{x}, \mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3}) \equiv \lambda_1, \lambda_2, \lambda_3 \in \{0, 1\} \wedge \mathbf{x} = \lambda_1\mathbf{e_1} + \lambda_2\mathbf{e_2} + \lambda_3\mathbf{e_3}.$$

The direction of the light is along the Z-axis. The projection plane P is the X-Y-plane.

Provided that κ is in generic position, i.e., not parallel to any of the axes, its aspect is the convex hull of 6 points with two possible choices for points which cannot be observed:

1. All e_{i3} coordinates have the same sign. Then the images of the origin and of its opposite are in the interior of the aspect.

2. Up to a permutation of the $\mathbf{e_i}$, the sign of e_{13} differs from that of e_{23} and e_{33}. Then the image of $\mathbf{e_1}$ and that of its opposite $\mathbf{e_2} + \mathbf{e_3}$ are in the interior of the aspect.

The concrete coordinate system is chosen by the observer of the aspect, who will exclude the first of the two cases above by placing one aspect corner into the origin.

We generate by quantifier elimination (255 ms) a quantifier-free generic aspect description ι' from the following formula:

$$\begin{aligned} \iota(\mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3}, i_1, i_2) \quad \equiv \quad & \mathbf{e_1e_2} = \mathbf{e_1e_3} = \mathbf{e_2e_3} = 0 \wedge \\ & \exists x_1\exists x_2\exists x_3\exists\lambda_1\exists\lambda_2\exists\lambda_3\big(\gamma(\lambda_1, \lambda_2, \lambda_3, \mathbf{x}, \mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3}) \wedge \\ & \neg(\lambda_1 \neq \lambda_2 = \lambda_3) \wedge i_1 = x_1 \wedge i_2 = x_2\big). \end{aligned}$$

For a given semi-algebraic description $\kappa_0(i_1, i_2)$ of aspect corners, we can now reconstruct the original cuboid by applying extended quantifier elimination to the following reconstruction formula:

$$\varrho(\kappa_0) \equiv \exists e_{11}\exists e_{12}\exists e_{13}\exists e_{21}\exists e_{22}\exists e_{23}\exists e_{31}\exists e_{32}\exists e_{33}\forall i_1\forall i_2(\iota' \longleftrightarrow \kappa_0).$$

For correct generic aspect descriptions κ_0 this elimination will yield "true" together with suitable vectors $\mathbf{e_1}$, $\mathbf{e_2}$, $\mathbf{e_3}$ as answer. If κ_0 does not describe the aspect of a cuboid in generic position, then the elimination will yield "false."

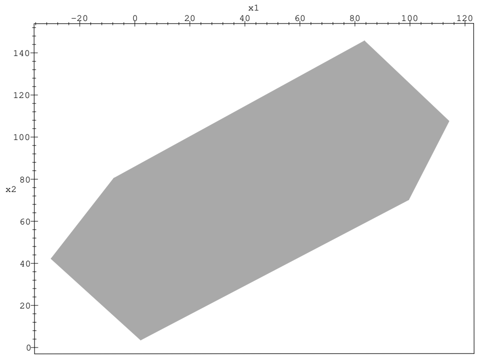

Figure 4.1: A sample aspect of a rectangular solid in generic position

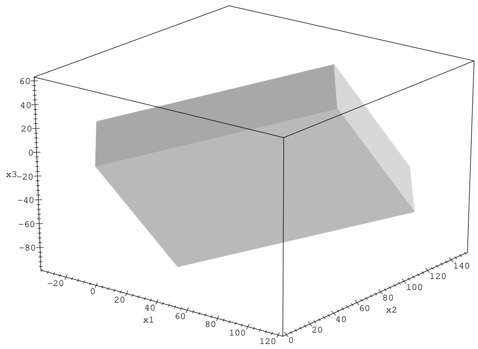

Figure 4.2: Rectangular solid reconstructed from the top aspect given in Figure 4.1

Example 4.3 (Solid reconstruction) Consider the aspect that is given in Figure 4.1. Its corners have the following semi-algebraic description:

$$\begin{aligned}\kappa_0 \quad \equiv \quad & (13i_1 - 1124 = 0 \wedge 13i_2 - 1960 = 0) \vee \\ & (13i_1 + 176 = 0 \wedge 13i_2 - 1050 = 0) \vee \\ & (13i_1 + 436 = 0 \wedge 13i_2 - 530 = 0) \vee (i_1 - 120 = 0 \wedge i_2 - 110 = 0) \vee \\ & (i_1 - 100 = 0 \wedge i_2 - 70 = 0) \vee (i_1 = 0 \wedge i_2 = 0).\end{aligned}$$

Extended quantifier elimination applied to $\varrho(\kappa_0)$ yields after 15.7 s "true" together with the correct reconstruction pictured in Figure 4.2:

$$\mathbf{e_1} = (20, 40, -96), \quad \mathbf{e_2} = \left(-\frac{436}{13}, \frac{530}{13}, 10\right), \quad \mathbf{e_3} = (100, 70, 50). \quad \square$$

4.3 Central Projection

In this section we consider the same problems as in the previous one but for central projection: We again consider, semialgebraic objects described by quantifier-free formulas $\alpha(\mathbf{u})$; their boundaries are described by $\partial(\alpha)(\mathbf{u})$ as defined in the previous section.

Let $\mathbf{c} = (c_1, c_2, c_3) \in S$ denote the position of a punctual light source in space. Then a point $\mathbf{u} = (u_1, u_2, u_3) \in S$ is the image of a point $\mathbf{x} = (x_1, x_2, x_3) \in S$ under central projection from the point $\mathbf{c}$ if and only if

$$\exists t\Big(t > 0 \wedge \bigwedge_{i=1}^{3} u_i = t(x_i - c_i)\Big).$$

Hence the region in space that is shaded by object $\alpha(\mathbf{u})$ under central projection from $\mathbf{c}$ can be described by the following formula:

$$\mathrm{SR}^{\mathrm{c}}(\alpha, \mathbf{c})(\mathbf{u}) \equiv \exists x_1 \exists x_2 \exists x_3 \exists t\Big(\alpha[\mathbf{u}/\mathbf{x}] \wedge t > 0 \wedge \bigwedge_{i=1}^{3} u_i = t(x_i - c_i)\Big).$$

In a similar way our descriptions given in the previous section of the shaded part, the lighted part, and of the boundary between these parts can be adapted to central projection. For the latter we again have to exclude degenerate situations, where the surface of the object contains more than one point on some light ray:

$$\mathrm{Sh}^{\mathrm{c}}(\alpha, \mathbf{c})(\mathbf{u}) \equiv \partial(\alpha) \wedge \mathrm{SR}^{\mathrm{c}}(\alpha, \mathbf{c}), \quad \mathrm{Li}^{\mathrm{c}}(\alpha, \mathbf{c})(\mathbf{u}) \equiv \partial(\alpha) \wedge \neg\mathrm{SR}^{\mathrm{c}}(\alpha, \mathbf{c}),$$

$$\mathrm{SL}^{\mathrm{c}}(\alpha, \mathbf{c})(\mathbf{u}) \equiv \partial(\alpha) \wedge \neg\exists x_1 \exists x_2 \exists x_3 \exists t\Big(\alpha[\mathbf{u}/\mathbf{x}] \wedge t \neq 0 \wedge \bigwedge_{i=1}^{3} u_i = t(x_i - c_i)\Big).$$

Let $\beta(\mathbf{u})$ be another object in S. Then the shadow cast by object $\alpha(\mathbf{u})$ onto the boundary of β is described by

$$\mathrm{CS}^{\mathrm{c}}(\alpha, \beta, \mathbf{c})(\mathbf{u}) \equiv \partial(\beta) \wedge \exists x_1 \exists x_2 \exists x_3 \exists t\Big(\alpha[\mathbf{u}/\mathbf{x}] \wedge t \geqslant 0 \wedge \bigwedge_{i=1}^{3} u_i = t(x_i - c_i)\Big).$$

This applies in particular to the case when the second object is a plane $\pi(\mathbf{u}) \equiv p_1u_1 + p_2u_2 + p_3u_3 + q = 0$. In this case the shadow cast onto π by α is called the *image* of α on π under central projection from $\mathbf{c}$:

$$\mathrm{Im}(\alpha, \pi, \mathbf{c})(\mathbf{u}) \equiv \pi \wedge \exists x_1 \exists x_2 \exists x_3 \exists t\Big(\alpha[\mathbf{u}/\mathbf{x}] \wedge t \geqslant 0 \wedge \bigwedge_{i=1}^{3} u_i = t(x_i - c_i)\Big).$$

Recall from the previous section that it is the great advantage of our approach that all the formulations may involve parametric objects. The parameters will occur in the quantifier-free final results just as they do in the input.

Example 4.4 (Parametric line) This example is inspired by an example that has been discussed by Kapur and Mundy [KM88]. We consider the central projection of the parametric line

$$\lambda(\mathbf{u}, \mathbf{a}, \mathbf{b}, d_1, d_2) \equiv a_1u_1 + a_2u_2 + a_3u_3 = d_1 \wedge b_1u_1 + b_2u_2 + b_3u_3 = d_2$$

from a punctual light source located at $\mathbf{c} = (0, 0, f)$ onto the projection plane $\pi(\mathbf{u}) \equiv u_3 = 0$:

$$\begin{aligned}\mathrm{Im}(\lambda, \pi, \mathbf{c})(\mathbf{u}, \mathbf{a}, \mathbf{b}, d_1, d_2, f) \quad \equiv \quad & \exists x_1 \exists x_2 \exists x_3 \big(a_1 x_1 + a_2 x_2 + a_3 x_3 = d_1 \wedge \\ & b_1 x_1 + b_2 x_2 + b_3 x_3 = d_2 \wedge \\ & u_1(f - x_3) = f x_1 \wedge u_2(f - x_3) = f x_2\big).\end{aligned}$$

Generic quantifier elimination with the input theory $\vartheta \equiv f > 0$ yields after 221 ms the quantifier-free equivalent

$$\begin{aligned}\mathrm{Im}(\lambda, \pi, \mathbf{c})' \quad \equiv \quad & a_1 b_3 f u_1 - a_1 d_2 u_1 - b_1 a_3 f u_1 + b_1 d_1 u_1 + a_2 b_3 f u_2 - \\ & a_2 d_2 u_2 - b_2 a_3 f u_2 + b_2 d_1 u_2 + a_3 d_2 f - b_3 d_1 f = 0\end{aligned}$$

subject to the extended theory $\vartheta' \equiv \vartheta \wedge a_1 u_1 + a_2 u_2 - a_3 f \neq 0$, i.e., there is the additional ND-condition $a_1 u_1 + a_2 u_2 - a_3 f \neq 0$ automatically derived during the elimination.

Consider another parametric line λ' in space:

$$\lambda'(\mathbf{u}, \mathbf{a}', \mathbf{b}', d_1', d_2') \equiv a_1' u_1 + a_2' u_2 + a_3' u_3 = d_1' \wedge b_1' u_1 + b_2' u_2 + b_3' u_3 = d_2'.$$

We can read off from the elimination result above the necessary and sufficient condition for the images of the two lines to be parallel under the corresponding ND-conditions, viz.:

$$\begin{aligned}&(a_1 b_3 f - a_1 d_2 - b_1 a_3 f + b_1 d_1)(a_2' b_3' f - a_2' d_2' - b_2' a_3' f + b_2' d_1') = \\ &\qquad (a_1' b_3' f - a_1' d_2' - b_1' a_3' f + b_1' d_1')(a_2 b_3 f - a_2 d_2 - b_2 a_3 f + b_2 d_1)\end{aligned}$$

provided that all bracketed items are non-zero. □

Example 4.5 (Unit ball) We compute the central projection of a unit sphere located at an arbitrary position $\mathbf{a} = (a_1, a_2, a_3)$:

$$\beta(\mathbf{u}, \mathbf{a}) \equiv \sum_{i=1}^{3} (u_i - a_i)^2 = 1.$$

Again, the light source is located at $\mathbf{c} = (0, 0, f)$, and the projection plane is given by $u_3 = 0$.

$$\begin{aligned}\mathrm{Im}(\beta, \mathbf{c})(\mathbf{u}, \mathbf{a}, f) \quad \equiv \quad & \exists x_1 \exists x_2 \exists x_3 \big((u_1 - a_1)^2 + (u_2 - a_2)^2 + (u_3 - a_3)^2 = 1 \wedge \\ & u_1(f - x_3) = f x_1 \wedge u_2(f - x_3) = f x_2\big).\end{aligned}$$

Generic quantifier elimination with the input theory $\vartheta \equiv f > 0$ yields after 187 ms:

$$\begin{aligned}&a_1^2 f^2 + a_1^2 u_2^2 - 2a_1 a_2 u_1 u_2 + 2a_1 a_3 f u_1 - 2a_1 f^2 u_1 + a_2^2 f^2 + a_2^2 u_1^2 + \\ &\quad 2a_2 a_3 f u_2 - 2a_2 f^2 u_2 + a_3^2 u_1^2 + a_3^2 u_2^2 - 2a_3 f u_1^2 - 2a_3 f u_2^2 + f^2 u_1^2 + \\ &\qquad\qquad f^2 u_2^2 - f^2 - u_1^2 - u_2^2 \quad \leqslant \quad 0\end{aligned}$$

without any additional ND-condition. □

In the previous section on parallel projections, we have recovered a rectangular solid from its aspect. We now turn to a more difficult problem, where we wish to recover such a cuboid from a central projection. To make the problem easier, we consider a wire frame instead of a solid avoiding visibility considerations. Still the cuboid will not be uniquely determined by one wire frame image.

We modify our aspect case model as follows: The cuboid is generated by the vectors $\mathbf{e_1}$, $\mathbf{e_2}$, $\mathbf{e_3}$ and shifted from the origin by some translation vector $\mathbf{v} = (v_1, v_2, v_3)$. The projection plane is the x_1-x_2 plane. The focal point is, say, $(0, 0, 5)$. Our idea is that the focal point lies between the image and the projection plane modeling photography: We take photos along the Z-axis. The 3-space origin is the center of our photo.

A generic image description ι' is derived by applying quantifier elimination (765 ms) to the following formula:

$$\begin{aligned}\iota \quad \equiv \quad & \mathbf{e_1 e_2} = \mathbf{e_1 e_3} = \mathbf{e_2 e_3} = 0 \land \\ & \exists x_1 \exists x_2 \exists x_3 \exists \lambda_1 \exists \lambda_2 \exists \lambda_3 \exists t \big(\lambda_1, \lambda_2, \lambda_3 \in \{0, 1\} \land \\ & \mathbf{x} = \lambda_1 \mathbf{e_1} + \lambda_2 \mathbf{e_2} + \lambda_3 \mathbf{e_3} + \mathbf{v} \land i_1 = t x_1 \land i_2 = t x_2 \land 0 = 5 + t(5 - x_3)\big).\end{aligned}$$

For a given semi-algebraic image description ω_0, we can recover information about the original wire frame by applying extended quantifier elimination to the following reconstruction formula:

$$\varrho^{\mathrm{c}}(\omega_0) \equiv \exists e_{11} \exists e_{12} \exists e_{13} \exists e_{21} \exists e_{22} \exists e_{23} \exists e_{31} \exists e_{32} \exists e_{33} \exists v_1 \exists v_2 \exists v_3 \forall i_1 \forall i_2 (\iota' \longleftrightarrow \omega_0).$$

In practice, the elimination can be supported by fixing the image of the shifted origin $\mathbf{v}$. Compare Example 4.6 below for details.

With two images taken simultaneously from different locations, a cuboid can be reconstructed uniquely provided that the choices for the $\mathbf{v}$, $\mathbf{e_1}$, $\mathbf{e_2}$, $\mathbf{e_3}$ are consistent. Unfortunately, in a corresponding reconstruction formula for a stereo camera one cannot play the trick of fixing the images of $\mathbf{v}$ as indicated above, because one cannot automatically determine which of the points in two different projection images correspond to each other.

Example 4.6 (Wire frame reconstruction) Figure 4.3 shows an image corresponding to a photo taken of the wire frame of the cuboid in Figure 4.2 shifted from the origin by $\mathbf{v} = (100, 200, 300)$. The semi-algebraic image description is as follows:

$$\begin{aligned}\omega_0(i_1, i_2) \quad \equiv \quad & (3367 i_1 - 12120 = 0 \land 3367 i_2 - 22800 = 0) \lor \\ & (923 i_1 - 2164 = 0 \land 923 i_2 - 4040 = 0) \lor \\ & (2717 i_1 - 5620 = 0 \land 2717 i_2 - 18250 = 0) \lor \\ & (793 i_1 - 864 = 0 \land 793 i_2 - 3130 = 0) \lor \\ & (249 i_1 - 1100 = 0 \land 249 i_2 - 1550 = 0) \lor \\ & (69 i_1 - 200 = 0 \land 23 i_2 - 90 = 0) \lor \\ & (199 i_1 - 600 = 0 \land 199 i_2 - 1200 = 0) \lor \\ & (59 i_1 - 100 = 0 \land 59 i_2 - 200 = 0).\end{aligned}$$

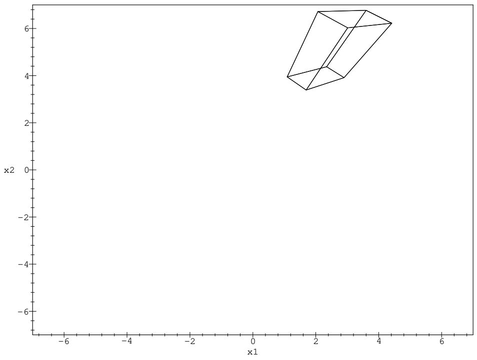

Figure 4.3: A photo of the wire frame of the cuboid in Figure 4.2

In the reconstruction formula we support the elimination procedure by fixing the image point $\left(\frac{100}{59}, \frac{200}{59}, 0\right)$ to be the image the shifted origin:

$$\begin{aligned}\overline{\varrho^c} \quad \equiv \quad & \exists e_{11}\exists e_{12}\exists e_{13}\exists e_{21}\exists e_{22}\exists e_{23}\exists e_{31}\exists e_{32}\exists e_{33}\exists v_1\exists v_2\exists v_3\forall i_1\forall i_2\Big((\iota' \longleftrightarrow \omega_0)\wedge \\ & \exists t\Big(\frac{100}{59} = tv_1 \wedge \frac{200}{59} = tv_2 \wedge 0 = 5 + t(5 - v_3)\Big)\Big).\end{aligned}$$

We obtain after 33 min the quantifier-free equivalent "true" together with the following answer:

$$\mathbf{v} = \Big(\frac{5\infty_1}{2}, 5\infty_1, \frac{59\infty_1 + 40}{8}\Big), \quad \mathbf{e_1} = \Big(\frac{5\infty_1}{2}, \frac{7\infty_1}{4}, \frac{5\infty_1}{4}\Big),$$

$$\mathbf{e_2} = \Big(\frac{\infty_1}{2}, \infty_1, \frac{-12\infty_1}{5}\Big), \quad \mathbf{e_3} = \Big(\frac{-109\infty_1}{130}, \frac{53\infty_1}{52}, \frac{\infty_1}{4}\Big).$$

For $\infty_1 = 40$ this is our original cuboid.

Without fixing the image of $\mathbf{v}$, i.e., using $\varrho^c(\omega_0)$ as reconstruction formula, the elimination takes 12 h yielding a correct reconstruction with different choices for the vectors. □

4.4 Offsets

Let A be an object that constitutes a non-empty closed set. Fix a metric $d : S^2 \to \mathbb{R}$ that induces the usual topology on S. Then the distance between a point $\mathbf{x} \in S$ and A is defined as

$$d(\mathbf{x}, A) = \min\{\, d(\mathbf{x}, \mathbf{u}) \mid \mathbf{u} \in A \,\}.$$

The existence of the minimum is guaranteed by the assumption that α is closed in S and that d is continuous. Let now r be a non-negative real number. Then the r-*offset* of A is defined as follows:

$$\mathrm{off_r}(A) = \{\, \mathbf{u} \in S \mid d\big(\mathbf{u}, A\big) = r \,\}.$$

We generalize this notion of an r-offset: The $\leqslant$-r-*offset* A and the $\geqslant$-r-*offset* of A are defined as

$$\mathrm{off}_{\leqslant \mathrm{r}}(A) = \{\, \mathbf{u} \in S \mid d\big(\mathbf{u}, A\big) \leqslant r \,\} \quad \text{and} \quad \mathrm{off}_{\geqslant \mathrm{r}}(A) = \{\, \mathbf{u} \in S \mid d\big(\mathbf{u}, A\big) \geqslant r \,\},$$

respectively. We then obviously have $\mathrm{off}_{\mathrm{r}}(A) = \mathrm{off}_{\leqslant \mathrm{r}}(A) \cap \mathrm{off}_{\geqslant \mathrm{r}}(A)$.

For a fixed metric d, suppose we have formulas $\delta_{\leqslant}(\mathbf{u}, \mathbf{x}, r)$ and $\delta_{\geqslant}(\mathbf{u}, \mathbf{x}, r)$ describing the facts that $d(\mathbf{u}, \mathbf{x}) \leqslant r$ and $d(\mathbf{u}, \mathbf{x}) \geqslant r$, respectively. Then the following formulas describe various offsets of a semialgebraic object $\alpha(\mathbf{u})$:

$$\begin{aligned}
\mathrm{off}_{\leqslant \mathrm{r}}(\alpha)(\mathbf{u}) &\equiv \exists x_1 \exists x_2 \exists x_3 \big(\alpha[\mathbf{u}/\mathbf{x}] \wedge \delta_{\leqslant}(\mathbf{x}, \mathbf{u}, r)\big), \\
\mathrm{off}_{\geqslant \mathrm{r}}(\alpha)(\mathbf{u}) &\equiv \forall x_1 \forall x_2 \forall x_3 \big(\alpha[\mathbf{u}/\mathbf{x}] \longrightarrow \delta_{\geqslant}(\mathbf{x}, \mathbf{u}, r)\big), \\
\mathrm{off}_{\mathrm{r}}(\alpha)(\mathbf{u}) &\equiv \mathrm{off}_{\leqslant \mathrm{r}}(\alpha)(\mathbf{u}) \wedge \mathrm{off}_{\geqslant \mathrm{r}}(\alpha)(\mathbf{u}).
\end{aligned}$$

We give the relevant formulas $\delta_{\geqslant}$ for some common metrics on S. The formulas $\delta_{\leqslant}$ are formed similarly:

- for the Euclidean (L_2) metric:

$$\delta_{\geqslant}(\mathbf{x}, \mathbf{u}, r) \equiv \sum_{i=1}^{3} (x_i - u_i)^2 \geqslant r^2;$$

- for the "Manhattan" (L_1) metric:

$$\begin{aligned}
\delta_{\geqslant}(\mathbf{x}, \mathbf{u}, r) \equiv \; & \exists d_1 \exists d_2 \exists d_3 \Big(\sum_{i=1}^{3} d_i \geqslant r \wedge \\
& \bigwedge_{i=1}^{3} \big(d_i \geqslant 0 \wedge (d_i = x_i - u_i \vee d_i = u_i - x_i)\big)\Big);
\end{aligned}$$

- for the maximum (L_∞) metric:

$$\delta_{\geqslant}(\mathbf{x}, \mathbf{u}, r) \equiv \bigvee_{i=1}^{3} \big((x_i - u_i) \geqslant r \vee (u_i - x_i) \geqslant r\big).$$

Note that the formulas $\mathrm{off}_{\mathrm{r}}(\alpha)$ for the r-offset involve 3 existential and 3 universal quantifiers and hence pose a non-trivial problem for quantifier elimination.

Example 4.7 We compute the parametric r-offsets of the cross $\gamma(\mathbf{u}) \equiv (u_1 = 0 \vee u_2 = 0) \wedge u_3 = 0$ consisting of the X-axis and the Y-axis in S. We start with the Euclidean metric:

$$\begin{aligned}
\mathrm{off}_r^{\mathrm{L}_2}(\gamma)(\mathbf{u}, r) \equiv \; & \exists x_1 \exists x_2 \exists x_3 \big((x_1 = 0 \vee x_2 = 0) \wedge x_3 = 0 \wedge \\
& (u_1 - x_1)^2 + (u_2 - x_2)^2 + (u_3 - x_3)^2 \leqslant r^2\big) \wedge \\
& \forall x_1 \forall x_2 \forall x_3 \big((x_1 = 0 \vee x_2 = 0) \wedge x_3 = 0 \longrightarrow \\
& (u_1 - x_1)^2 + (u_2 - x_2)^2 + (u_3 - x_3)^2 \geqslant r^2\big),
\end{aligned}$$

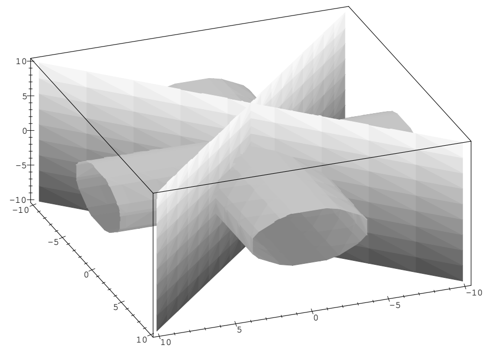

Figure 4.4: 4-Offset of the X-axis and the Y-axis together with the planes $X = Y$ and $X = -Y$ separating the relevant segments of the tubes

Quantifier elimination with subsequent Gröbner basis simplification applied to $\mathrm{off}_r^{\mathrm{L}_2}(\gamma)$ yields after 1.2 s the following quantifier-free description:

$$\mathrm{off}_r^{\mathrm{L}_2}(\gamma)'(\mathbf{u}) \equiv (r^2-u_2^2-u_3^2 = 0 \wedge u_1^2-u_2^2 \geqslant 0) \vee (r^2-u_1^2-u_3^2 = 0 \wedge u_1^2-u_2^2 \leqslant 0).$$

The offset thus consists of segments of the tubes $r^2 - u_2^2 - u_3^2 = 0$ and $r^2 - u_1^2 - u_3^2 = 0$ separated by the planes $u_1 = u_2$ and $u_1 = -u_2$ at which they exactly fit together. Figure 4.4 pictures this for $r = 4$.

For the Manhattan metric, the quantifier elimination input looks as follows:

$$\begin{aligned}
\mathrm{off}_r^{\mathrm{L}_1}(\gamma)(\mathbf{u}, r) \quad \equiv \quad & \exists x_1 \exists x_2 \exists x_3 \big((x_1 = 0 \vee x_2 = 0) \wedge x_3 = 0 \wedge \\
& \exists d_1 \exists d_2 \exists d_3 (d_1 \geqslant 0 \wedge (d_1 = u_1 - x_1 \vee d_1 = x_1 - u_1) \wedge \\
& d_2 \geqslant 0 \wedge (d_2 = u_2 - x_2 \vee d_2 = x_2 - u_2) \wedge d_3 \geqslant 0 \wedge \\
& (d_3 = u_3 - x_3 \vee d_3 = x_3 - u_3) \wedge d_1 + d_2 + d_3 \leqslant r)\big) \wedge \\
& \forall x_1 \forall x_2 \forall x_3 \big((x_1 = 0 \vee x_2 = 0) \wedge x_3 = 0 \longrightarrow \\
& \exists d_1 \exists d_2 \exists d_3 (d_1 \geqslant 0 \wedge (d_1 = u_1 - x_1 \vee d_1 = x_1 - u_2) \wedge \\
& d_2 \geqslant 0 \wedge (d_2 = u_2 - x_2 \vee d_2 = x_2 - u_2) \wedge d_3 \geqslant 0 \wedge \\
& (d_3 = u_3 - x_3 \vee d_3 = x_3 - u_3) \wedge d_1 + d_2 + d_3 \geqslant r)\big).
\end{aligned}$$

Here we obtain after 11.6 s a quantifier-free formula with 4242 atomic subformulas, which cannot be further simplified automatically. □

For certain smooth algebraic surfaces $V : f(\mathbf{u}) = 0$ without singularities in S we would like to compute $\mathrm{off}_\mathrm{r}(V)$ using the following idea: The point $\mathbf{u} \in V$ of minimal distance to a given point $\mathbf{x} \in S$ is determined by the fact that the vector $\mathbf{x} - \mathbf{u}$ is perpendicular to the tangent plane of V at $\mathbf{u}$. In other words, this vector is a real multiple of the gradient vector ∇f. Accordingly, we define $\mathrm{off}_\mathrm{r}{}^\nabla(\nu)$ for the semialgebraic description $\nu(\mathbf{u}) \equiv f(\mathbf{u}) = 0$ of V as follows:

$$\mathrm{off}_r^\nabla(\nu)(\mathbf{u}) \equiv \exists x_1 \exists x_2 \exists x_3 \exists t \Big(f[\mathbf{u}/\mathbf{x}] = 0 \wedge \delta(\mathbf{x}, \mathbf{u}, r) \wedge \bigwedge_{i=1}^{3} t \frac{\partial f[\mathbf{u}/\mathbf{x}]}{\partial x_i} = u_i - x_i\Big).$$

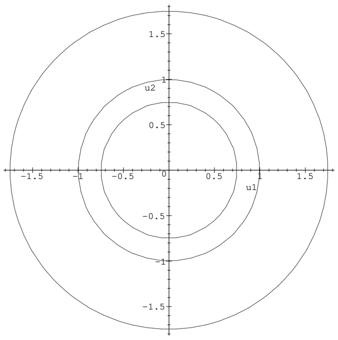

Figure 4.5: The unit circle and its $\frac{3}{4}$-∇-offset

Here $\delta(\mathbf{x}, \mathbf{u}, r)$ describes the fact that $d(\mathbf{x}, \mathbf{u}) = r$.

Unfortunately, $\mathrm{off_r}^\nabla$ does not necessarily equal the offset. Instead, we do only know that

$$\mathrm{off_r}(V) \subseteq \mathrm{off_r}^\nabla(V).$$

The reason is that points that lie very far away from the starting point of the gradient vector on the surface can be closer than r to the endpoint of the gradient vector. Figure 4.5 illustrates the problem by means of the unit circle $u_1^2 + u_2^2 = 1$ in 2-space: Its $\frac{3}{4}$-∇-offset consists of the circles $u_1^2 + u_2^2 = \left(\frac{3}{4}\right)^2$ and $u_1^2 + u_2^2 = \left(\frac{7}{4}\right)^2$, but the latter circle alone forms the proper $\frac{3}{4}$-offset.

It is, however, easy to see that for arbitrary smooth algebraic surfaces V without singularities we always have

$$\mathrm{off_r}(V) = \mathrm{off_r}^\nabla(V) \cap \mathrm{off_{\geqslant r}}(V).$$

The idea of using the gradient vectors can be extended to smooth algebraic curves $C : f(\mathbf{u}) = 0 \wedge g(\mathbf{u}) = 0$ without singularities in S. We then have the following description of the corresponding superset of the r-offset of $\gamma(\mathbf{u}) \equiv f(\mathbf{u}) = 0 \wedge g(\mathbf{u}) = 0$:

$$\begin{aligned} \mathrm{off}_r^\nabla(\gamma)(\mathbf{u}) \quad \equiv \quad & \exists x_1 \exists x_2 \exists x_3 \exists s \exists t \Big(f[\mathbf{u}/\mathbf{x}] = 0 \wedge g[\mathbf{u}/\mathbf{x}] = 0 \wedge \delta(\mathbf{x}, \mathbf{u}, r) \wedge \\ & \bigwedge_{i=1}^{3} s \frac{\partial f[\mathbf{u}/\mathbf{x}]}{\partial x_i} + t \frac{\partial g[\mathbf{u}/\mathbf{x}]}{\partial x_i} = u_i - x_i \Big). \end{aligned}$$

Example 4.8 We compute, wrt. the Euclidean metric, the parametric r-offset of the parabola $\pi(\mathbf{u}) \equiv u_2 = u_1^2 \wedge u_3 = 0$ in S.

$$\begin{aligned} \mathrm{off}_r^\nabla(\pi)(\mathbf{u}) \quad \equiv \quad & \exists x_1 \exists x_2 \exists x_3 \exists s \exists t \big(x_2 = x_1^2 \wedge x_3 = 0 \wedge \\ & (x_1 - u_1)^2 + (x_2 - u_2)^2 + (x_3 - u_3)^2 = r \wedge \\ & -2 s x_1 = u_1 - x_1 \wedge s = u_2 - x_2 \wedge t = u_3 - x_3 \big). \end{aligned}$$

Generic quantifier elimination cannot eliminate x_1 due to the degree restrictions imposed by the virtual substitution method. Its result after 10 ms is the following:

$$\exists x_1(r-u_1^2+2u_1x_1-u_2^2+2u_2x_1^2-u_3^2-x_1^4-x_1^2=0 \wedge u_1+2u_2x_1-2x_1^3-x_1=0).$$

Denote the left hand side polynomials of the first and of the second equation by f_1 and f_2, respectively. We conjunctively add another equation $h(\mathbf{u}, r, x_1) = 0$, where h is the remainder of the univariate pseudo-division

$$f_1(\mathbf{u}, r)(x_1)/f_2(\mathbf{u}, r)(x_1).$$

This remainder h is quadratic in x_1. The generic elimination of x_1 can then be completed automatically, yielding a quantifier-free formula $\mathrm{off}_r^{\mathrm{c}}(\pi)'(\mathbf{u})$ with 7 atomic subformulas under the ND-condition $2u_2-1 \neq 0$. The second elimination step takes 4.7 s. The solution is obviously correct for $r < \frac{1}{2}$, which is the minimum radius of the curvature of π adopted at the origin. □

For certain special objects in 3-space there are even simpler descriptions available: Consider, e.g., the r_2-offset of a circle $u_1^2+u_2^2=r_1^2$ wrt. the Euclidean metric, which is a torus T. This torus can be defined by the following formula τ with only one quantifier:

$$\tau(\mathbf{u}) \equiv \exists x\bigl(x^2+u_3^2=r_2^2 \wedge (r_1+x)^2=u_1^2+u_2^2\bigr).$$

The idea behind this description is to regard T as a union of circles in planes parallel to the X-Y-plane.

Example 4.9 We compute a quantifier-free description of the torus with parametric radii given by

$$\tau(\mathbf{u}, r_1, r_2) \equiv \exists x\bigl(x^2+u_3^2=r_2^2 \wedge (r_1+x)^2=u_1^2+u_2^2\bigr).$$

Quantifier elimination and subsequent Gröbner basis simplification yields within 1.2 s the following quantifier-free description:

$$\begin{aligned}\tau'(\mathbf{u}, r_1, r_2) \quad \equiv \quad & r_1^4-2r_1^2r_2^2-2r_1^2u_1^2-2r_1^2u_2^2+2r_1^2u_3^2+r_2^4-2r_2^2u_1^2-2r_2^2u_2^2-\\ & 2r_2^2u_3^2+u_1^4+2u_1^2u_2^2+2u_1^2u_3^2+u_2^4+2u_2^2u_3^2+u_3^4=0 \wedge\\ & r_2^2-u_3^2 \geqslant 0. \quad \square\end{aligned}$$

The concept of r-offsets in the plane is defined analogously to that in 3-space. It should be obvious how to modify all the formulas discussed throughout this section for the planar case. The following example suggests to use iterated offset computations for the problem of rounding corners and edges [Hof96]. We will discuss this in detail in the following chapter; see also [SW97].

Example 4.10 Consider in 2-space the following rectangle of side lengths 1 and 2:

$$\varrho(\mathbf{u}, r) \equiv 0 \leqslant u_1 \leqslant 1 \wedge 0 \leqslant u_2 \leqslant 2.$$

We want to round the corners of ϱ by circular segments of parametric radius $0 < r < \frac{1}{2}$. For this purpose consider the closure of the complement of ϱ described by

$$\overline{\varrho}(\mathbf{u}, r) \equiv \neg(0 < u_1 < 1 \wedge 0 < u_2 < 2).$$

The $\geqslant$-r-offset of $\overline{\varrho}$ is described by the following formula

$$\mathrm{off}_{\geqslant \mathrm{r}}(\overline{\varrho})(\mathbf{u},r) \equiv \forall x_1 \forall x_2 \bigl(\overline{\varrho}[\mathbf{u}/\mathbf{x}] \longrightarrow (x_1-u_1)^2+(x_2-u_2)^2 \geqslant r^2\bigr).$$

Quantifier elimination using the input theory $\vartheta \equiv 0 < r \wedge r < \frac{1}{2}$ and with subsequent Gröbner simplification yields after 153 ms the following quantifier-free description:

$$\begin{aligned}\mathrm{off}_{\geqslant \mathrm{r}}(\overline{\varrho})'(\mathbf{u},r) \quad \equiv \quad & r^2-u_1^2+2u_1-1 \leqslant 0 \wedge r^2-u_1^2-u_2^2+4u_2-4 \leqslant 0 \wedge \\ & r^2-u_1^2-u_2^2 \leqslant 0 \wedge r^2-u_1^2 \leqslant 0 \wedge r^2-u_2^2+4u_2-4 \leqslant 0 \wedge \\ & r^2-u_2^2 \leqslant 0 \wedge u_1-1<0 \wedge u_1>0 \wedge u_2-2<0 \wedge u_2>0.\end{aligned}$$

Under our theory ϑ we can transform this by some easy hand computations including completion of squares into the following simpler form:

$$\mathrm{off}_{\geqslant \mathrm{r}}(\overline{\varrho})''(\mathbf{u},r) \equiv u_1 \geqslant r \wedge 1-r \geqslant u_1 \wedge u_2 \geqslant r \wedge 2-r \geqslant u_2.$$

The correctness of our simplification can be proved automatically in less than 17 ms by applying quantifier elimination to

$$\forall u_1 \forall u_2 \forall r \Bigl(0<r<\frac{1}{2} \longrightarrow \bigl(\mathrm{off}_{\geqslant \mathrm{r}}(\overline{\varrho})'(\mathbf{u},r) \longleftrightarrow \mathrm{off}_{\geqslant \mathrm{r}}(\overline{\varrho})''(\mathbf{u},r)\bigr)\Bigr).$$

The final result of rounding is obtained as

$$\mathrm{off}_{\leqslant \mathrm{r}}\bigl(\mathrm{off}_{\geqslant \mathrm{r}}(\overline{\varrho})''\bigr)(\mathbf{u},r).$$

The corresponding quantifier elimination, again with the theory ϑ, delivers after 323 ms the following quantifier-free description of the rounded rectangle:

$$\begin{aligned}&\mathrm{off}_{\leqslant \mathrm{r}}\bigl(\mathrm{off}_{\geqslant \mathrm{r}}(\overline{\varrho})''\bigr)'(\mathbf{u},r) \equiv \\ &\quad (r^2+2ru_1+2ru_2-6r+u_1^2-2u_1+u_2^2-4u_2+5 \leqslant 0) \vee \\ &\quad (r^2+2ru_1-2ru_2-2r+u_1^2-2u_1+u_2^2+1 \leqslant 0) \vee \\ &\quad (r^2-2ru_1+2ru_2-4r+u_1^2+u_2^2-4u_2+4 \leqslant 0) \vee \\ &\quad (r^2-2ru_1-2ru_2+u_1^2+u_2^2 \leqslant 0) \vee \\ &\quad (r+u_1-1 \leqslant 0 \wedge r-u_1 \leqslant 0 \wedge r+u_2-2 \leqslant 0 \wedge r-u_2 \leqslant 0) \vee \\ &\quad (2ru_1-2r+u_1^2-2u_1+1 \leqslant 0 \wedge r+u_2-2 \leqslant 0 \wedge r-u_2 \leqslant 0) \vee \\ &\quad (2ru_1-u_1^2 \geqslant 0 \wedge r+u_2-2 \leqslant 0 \wedge r-u_2 \leqslant 0) \vee \\ &\quad (2ru_2-4r+u_2^2-4u_2+4 \leqslant 0 \wedge r+u_1-1 \leqslant 0 \wedge r-u_1 \leqslant 0) \vee \\ &\quad (2ru_2-u_2^2 \geqslant 0 \wedge r+u_1-1 \leqslant 0 \wedge r-u_1 \leqslant 0).\end{aligned}$$

It contains 20 atomic formulas. The shortest description we are able to produce by hand contains 12 atomic formulas. For fixed values of r the equivalence between the automatic result and our hand formulation can be shown by quantifier elimination in less than one minute. $\square$

Compare [PP97] and [Wan93a] for a discussion of offsets and examples, using other methods.

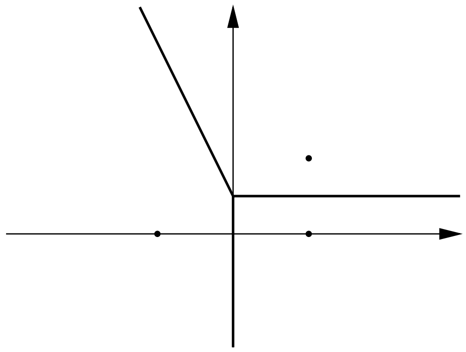

Figure 4.6: Voronoi diagram of three points (Example 4.11)

4.5 Voronoi Diagrams

Let $F = \{A_1, \dots, A_n\}$ be a finite family of objects that constitute non-empty closed sets in space. Fix a metric $d : S^2 \to \mathbb{R}$ that induces the usual topology on S. Then the *Voronoi diagram* wrt. d of F is defined as follows:

$$\mathrm{Vor}(F) = \Big\{ \mathbf{u} \in S \;\Big|\; d(\mathbf{u}, A_i) = d(\mathbf{u}, A_j) = \min_{1 \leqslant k \leqslant n} d(\mathbf{u}, A_k) \text{ with } 1 \leqslant i < j \leqslant n \Big\}.$$

That is the set of all points of minimal equal distance to at least two of the A_1, $\dots$, A_n. The special case of Voronoi diagrams in the plane has been studied in [PS85].

The notion of a Voronoi diagram can be restated in terms of offsets: Suppose that the objects $A_1, \dots, A_n$ are as usual given by quantifier-free formulas $\alpha_1(\mathbf{u})$, $\dots$, $\alpha_n(\mathbf{u})$. Then

$$\mathrm{Vor}(\alpha_1, \dots, \alpha_n)(\mathbf{u}) \quad\equiv\quad \bigvee_{i=1}^{n} \bigvee_{\substack{j=1\\ j\neq i}}^{n} \exists r \Big(\mathrm{off}_{\mathrm{r}}(\alpha_i) \wedge \mathrm{off}_{\mathrm{r}}(\alpha_j) \wedge \bigwedge_{\substack{k=1\\ k\neq i,j}}^{n} \mathrm{off}_{\geqslant\mathrm{r}}(\alpha_k) \Big).$$

Example 4.11 (3 points) In the real plane, the Voronoi diagram of the three points $(-1, 0)$, $(1, 0)$, $(1, 1)$ is described by the following formula:

$$\begin{aligned}
\varphi(\mathbf{u}) \quad\equiv\quad & \exists r\big(r^2 = (u_1-1)^2 + u_2^2 = (u_1+1)^2 + u_2^2 \leqslant (u_1-1)^2 + (u_2-1)^2\big) \vee \\
& \exists r\big(r^2 = (u_1-1)^2 + u_2^2 = (u_1-1)^2 + (u_2-1)^2 \leqslant (u_1+1)^2 + u_2^2\big) \vee \\
& \exists r\big(r^2 = (u_1+1)^2 + u_2^2 = (u_1-1)^2 + (u_2-1)^2 \leqslant (u_1-1)^2 + u_2^2\big).
\end{aligned}$$

Quantifier elimination with subsequent Gröbner simplification yields the following quantifier-free description (170 ms):

$$\begin{aligned}
\varphi'(\mathbf{u}) \quad\equiv\quad & (4u_1 + 2u_2 - 1 = 0 \wedge 4u_2^2 - 4u_2 + 5 \geqslant 0 \wedge 2u_2 - 1 \geqslant 0) \vee \\
& (4u_1^2 - 8u_1 + 5 \geqslant 0 \wedge u_1 \geqslant 0 \wedge 2u_2 - 1 = 0) \vee \\
& (u_1 = 0 \wedge 2u_2 - 1 \leqslant 0). \quad \square
\end{aligned}$$

The two quadratic inequalities in φ' have obviously positive definite polynomials on their left hand sides and are thus redundant:

$$4u_2^2 - 4u_2 + 5 = 4\Big(x - \frac{1}{2}\Big)^2 + 4, \quad 4u_2^2 - 8u_2 + 5 = 4(x-1)^2 + 4.$$

The remainder of φ' yields a nice description of the Voronoi diagram pictured in Figure 4.6.

Example 4.12 (2 lines, circle) The Voronoi diagram of the unit circle and the lines $X = -2$ and $X = 2$ in the plane is described by the formula

$$\begin{aligned}\psi(\mathbf{u}) \quad\equiv\quad & \exists r \exists x_1 \exists x_2 \bigl(x_1^2 + x_2^2 = 1 \land x_2 u_1 = x_1 u_2 \land x_1 u_1 \geqslant 0 \land x_2 u_2 \geqslant 0 \land \\ & \bigl(r^2 = (u_1+2)^2 = (u_1-2)^2 \leqslant (u_1-x_1)^2 + (u_2-x_2)^2 \lor \\ & r^2 = (u_1+2)^2 = (u_1-x_1)^2 + (u_2-x_2)^2 \leqslant (u_1-2)^2 \lor \\ & r^2 = (u_1-2)^2 = (u_1-x_1)^2 + (u_2-x_2)^2 \leqslant (u_1+2)^2\bigr)\bigr).\end{aligned}$$

Generic quantifier elimination yields after 600 ms a formula with 20 atomic subformulas plus the ND-condition $u_1 \neq 0$. Subsequent Gröbner simplification produces the following result (200 ms):

$$\begin{aligned}\psi'(\mathbf{u}) \quad\equiv\quad & (u_1^2 + 3u_1 \leqslant 0 \land 6u_1 - u_2^2 + 9 = 0 \land u_1 + 3 \neq 0 \land u_1 < 0) \lor \\ & (u_1^2 + u_1 \geqslant 0 \land 2u_1 - u_2^2 + 1 = 0 \land u_1 + 1 \neq 0 \land u_1 < 0) \lor \\ & (u_1^2 - 3u_1 \leqslant 0 \land 6u_1 + u_2^2 - 9 = 0 \land u_1 - 3 \neq 0 \land u_1 > 0) \lor \\ & (u_1^2 - u_1 \geqslant 0 \land 2u_1 + u_2^2 - 1 = 0 \land u_1 - 1 \neq 0 \land u_1 > 0).\end{aligned}$$

We easily discover by quantifier elimination on the existential closures of the four constituents of the disjunction ψ' that the second one and the last one are contradictory (50 ms). We thus obtain

$$\begin{aligned}\psi''(\mathbf{u}) \quad\equiv\quad & (u_1^2 + 3u_1 \leqslant 0 \land 6u_1 - u_2^2 + 9 = 0 \land u_1 + 3 \neq 0 \land u_1 < 0) \lor \\ & (u_1^2 - 3u_1 \leqslant 0 \land 6u_1 + u_2^2 - 9 = 0 \land u_1 - 3 \neq 0 \land u_1 > 0).\end{aligned}$$

The entire geometric configuration is pictured in Figure 4.7.

Our ψ'' is obviously equivalent to

$$\begin{aligned}\psi'''(\mathbf{u}) \quad\equiv\quad & (6u_1 - u_2^2 + 9 = 0 \land u_1 + 3 > 0 \land u_1 < 0) \lor \\ & (6u_1 + u_2^2 - 9 = 0 \land u_1 - 3 < 0 \land u_1 > 0).\end{aligned}$$

The current development version of REDLOG contains a an improved simplifier by A. Dolzmann that discovers multiplicative relationships between atomic formulas. This version automatically obtains ψ''' as the result of the Gröbner simplification after generic quantifier elimination.

The ND-condition $u_1 \neq 0$ excludes the u_2 axis, which is obviously the Voronoi diagram for $u_2 \geqslant 3$ and for $u_2 \leqslant -3$. This information is automatically obtained by substituting 0 for u_1 in ψ yielding $\psi_0 = \psi[u_1/0]$. Regular quantifier elimination plus subsequent Gröbner simplification on ψ_0 yields after 100 ms this part of the Voronoi diagram for $u_1 = 0$:

$$\begin{aligned}\psi_0' \quad\equiv\quad & (u_2 + 3 = 0 \lor u_2 - 3 = 0) \lor (u_2^2 + 2u_2 - 3 \geqslant 0 \land u_2 \leqslant 0) \lor \\ & (u_2^2 - 2u_2 - 3 \geqslant 0 \land u_2 \geqslant 0).\end{aligned}$$

This ψ_0' is obviously equivalent to the formula $u_2 \geqslant 3 \lor u_2 \leqslant -3$, which can be automatically shown within 50 ms by applying quantifier elimination to the universal closure of the equivalence. □

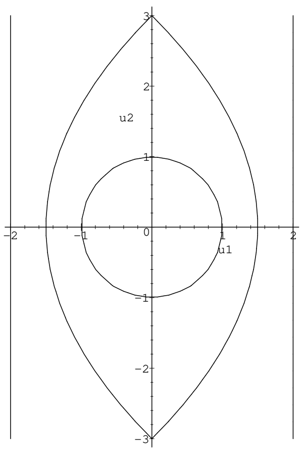

Figure 4.7: Voronoi diagram of a circle and two lines (Example 4.12)

In the special case of only two objects $F = \{A_1, A_2\}$ in 3-space, the Voronoi diagram is the *equi-distance surface* of the objects A_1 and A_2. The following two examples are concerned with the computation of such equi-distance surfaces [Hof96].

Example 4.13 We wish to find a description of the surface defined by equal distance to the unit ball and a parametric plane $X = a$:

$$\begin{aligned}
\sigma_1(\mathbf{u}, a) \quad \equiv \quad & \exists x_1 \exists x_2 \exists x_3 \big((u_1 - a)^2 = (u_1 - x_1)^2 + (u_2 - x_2)^2 + (u_3 - x_3)^2 \land \\
& x_1^2 + x_2^2 + x_3^2 = 1 \land x_1 u_2 = x_2 u_1 \land x_1 u_3 = x_3 u_1 \land \\
& x_2 u_3 = x_3 u_2 \land \big(u_1 x_1 > 0 \lor (u_1 = 0 \land x_1 = 0)\big) \land \\
& \big(u_2 x_2 > 0 \lor (u_2 = 0 \land x_2 = 0)\big) \land \\
& \big(u_3 x_3 > 0 \lor (u_3 = 0 \land x_3 = 0)\big)\big).
\end{aligned}$$

Generic quantifier elimination produces after 1 s a formula σ_1' with 8 atomic subformulas under the ND-condition $u_1 \neq 0$. Fixing $a = 5$ and then applying the Gröbner simplifier, we obtain

$$\begin{aligned}
\sigma_1''(\mathbf{u}) \quad \equiv \quad & (u_1^2 - 6u_1 \geqslant 0 \land 12u_1 + u_2^2 + u_3^2 - 36 = 0 \land u_1 < 0) \lor \\
& (u_1^2 - 4u_1 \leqslant 0 \land 8u_1 + u_2^2 + u_3^2 - 16 = 0 \land u_1 < 0) \lor \\
& (u_1^2 - 6u_1 \leqslant 0 \land 12u_1 + u_2^2 + u_3^2 - 36 = 0 \land u_1 - 6 \neq 0 \land u_1 > 0) \lor \\
& (u_1^2 - 4u_1 \geqslant 0 \land 8u_1 + u_2^2 + u_3^2 - 16 = 0 \land u_1 - 4 \neq 0 \land u_1 > 0).
\end{aligned}$$

We suspect that one of the two contained equations describes the Voronoi diagram: Applying quantifier elimination to

$$\forall u_1 \forall u_2 \forall u_3 \big(u_1 \neq 0 \longrightarrow (\sigma_1'' \longleftrightarrow 12u_1 + u_2^2 + u_3^2 - 36 = 0)\big)$$

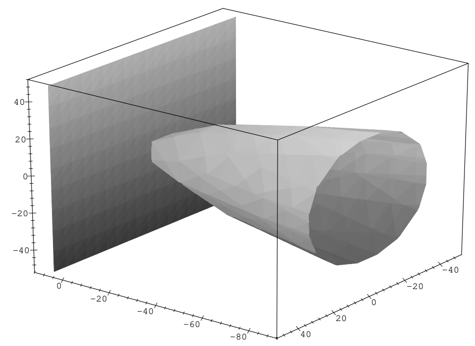

Figure 4.8: The plane $X = 5$ and its equi-distance surface with the unit ball

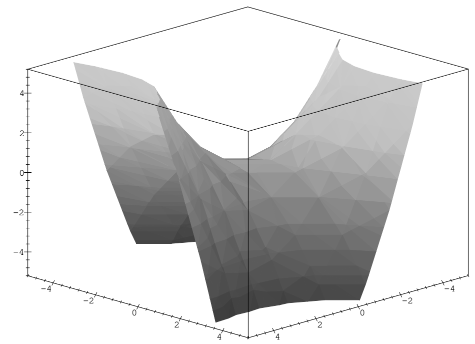

Figure 4.9: The equi-distance surface of the X-axis and the line $X = Y$, $Z = 1$

we in fact obtain "true" after 170 ms. That is, the desired solution for our special case $a = 5$ is the rotation paraboloid

$$u_1 = -\frac{u_2^2}{12} - \frac{u_3^2}{12} + 3.$$

Figure 4.8 pictures this. □

Example 4.14 We compute the equi-distance surface for the X-axis and the line $X = Y$, $Z = 1$:

$$\sigma_2(\mathbf{u}) \equiv \exists x\big(u_2^2 + u_3^2 = (u_1 - x)^2 + (u_2 - x)^2 + (u_3 - 1)^2 \wedge u_1 - x = -(u_2 - x)\big).$$

Elimination yields $u_1^2 - 2u_1u_2 - u_2^2 - 4u_3 + 2 = 0$ after 17 ms. This hyperbolic paraboloid is pictured in Figure 4.9. □

4.6 Collision Problems

Let A, B be objects moving in space from their initial position with time. The question is, whether A and B will collide, and—if yes—when and where is a collision going to happen?

In the easiest case, A and B move along straight lines with constant velocities. Suppose $\alpha(\mathbf{u})$ and $\beta(\mathbf{u})$ describe the objects A and B at time $t = 0$, and $\mathbf{v}^\alpha$, $\mathbf{v}^\beta$ are the velocity vectors for the motion of A and B, respectively. Then the position of the objects A and B at time t is given by $\alpha[\mathbf{u}/\mathbf{u} - t\mathbf{v}^\alpha]$ and $\beta[\mathbf{u}/\mathbf{u} - t\mathbf{v}^\beta]$, respectively. A collision situation of A and B is characterized by both objects having at least one point in common at a certain time:

$$\mathrm{col}(\alpha, \mathbf{v}^\alpha, \beta, \mathbf{v}^\beta, t)(t) \equiv \exists u_1 \exists u_2 \exists u_3 \big(\alpha[\mathbf{u}/\mathbf{u} - t\mathbf{v}^\alpha] \wedge \beta[\mathbf{u}/\mathbf{u} - t\mathbf{v}^\beta]\big).$$

The following formula states that there is a collision time:

$$\mathrm{Col}(\alpha, \mathbf{v}^\alpha, \beta, \mathbf{v}^\beta) \equiv \exists t\big(t > 0 \wedge \mathrm{col}(\alpha, \mathbf{v}^\alpha, \beta, \mathbf{v}^\beta, t)\big).$$

If we eliminate from $\mathrm{Col}(\alpha, \mathbf{v}^\alpha, \beta, \mathbf{v}^\beta)$ the four quantifiers by extended quantifier elimination, we get either "false," i.e., A and B will never collide, or we get "true" together with a common point and the corresponding time.

Example 4.15 (Disc and square) Consider the unit disk $\delta \equiv \mathbf{u}^2 \leqslant 1$ moving with velocity 1 along the X-axis, and a square

$$\sigma \equiv -1 \leqslant u_1 \leqslant 1 \wedge -9 \leqslant u_2 \leqslant -7$$

of side length 2 with center at $(0, -8)$ moving with parametric velocity vector $\mathbf{a}$ from its starting position. A collision situation is described by

$$\begin{aligned} \mathrm{col}(\delta, (1,0), \sigma, \mathbf{a}, t) \quad \equiv \quad & \exists u_1 \exists u_2 \big((u_1 - t)^2 + u_1^2 \leqslant 1 \wedge \\ & -1 \leqslant u_1 - a_1 t \leqslant 1 \wedge -9 \leqslant u_2 - a_2 t \leqslant -7\big). \end{aligned}$$

For obtaining necessary and sufficient conditions on the parametric velocity vector $\mathbf{a}$ for a collision to take place, we apply generic quantifier elimination to the corresponding formula

$$\mathrm{Col}(\delta, (1,0), \sigma, \mathbf{a})(\mathbf{a}) \equiv \exists t\big(t > 0 \wedge \mathrm{col}(\delta, (1,0), \sigma, \mathbf{a}, t)\big).$$

This yields after 1.3 s a quantifier-free formula $\mathrm{Col}(\delta, (1,0), \sigma, \mathbf{a})'(\mathbf{a})$ with 69 atomic subformulas under the ND-condition $a_1^2 - 2a_1 + a_2^2 + 1 \neq 0$, which obviously excludes only the velocity vector $\mathbf{a} = (1, 0)$.

This example is generalized from a decision problem in [CH91], where the parameters a_1, a_2 were both fixed to $\frac{17}{16}$. We plug into $\mathrm{Col}(\delta, (1,0), \sigma, \mathbf{a})(\mathbf{a})$ these original values for $\mathbf{a}$, and apply extended quantifier elimination. The result after 85 ms is "true" together with the answer $t = \frac{144}{17}$ and $\mathbf{u} = \big(\frac{144}{17}, 0\big)$. $\square$

The elimination answer for $\mathrm{Col}(\alpha, \mathbf{v}^\alpha, \beta, \mathbf{v}^\beta)$ will, in general, not describe the first collision of A and B. In order to obtain time and location of the first contact of the two objects, we have to apply quantifier elimination with answer to the formula

$$\begin{aligned} \mathrm{Col}^1(\alpha, \mathbf{v}^\alpha, \beta, \mathbf{v}^\beta) \quad \equiv \quad & \exists t\big(t > 0 \wedge \mathrm{col}(\alpha, \mathbf{v}^\alpha, \beta, \mathbf{v}^\beta, t) \wedge \\ & \forall t'(0 < t' < t \longrightarrow \neg\mathrm{col}(\alpha, \mathbf{v}^\alpha, \beta, \mathbf{v}^\beta, t'))\big). \end{aligned}$$

Example 4.16 (Disc and square revisited) We return to our previous example. We now check whether the collision found for the concrete velocity vector $\mathbf{a} = \left(\frac{17}{16}, \frac{17}{16}\right)$ describes the first collision. For this we eliminate the quantifiers from the formula

$$\begin{aligned}\mathrm{Col}^1\left(\delta, (1,0), \sigma, \left(\tfrac{17}{16}, \tfrac{17}{16}\right)\right) \equiv\ & \exists t\big(t > 0 \wedge \mathrm{col}\big(\delta, (1,0), \sigma, \left(\tfrac{17}{16}, \tfrac{17}{16}\right), t\big) \wedge \\ & \forall t'\big(0 < t' < t \longrightarrow \neg\mathrm{col}\big(\delta, (1,0), \sigma, \left(\tfrac{17}{16}, \tfrac{17}{16}\right), t'\big)\big)\big).\end{aligned}$$

The result obtained after 629 ms is "true" together with the answer $t = \frac{96}{17}$ and $\mathbf{u} = \left(\frac{96}{17}, -1\right)$ showing the true first collision. $\square$

The next example is a very strong 3-dimensional generalization of the disc-square collision considered above.

Example 4.17 (Quadric and cube) Starting with its center located at the origin, a solid parametric quadric $\eta(\mathbf{u}, \mathbf{a}) \equiv a_1 u_1^2 + a_2 u_2^2 + a_3 u_3^2 \leqslant 1$ with main axes parallel to the coordinate axes moves with velocity 1 along the X-axis. A cube $\kappa(\mathbf{u}, \mathbf{b}) \equiv \bigwedge_{i=1}^3 b_i - 1 \leqslant u_i \leqslant b_i + 1$ with side length 2 moves starting with its center at $\mathbf{b}$ with constant parametric velocity $\mathbf{v} = (v_1, v_2, v_3)$ through space. We have

$$\begin{aligned}\mathrm{col}(\eta, \mathbf{e_1}, \kappa, \mathbf{v}, t)(\mathbf{a}, \mathbf{b}, \mathbf{v}, t) \equiv\ & \exists u_1 \exists u_2 \exists u_3 \Big(a_1(u_1 - t) + a_2 u_2 + a_3 u_3 \leqslant 1 \wedge \\ & \bigwedge_{i=1}^{3} b_i - 1 \leqslant u_i - v_i t \leqslant b_i + 1\Big).\end{aligned}$$

Our input formula for generic quantifier elimination is the corresponding formula

$$\mathrm{Col}(\eta, \mathbf{e_1}, \kappa, \mathbf{v})(\mathbf{a}, \mathbf{b}, \mathbf{v}) \equiv \exists t\big(t > 0 \wedge \mathrm{col}(\eta, \mathbf{e_1}, \kappa, \mathbf{v}, t)(\mathbf{a}, \mathbf{b}, \mathbf{v}, t)\big),$$

which contains 9 parameters. Generic quantifier elimination yields a quantifier-free formula with 2792 atomic subformulas under 7 ND-conditions in 43.4 s.

We now leave only the velocity vector $\mathbf{v}$ of the cube parametric, but fix the dimensions of the quadric to $\mathbf{a_0} = (3, 2, 1)$ and the starting position for the cube to $\mathbf{b_0} = (1, -10, 15)$ yielding $\eta_0(\mathbf{u})$ and $\kappa_0(\mathbf{u})$, respectively. This reduces the timing for generic quantifier elimination on the corresponding $\mathrm{Col}(\eta_0, \mathbf{e_1}, \kappa_0, \mathbf{v})(\mathbf{v})$ to 9.4 s yielding only 336 atomic formulas under 5 ND-conditions. For this specialized situation, we ask for a velocity vector such that a collision takes place not later than at the time $t = 10$:

$$\exists v_1 \exists v_2 \exists v_3 \exists t\big(0 < t \leqslant 10 \wedge \mathrm{col}(\eta_0, \mathbf{e_1}, \kappa_0, \mathbf{v}, t)(t, \mathbf{v})\big).$$

Applying extended quantifier elimination, we obtain after 272 ms "true" together with the answer $\mathbf{v} = \left(1, \frac{11}{10}, -\frac{7}{5}\right)$, $t = 10$, and $\mathbf{u} = (10, 0, 0)$. $\square$

The remainder of this section is devoted to a case study that illustrates some more sophisticated techniques for modeling with first order formulas over the reals. This includes in particular coping with periodicities and handling of transcendental items.

Example 4.18 (Circling descending objects) We consider two objects circling in descending spirals in a cylinder above a circle with radius r. Each object

Table 4.1: Data in both models for object collision; k, l, and n are not first-order variables but concrete integers

t	time
r	radius of the circle
(p_1, p_2)	starting position of object 1
(q_1, q_2)	starting position of object 2
(v_1, v_2)	velocity of object 1
(w_1, w_2)	velocity of object 2
k	number of complete laps of object 1
l	number of complete laps of object 2
n	a bound on the number of complete laps of both objects

has a given position and constant horizontal and vertical velocities. The question is, whether the two objects are going to crash within a given number of laps.

All distances are given in meters, velocities are given in meters per second. For velocities and positions we take coordinates (C_1, C_2), where C_1 is the position on the circumference, i.e. the distance on the circumference of the circle from a fixed reference point, and C_2 is the height. We consider two different ways of modeling the situation:

Model 1 In the simpler model objects are modeled as points, and a collision occurs if the two points coincide. Table 4.1 summarizes the data involved in this model (and also in the other one). A collision is then described by the following formula:

$$\begin{aligned}\varphi_1 \quad \equiv \quad & p_2 + tv_2 = q_2 + tw_2 \wedge \bigvee_{k=0}^{n} \bigvee_{l=0}^{n} \big(2\pi rk \leqslant tv_1 < 2\pi r(k+1) \wedge \\ & 2\pi rl \leqslant tw_1 < 2\pi r(l+1) \wedge p_1 + tv_1 - 2\pi rk = q_1 + tw_1 - 2\pi rl\big).\end{aligned}$$

Model 2 In the second, more realistic, model we prescribe, in addition, horizontal and vertical safety distances, see Table 4.2. A "collision" occurs, if the points representing the objects violate both safety distances. The description of a collision then reads as follows:

$$\begin{aligned}\varphi_2 \quad \equiv \quad & -h \leqslant p_2 + tv_2 - (q_2 + tw_2) \leqslant h \wedge \\ & \bigvee_{k=0}^{n} \bigvee_{l=0}^{n} \big(2\pi rk \leqslant tv_1 < 2\pi r(k+1) \wedge 2\pi rl \leqslant tw_1 < 2\pi r(l+1) \wedge \\ & -w \leqslant p_1 + tv_1 - 2\pi rk - (q_1 + tw_1 - 2\pi rl) \leqslant w\big).\end{aligned}$$

One may be tempted to replace the disjunctions by an unbounded existential quantification $\exists k \exists l$. This is, however, not possible in out context, because these quantifiers would range over integers. In principle mixed real/integer elimination is possible [Wei90, Wei97a, Wei99].

For obtaining a well-formed input formula we have to fix n to a natural number. Setting $n = 3$ our φ_1 has 81 atomic subformulas in 10 parameters plus

Table 4.2: Additional data for Model 2

w	horizontal safety distance
h	vertical safety distance

the symbol π. Our algorithms do not know anything about π. It plays the role of just another parameter.

Eliminating the time t by applying generic quantifier elimination to $\psi_1 \equiv \exists t \varphi_1$, we obtain after 1.9 s necessary and sufficient conditions

$$\psi_1'(r, p_1, p_2, q_1, q_2, v_1, v_2, w_1, w_2, \pi)$$

in the other parameters for a collision to take place. This ψ_1' is valid under the simultaneously obtained ND-condition $v_2 - w_2 \neq 0$. It contains 80 atomic formulas.

We proceed analogously for the second model, and obtain after 48.8 s a quantifier-free formula

$$\psi_2'(r, p_1, p_2, q_1, q_2, v_1, v_2, w_1, w_2, h, w, \pi)$$

without any ND-condition. This ψ_2' contains 4597 atomic formulas.

Let us now turn to a less parametric situation. We fix in φ_1 and φ_2 the following values for the radius of the circle, the starting positions, both velocities of object 1, the horizontal velocity of object 2, and the horizontal and vertical safety distances yielding $\varphi_1^*(t, w_2, \pi)$ and $\varphi_2^*(t, w_2, \pi)$, respectively:

$$\begin{aligned} r &= 10000 \\ (p_1, p_2) &= (0, 9000) \\ (q_1, q_2) &= (2000, 10000) \\ (v_1, v_2) &= (100, -3) \\ w_1 &= 50 \\ h &= 50 \\ w &= 500. \end{aligned}$$

Applying extended quantifier elimination to $\exists t \varphi_1^*$ we ask for vertical velocities w_2 of object 2 that will lead to a collision within at most 3 laps of each object. We obtain after 391 ms two possible conditions on w_2 together with corresponding answers for the collision times:

$$\begin{bmatrix} 10\pi w_2 + 30\pi + w_2 + 28 = 0 \land 5\pi - 1 > 0 & t = 400\pi + 40 \\ 5\pi - 1 > 0 \land w_2 + 28 = 0 & t = 40 \end{bmatrix}.$$

This means in rounded numerical values that a collision occurs within the first 3 laps if either $w_2 \approx -3.77$ or $w_2 = -28$. In the first case the collision time is $t \approx 1296.64$, in the second case it is $t = 40$.

The corresponding call applied to $\exists t \varphi_2^*$ takes 1.7 s yielding 6 conditions involving w_2 and π together with suitable answers for t. At least two of these cases have conditions γ which cannot hold for the true π. This is automatically found out by deciding $\exists w_2 \exists \pi (3.1 < \pi < 3.2 \wedge \gamma)$ via quantifier elimination, which takes 8.7 s for all 6 cases together.

Finally, we check the danger of a collision for concrete vertical speed ranges in the second model: Deciding the formula

$$\exists w_2 \exists \pi \exists t (3.1 < \pi < 3.2 \wedge w_2 \geqslant -3.8 \wedge \varphi_2^*)$$

by quantifier elimination yields "true" after 799 ms, while for

$$\exists w_2 \exists \pi \exists t (3.1 < \pi < 3.2 \wedge w_2 \geqslant -3.7 \wedge \varphi_2^*)$$

we obtain "false" after 835 ms. Taking these results together, some vertical speed $-3.8 \leqslant w_2 < -3.7$ will—modulo the approximate π—lead to a collision during the first 3 laps. The prescription $w_2 \geqslant -3.7$, in contrast, is collision-safe for this period. $\square$

4.7 Conclusions

We have discussed in this chapter the formulation of various standard problems of computational geometry as quantifier elimination problems or extended quantifier elimination problems. Applying both regular and extended generic quantifier elimination to such formulations, it has turned out that we can handle a large range of these problems even parametrically. Such a parametric solution actually provides the solutions for an entire class of problems given by all possible interpretations for the parameters. The single or even double exponential worst-case complexity of our method has to be viewed and judged on the basis of this fact.

Again, generic quantifier elimination has turned out crucial for our success. In contrast to the application to geometry proving in the previous chapter, this is, however, mainly for efficiency reasons. The effect of obtaining important extra information by the assumptions made by the generic elimination procedure has not shown here. On the other hand, the use of generic quantifier elimination does not impose any restrictions that cannot be overcome; with Example 4.12 we have demonstrated how to deal with ND-conditions that even excluded highly relevant parts of the solution. The major part of the examples discussed here cannot be satisfactorily treated by any available implementation of regular, i.e. non-extended and non-generic, quantifier elimination.

Recall that we have defined generic quantifier elimination in Section 2.4.2 in such a way that it does not only generate a theory of subsidiary conditions in the parameters but also already accepts conditions in the parameters as an input theory, which is then extended. In Example 4.4 we have seen how to make use of this concept for efficiently and cleanly coding positiveness conditions.

By means of a rather sophisticated analysis of possible collisions of two objects we have finally illustrated how to operate with two concepts that do not straightforwardly fit into our general setting: periodicities and computing with given transcendental numbers such as π.

Chapter 5

Solid Modeling

Solid modeling knows two major representation schemes for solid objects: *boundary representation* (B-REP) and *constructive solid geometry* (CSG). B-REP is very close to the image rendered on the screen. CSG, in contrast, resembles the inherent structure of the described solid by combining primitive solids via regularized boolean operations. Today's modelers are usually dual representation modelers, which combine both schemata to make use of the particular advantages of each of them [Hof89].

We propose to supplement modelers with a third component representing solids as quantifier-free formulas. Among all semialgebraic sets, we define regular closed sets to be valid solids. The treatment of various problems from computational geometry within our framework has already been discussed in the previous chapter, cf. also [SW98]. This includes parallel and central projections, corresponding reconstruction problems, offsets of algebraic surfaces, Voronoi diagrams, and collision problems. From a theoretical point of view, it is clear that our descriptions are more general than CSG and can thus replace it. It will have to be examined whether this is feasible in practice.

Our notion of a solid obviously extends that of CSG. In particular, it includes common types of offsetted, rounded and blended solids. In fact, all operations commonly encountered in today's solid modelers do not lead outside the realm of semialgebraic sets. To show this, we have to redefine all desired operations to map formulas to formulas, and we have to give algorithms computing these maps. For operations involving parallel or central projection we have in fact already done so in the previous chapter. It turns out that all other operations can also be straightforwardly encoded with quantifiers.

Section 5.1 does so for the usual boolean and topological operations. Section 5.2 then formally introduces solids as regular closed semialgebraic sets. Section 5.3 examines in detail offsets of solids in our sense. The relevance of offsets for rounding and blending has already been indicated in Example 4.10 in the previous chapter. Rounding and blending of solids by iterated offset computations is discussed in detail in Section 5.4. Our approach to this allows to naturally handle *arbitrary* semialgebraic sets as intermediate results of our operations. It shall turn out that this freedom, together with a new concept of defining blendings in terms of offsets, allows us to understand and to overcome some well-known problems with the definition of blendings via offsets [Ros85, RR86]. Section 5.5 returns to the issue of integrating our ideas into the framework currently available. We illustrate, how the B-REP representation of our solids can be computed from our description by quantifier-free formulas. Section 5.6 finally shows how to treat geometric feasibility problems in our framework. The background for this is that the latest generation of CAD systems is being designed to accept as input declarative descriptions of scenes instead of procedural drawing commands.

As in the previous chapter, we will be able to solve all problems parametrically. We can, e.g., compute the blend of a solid with respect to a parametric radius r. The resulting description will contain case distinctions that make it correct for any substitution of a positive real number for r. Similarly, the solids themselves can be described parametrically, and the quantifier elimination work can be done once for an entire class of solids. The obtained information can be stored in a database, and then be used by simply substituting concrete dimensions.

In the literature there have been already purely algebraic approaches discussed [Hof89, Hof90]. Such approaches can in contrast to our semialgebraic quantifier elimination techniques, of course, not handle polynomial inequalities.

Throughout this chapter, let—except where mentioned otherwise—d denote the Euclidean metric in $\mathbb{R}^n$, and recall the definition of the corresponding formulas δ, $\delta_{\geqslant}$, and $\delta_{\leqslant}$ from Section 4.4. These definitions extend naturally to $\delta_{>}$ and $\delta_{<}$.

We have again performed all our computations on a SUN Ultra 1 Model 140 workstation using 32 MB of memory.

5.1 Boolean and Topological Operations

Taking unions, intersections, complements, interiors, and closures leads us from semialgebraic sets to semialgebraic sets. Denote by

$$S\big(\varphi(\mathbf{u})\big) = \big\{\, \mathbf{x} \in \mathbb{R}^n \;\big|\; \varphi(\mathbf{x}) \,\big\}$$

the semialgebraic set described by the quantifier-free formula φ. We can then carry out boolean and closure operations on the formula representations as follows:

$$\begin{aligned}
S(\varphi) \cup S(\psi) &= S(\varphi \vee \psi)\\
S(\varphi) \cap S(\psi) &= S(\varphi \wedge \psi)\\
\sim S(\varphi) &= S(\neg\varphi)\\
S(\varphi(\mathbf{u}))^{\circ} &= S\big(\exists r(r > 0 \wedge \forall x_1 \forall x_2 \forall x_3 (\delta_{<}(\mathbf{x}, \mathbf{u}, r) \longrightarrow \varphi[\mathbf{u}/\mathbf{x}]))\big)\\
\overline{S(\varphi(\mathbf{u}))} &= S\big(\forall r(r > 0 \longrightarrow \exists x_1 \exists x_2 \exists x_3 (\delta_{<}(\mathbf{x}, \mathbf{u}, r) \wedge \varphi[\mathbf{u}/\mathbf{x}]))\big).
\end{aligned}$$

Note that this explains also how to perform any composition of these operations on the formula level, including particularly the regular closure, regularized boolean operations, and taking the boundary $\partial S(\varphi) = \overline{S(\varphi)} \cap \overline{\sim S(\varphi)}$.

5.2 Solids as Regular Closed Semialgebraic Sets

Our intuition of a solid in $\mathbb{R}^3$ is that it is a closed semialgebraic set that is "everywhere 3-dimensional." It is thus reasonable and compatible with the more specialized definitions commonly used [Hof90] to define a solid as a non-empty *regular closed* semialgebraic subset of $\mathbb{R}^3$.

Definition 5.1 (Regular closure) For $A \subseteq \mathbb{R}^n$ the *regular closure* A^* is defined as the closure of the interior $\overline{A^\circ}$ of A, and A is called regular closed if $A^* = A$.

The dual concept of regular open sets has been studied intensively in the theory of boolean algebras, where it is the crucial tool for the completion of boolean algebras [Hal63]. The results on regular open sets can easily be transfered to regular closed sets via duality.

For open sets A we have $A^* \supseteq A$, and for closed sets A we have $A^* \subseteq A$. If A is neither open nor closed, there need not hold any inclusion, e.g.:

$$\big(]0,1[\cup \{2\}\big)^* = [0,1].$$

Since both interior and closure are monotone, so is the regular closure, i.e., $X \subseteq A$ implies $X^* \subseteq A^*$. It follows that if $X \subseteq A$ is already regular closed, then $X \subseteq A^*$. Finally, it is not hard to see that the regular closure is idempotent, i.e. $A^{**} = A^*$.

For closed sets A the formation of A^* will—roughly speaking—kill lower dimensional parts of A. This is exactly what is intended in the well-known formation of the normalized boolean operations such as the normalized intersection

$$A \cap^* B := (A \cap B)^*.$$

In the literature this regularization is usually carried out with all boolean operations. From a computational point of view, it might be noteworthy that this is a certain overkill. For regular closed input solids A and B we have:

$$(A \cup B)^* = A \cup B, \quad (\sim A)^* = \overline{\sim A}, \quad (A \setminus B)^* = \overline{A \setminus B},$$

i.e., the union requires no regularization at all, and for regularizing complement and difference one need not take the interior. Regularized intersection, in contrast, really requires the time consuming operations of computing interior and closure. Note that, at least in our framework, it cannot be efficiently recognized whether a set is already open. So this external knowledge will save superfluous computations. For the normalized intersection however, we actually have to compute the closure of the interior.

5.3 Offsets

An important geometric construction for our purposes is the formation of *offsets* as already introduced in Section 4.4. Given *arbitrary* $A \subseteq \mathbb{R}^n$ and $0 < r \in \mathbb{R}$ we define:

$$\mathrm{off}_{\leqslant \mathrm{r}}(A) = \{\, \mathbf{x} \in \mathbb{R}^n \mid \exists(\mathbf{a} \in A) : d(\mathbf{x}, \mathbf{a}) \leqslant r \,\}.$$

As we already know, the offset of a semialgebraic set is again semialgebraic:

$$\mathrm{off}_{\leqslant \mathrm{r}}\big(S(\varphi(\mathbf{u}))\big) = S\big(\exists x_1 \exists x_2 \exists x_3 (\varphi[\mathbf{u}/\mathbf{x}] \wedge \delta_{\leqslant}(\mathbf{x}, \mathbf{u}, r)\big).$$

This notion of an offset, also called *expanding*, is quite standard [RR86]. Also, it has already been applied to non-regular sets [Ros85]. For regular closed A, we have that $\mathrm{off}_{\leqslant \mathrm{r}}(A)$ is again regular closed.

There is an "opposite" operation of *shrinking* a set. The result of this is not necessarily regular closed even when applied to regular closed input sets. Thus shrinking has usually been used in combination with regularization similar to the regularized boolean operations discussed above. Compositions of expanding and shrinking have been used to perform blending operations on solids [RR86].

Combining expanding and shrinking for blending solids, one is unfortunately faced with unintuitive results, which we are going to analyze in detail in Section 5.4.3. We will exhibit that these problems are actually caused by the regularization of shrinkings. To overcome the problems, we choose an alternate approach to defining blendings in terms of offsets. Therefore, we generalize our notion of an offset by defining also the following:

$$\begin{aligned}
\mathrm{off}_{\geqslant \mathrm{r}}(A) &= \{\, \mathbf{x} \in \mathbb{R}^n \mid \forall(\mathbf{a} \in A) : d(\mathbf{x}, \mathbf{a}) \geqslant r \,\}, \\
\mathrm{off}_{< \mathrm{r}}(A) &= \{\, \mathbf{x} \in \mathbb{R}^n \mid \exists(\mathbf{a} \in A) : d(\mathbf{x}, \mathbf{a}) < r \,\}, \\
\mathrm{off}_{> \mathrm{r}}(A) &= \{\, \mathbf{x} \in \mathbb{R}^n \mid \forall(\mathbf{a} \in A) : d(\mathbf{x}, \mathbf{a}) > r \,\}.
\end{aligned}$$

Just like $\mathrm{off}_{\leqslant \mathrm{r}}$, all these operations lead from semialgebraic sets to semialgebraic sets, e.g.,

$$\mathrm{off}_{\geqslant \mathrm{r}}\big(S\big(\varphi(\mathbf{u})\big)\big) = S\big(\forall x_1 \forall x_2 \forall x_3 (\varphi[\mathbf{u}/\mathbf{x}] \longrightarrow \delta_{\leqslant}(\mathbf{x}, \mathbf{u}, r))\big).$$

The remainder of this section is devoted to a careful study of the properties of our various offsets and their interaction with boolean and topological operations.

Note that our definitions are not necessarily conform with the notion of *distance* between points and sets. For instance,

$$1 \notin \mathrm{off}_{\leqslant 1}\,]{-\infty}, 0[\; = \;]{-\infty}, 1[\,, \quad \text{but} \quad d\big(1,]{-\infty}, 0[\big) = \inf_{x<0} d(1, x) = 1.$$

In other words, there is a point of distance 1 from the given set not contained in its $\mathrm{off}_{\leqslant 1}$. For closed $A \subseteq \mathbb{R}^n$, we can, however, assert that

$$\mathrm{off}_{\leqslant \mathrm{r}}(A) = \{\, \mathbf{x} \in \mathbb{R}^n \mid d(\mathbf{x}, A) \leqslant r \,\} \quad \text{and} \quad \mathrm{off}_{\geqslant \mathrm{r}}(A) = \{\, \mathbf{x} \in \mathbb{R}^n \mid d(\mathbf{x}, A) \geqslant r \,\}.$$

It follows immediately from the definitions that $\mathrm{off}_{\leqslant \mathrm{r}}(A) = {\sim}\,\mathrm{off}_{> \mathrm{r}}(A)$ and $\mathrm{off}_{< \mathrm{r}}(A) = {\sim}\,\mathrm{off}_{\geqslant \mathrm{r}}(A)$. The following lemma is concerned with offsets of unions.

Lemma 5.2 For $A \subseteq \mathbb{R}^n$ and $0 < r \in \mathbb{R}$ the following assertions hold:

1. $\mathrm{off}_{\leqslant \mathrm{r}}(A \cup B) = \mathrm{off}_{\leqslant \mathrm{r}}(A) \cup \mathrm{off}_{\leqslant \mathrm{r}}(B)$
2. $\mathrm{off}_{\geqslant \mathrm{r}}(A \cup B) = \mathrm{off}_{\geqslant \mathrm{r}}(A) \cap \mathrm{off}_{\geqslant \mathrm{r}}(B)$.

Analogous results hold when replacing "$\leqslant$" by "$<$" and "$\geqslant$" by "$>$," respectively.

Proof 1. $\mathbf{x} \in \mathrm{off}_{\leqslant \mathrm{r}}(A \cup B)$ iff there exists $\mathbf{y} \in A \cup B$ such that $d(\mathbf{x}, \mathbf{y}) \leqslant r$ iff there exists $\mathbf{y} \in A$ such that $d(\mathbf{x}, \mathbf{y}) \leqslant r$ or there exists $\mathbf{y} \in B$ such that $d(\mathbf{x}, \mathbf{y}) \leqslant r$ iff $\mathbf{x} \in \mathrm{off}_{\leqslant \mathrm{r}}(A)$ or $\mathbf{x} \in \mathrm{off}_{\leqslant \mathrm{r}}(B)$ iff $\mathbf{x} \in \mathrm{off}_{\leqslant \mathrm{r}}(A) \cup \mathrm{off}_{\leqslant \mathrm{r}}(B)$. The same argument holds for "$<$" instead of "$\leqslant$."

2. $\mathbf{x} \in \mathrm{off}_{\geqslant \mathrm{r}}(A \cup B)$ iff for all $\mathbf{c} \in A \cup B$ we have $d(\mathbf{x}, \mathbf{c}) \geqslant r$ iff for all $\mathbf{a} \in A$ we have $d(\mathbf{x}, \mathbf{a}) \geqslant r$ and for all $\mathbf{b} \in B$ we have $d(\mathbf{x}, \mathbf{b}) \geqslant r$ iff $\mathbf{x} \in \mathrm{off}_{\geqslant \mathrm{r}}(A)$ and $\mathbf{x} \in \mathrm{off}_{\geqslant \mathrm{r}}(B)$ iff $\mathbf{x} \in \mathrm{off}_{\geqslant \mathrm{r}}(A) \cap \mathrm{off}_{\geqslant \mathrm{r}}(B)$. The same argument holds for "$>$" instead of "$\geqslant$." $\square$

Next, the closure properties of our offsets are quite surprising. For some offsets they depend on the set considered, while for other offsets they do not.

Lemma 5.3 For $A \subseteq \mathbb{R}^n$ and $0 < r \in \mathbb{R}$ the following assertions hold:

1. $\mathrm{off}_{<\mathrm{r}}(A)$ is open and $\mathrm{off}_{\geqslant \mathrm{r}}(A)$ is closed.

2. If A is open, then $\mathrm{off}_{\leqslant \mathrm{r}}(A)$ is open and $\mathrm{off}_{>\mathrm{r}}(A)$ is closed.

3. If A is closed, then $\mathrm{off}_{\leqslant \mathrm{r}}(A)$ is closed and $\mathrm{off}_{>\mathrm{r}}(A)$ is open.

Proof 1. Let $\mathbf{x} \in \mathrm{off}_{<\mathrm{r}}(A)$, pick $\mathbf{a} \in A$, $0 < \varepsilon \in \mathbb{R}$ such that $d(\mathbf{x}, \mathbf{a}) = r - \varepsilon < r$. For $\mathbf{y} \in \mathrm{B}(\mathbf{x}, \varepsilon)$ we have $d(\mathbf{y}, \mathbf{a}) \leqslant d(\mathbf{y}, \mathbf{x}) + d(\mathbf{x}, \mathbf{a}) < \varepsilon + r - \varepsilon = r$, i.e., $\mathrm{B}(\mathbf{x}, \varepsilon) \subseteq \mathrm{off}_{<\mathrm{r}}(A)$.

2. For open A let $\mathbf{x} \in \mathrm{off}_{\leqslant \mathrm{r}}(A)$, say $\mathbf{a} \in A$ such that $d(\mathbf{x}, \mathbf{a}) \leqslant r$. Choose $0 < \varepsilon \in \mathbb{R}$ such that $\mathrm{B}(\mathbf{a}, \varepsilon) \subseteq A$. For $\mathbf{y} \in \mathrm{B}(\mathbf{x}, \frac{\varepsilon}{2})$ pick $\mathbf{z}$ on the line segment $\overline{\mathbf{ax}}$ with $d(\mathbf{a}, \mathbf{z}) = \frac{\varepsilon}{2}$. Then $d(\mathbf{y}, \mathbf{z}) \leqslant d(\mathbf{y}, \mathbf{x}) + d(\mathbf{x}, \mathbf{z}) < \frac{\varepsilon}{2} + r - \frac{\varepsilon}{2} = r$, i.e., $\mathrm{B}(\mathbf{x}, \frac{\varepsilon}{2}) \subseteq \mathrm{off}_{\leqslant \mathrm{r}}(A)$.

3. For closed A, we show that $\overline{\mathrm{off}_{\leqslant \mathrm{r}}(A)} \subseteq \mathrm{off}_{\leqslant \mathrm{r}}(A)$: Let $\mathbf{x} \in \overline{\mathrm{off}_{\leqslant \mathrm{r}}(A)}$, and assume for a contradiction that $\mathbf{x} \in \mathrm{off}_{>\mathrm{r}}(A) = {\sim}\mathrm{off}_{\leqslant \mathrm{r}}(A)$. Since A is closed, there is $\mathbf{a} \in A$ and $0 < \varepsilon \in \mathbb{R}$ such that $d(\mathbf{x}, \mathbf{a}) = d(\mathbf{x}, A) = r + \varepsilon > r$. Pick $\mathbf{y} \in \mathrm{B}(\mathbf{x}, \varepsilon) \cap \mathrm{off}_{\leqslant \mathrm{r}}(A) \neq \emptyset$, then $d(\mathbf{x}, \mathbf{a}) \leqslant d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{a}) < \varepsilon + r$, a contradiction. $\square$

Finally, we turn to the interplay between offsets and topological operations and regularizations.

Lemma 5.4 1. Let $A \subseteq \mathbb{R}^n$, and let $0 < r \in \mathbb{R}$. Then

$$\begin{aligned}
\mathrm{off}_{>\mathrm{r}}(A)^\circ &= \overset{\circ}{\overline{\mathrm{off}_{>\mathrm{r}}(A)}} = \mathrm{off}_{\geqslant \mathrm{r}}(A)^\circ = \overset{\circ}{\overline{\mathrm{off}_{\geqslant \mathrm{r}}(A)}} = \mathrm{off}_{>\mathrm{r}}(\overline{A}) \\
&\subseteq \mathrm{off}_{>\mathrm{r}}(A) \\
&\subseteq \overline{\mathrm{off}_{>\mathrm{r}}(A)} \\
&\subseteq \mathrm{off}_{\geqslant \mathrm{r}}(\overline{A}) = \mathrm{off}_{\geqslant \mathrm{r}}(A) = \overline{\mathrm{off}_{\geqslant \mathrm{r}}(A)}, \\
\mathrm{off}_{>\mathrm{r}}(A)^* &= \mathrm{off}_{\geqslant \mathrm{r}}(A)^*.
\end{aligned}$$

2. If A is open, then

$$\begin{aligned}
\mathrm{off}_{>\mathrm{r}}(A)^\circ &= \overset{\circ}{\overline{\mathrm{off}_{>\mathrm{r}}(A)}} = \mathrm{off}_{\geqslant \mathrm{r}}(A)^\circ = \overset{\circ}{\overline{\mathrm{off}_{\geqslant \mathrm{r}}(A)}} = \mathrm{off}_{>\mathrm{r}}(\overline{A}) \\
&\subseteq \mathrm{off}_{>\mathrm{r}}(A)^* = \mathrm{off}_{\geqslant \mathrm{r}}(A)^* \\
&\subseteq \mathrm{off}_{>\mathrm{r}}(A) = \overline{\mathrm{off}_{>\mathrm{r}}(A)} = \mathrm{off}_{\geqslant \mathrm{r}}(\overline{A}) = \mathrm{off}_{\geqslant \mathrm{r}}(A) = \overline{\mathrm{off}_{\geqslant \mathrm{r}}(A)}.
\end{aligned}$$

3. If A is closed, then

$$\begin{aligned}
\mathrm{off}_{>\mathrm{r}}(A)^{\circ} &= \mathring{\overline{\mathrm{off}_{>\mathrm{r}}(A)}} = \mathrm{off}_{\geqslant\mathrm{r}}(A)^{\circ} = \mathring{\overline{\mathrm{off}_{\geqslant\mathrm{r}}(A)}} = \mathrm{off}_{>\mathrm{r}}(\overline{A}) = \mathrm{off}_{>\mathrm{r}}(A) \\
&\subseteq \mathrm{off}_{>\mathrm{r}}(A)^{*} = \overline{\mathrm{off}_{>\mathrm{r}}(A)} = \mathrm{off}_{\geqslant\mathrm{r}}(A)^{*} \\
&\subseteq \mathrm{off}_{\geqslant\mathrm{r}}(\overline{A}) = \mathrm{off}_{\geqslant\mathrm{r}}(A) = \overline{\mathrm{off}_{\geqslant\mathrm{r}}(A)}.
\end{aligned}$$

Proof 1. The inclusion $\mathrm{off}_{>\mathrm{r}}(A)^{\circ} \subseteq \mathring{\overline{\mathrm{off}_{>\mathrm{r}}(A)}}$ follows from the obvious inclusion $\mathrm{off}_{>\mathrm{r}}(A) \subseteq \overline{\mathrm{off}_{>\mathrm{r}}(A)}$ by the monotony of the interior operator.

In the same way $\mathring{\overline{\mathrm{off}_{>\mathrm{r}}(A)}} \subseteq \mathrm{off}_{\geqslant\mathrm{r}}(A)^{\circ}$ follows from $\overline{\mathrm{off}_{>\mathrm{r}}(A)} \subseteq \mathrm{off}_{\geqslant\mathrm{r}}(A)$ shown below.

Since $\mathrm{off}_{\geqslant\mathrm{r}}(A)$ is closed by Lemma 5.3, we have $\mathrm{off}_{\geqslant\mathrm{r}}(A)^{\circ} = \mathring{\overline{\mathrm{off}_{\geqslant\mathrm{r}}(A)}}$.

We show $\mathrm{off}_{\geqslant\mathrm{r}}(A)^{\circ} \subseteq \mathrm{off}_{>\mathrm{r}}(\overline{A})$: Let $\mathbf{x} \in \mathrm{off}_{\geqslant\mathrm{r}}(A)^{\circ}$, and let $\mathbf{y} \in \overline{A}$. Assume for a contradiction that $d(\mathbf{x},\mathbf{y}) \leqslant r$. Choose $0 < \varepsilon \in \mathbb{R}$ such that $\mathrm{B}(\mathbf{x},\varepsilon) \subseteq \mathrm{off}_{\geqslant\mathrm{r}}(A)$. Pick $\mathbf{z}$ on the line segment $\overline{\mathbf{xy}}$ such that $d(\mathbf{x},\mathbf{z}) = \frac{\varepsilon}{2}$, and pick $\mathbf{a} \in \mathrm{B}(\mathbf{y},\frac{\varepsilon}{2}) \cap A$. Then $d(\mathbf{z},\mathbf{a}) \leqslant d(\mathbf{z},\mathbf{y}) + d(\mathbf{y},\mathbf{a}) < r - \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = r$, but $\mathbf{z} \in \mathrm{off}_{\geqslant\mathrm{r}}(A)$, a contradiction.

$\mathrm{off}_{>\mathrm{r}}(\overline{A}) \subseteq \mathrm{off}_{>\mathrm{r}}(A) \subseteq \overline{\mathrm{off}_{>\mathrm{r}}(A)}$ is obvious.

We show the contrapositive of $\mathrm{off}_{>\mathrm{r}}(A) \subseteq \mathrm{off}_{\geqslant\mathrm{r}}(\overline{A})$. This implies that $\overline{\mathrm{off}_{>\mathrm{r}}(A)} \subseteq \mathrm{off}_{\geqslant\mathrm{r}}(\overline{A})$ since $\mathrm{off}_{\geqslant\mathrm{r}}(\overline{A})$ is closed: Let $\mathbf{x} \notin \mathrm{off}_{\geqslant\mathrm{r}}(\overline{A})$, say $\mathbf{y} \in \overline{A}$ and $0 < \varepsilon \in \mathbb{R}$ such that $d(\mathbf{x},\mathbf{y}) = r - \varepsilon < r$. Choose $\mathbf{a} \in \mathrm{B}(\mathbf{y},\varepsilon) \cap A$. Then $d(\mathbf{x},\mathbf{a}) \leqslant d(\mathbf{x},\mathbf{y}) + d(\mathbf{y},\mathbf{a}) < r - \varepsilon + \varepsilon = r$, i.e., $\mathbf{x} \notin \mathrm{off}_{>\mathrm{r}}(A)$.

$\mathrm{off}_{\geqslant\mathrm{r}}(\overline{A}) \subseteq \mathrm{off}_{\geqslant\mathrm{r}}(A) \subseteq \overline{\mathrm{off}_{\geqslant\mathrm{r}}(A)}$ is again obvious.

We know by Lemma 5.3 that $\mathrm{off}_{>\mathrm{r}}(\overline{A}) = \mathrm{off}_{>\mathrm{r}}(\overline{A})^{\circ}$. From $\mathrm{off}_{>\mathrm{r}}(\overline{A}) \subseteq \mathrm{off}_{>\mathrm{r}}(A)$ proved above, it follows by the monotony of the interior operator that $\mathrm{off}_{>\mathrm{r}}(\overline{A})^{\circ} \subseteq \mathrm{off}_{>\mathrm{r}}(A)^{\circ}$; together $\mathrm{off}_{>\mathrm{r}}(\overline{A}) \subseteq \mathrm{off}_{>\mathrm{r}}(A)^{\circ}$.

We show the contrapositive of $\mathrm{off}_{\geqslant\mathrm{r}}(A) \subseteq \mathrm{off}_{\geqslant\mathrm{r}}(\overline{A})$: Let $\mathbf{x} \notin \mathrm{off}_{\geqslant\mathrm{r}}(\overline{A})$, say $\mathbf{y} \in \overline{A}$, $0 < \varepsilon \in \mathbb{R}$ such that $d(\mathbf{x},\mathbf{y}) = r - \varepsilon < r$. Then for $\mathbf{a} \in \mathrm{B}(\mathbf{y},\varepsilon) \cap A$ we have $d(\mathbf{x},\mathbf{a}) \leqslant d(\mathbf{x},\mathbf{y}) + d(\mathbf{y},\mathbf{a}) < r - \varepsilon + \varepsilon = r$, i.e., $\mathbf{x} \notin \mathrm{off}_{\geqslant\mathrm{r}}(A)$.

We know that $\mathrm{off}_{\geqslant\mathrm{r}}(A) = \overline{\mathrm{off}_{\geqslant\mathrm{r}}(A)}$ by Lemma 5.3.

From $\mathrm{off}_{>\mathrm{r}}(A)^{\circ} = \mathrm{off}_{\geqslant\mathrm{r}}(A)^{\circ}$ it follows that $\mathrm{off}_{>\mathrm{r}}(A)^{*} = \mathrm{off}_{\geqslant\mathrm{r}}(A)^{*}$.

2. $\mathrm{off}_{>\mathrm{r}}(A)^{\circ} \subseteq \mathrm{off}_{>\mathrm{r}}(A)^{*}$ is obvious.

Since $\mathrm{off}_{>\mathrm{r}}(A)$ is closed by Lemma 5.3, we may conclude that $\mathrm{off}_{>\mathrm{r}}(A)^{*} \subseteq \mathrm{off}_{>\mathrm{r}}(A)$.

We show that $\mathrm{off}_{\geqslant\mathrm{r}}(A) \subseteq \mathrm{off}_{>\mathrm{r}}(A)$: Let $\mathbf{x} \in \mathrm{off}_{\geqslant\mathrm{r}}(A)$, and let $\mathbf{a} \in A$. Then $d(\mathbf{x},\mathbf{a}) \geqslant r$. Assume for a contradiction that $d(\mathbf{x},\mathbf{a}) = r$. Pick $0 < \varepsilon \in \mathbb{R}$ such that $\mathrm{B}(\mathbf{a},\varepsilon) \subseteq A$. Let $\mathbf{z}$ be a point on the line segment $\overline{\mathbf{xa}}$ with $d(\mathbf{z},\mathbf{a}) = \frac{\varepsilon}{2}$. Then $\mathbf{z} \in A$ but $d(\mathbf{x},\mathbf{z}) = r - \frac{\varepsilon}{2} < r$, a contradiction.

3. If A is already closed, then obviously $\mathrm{off}_{>\mathrm{r}}(\overline{A}) = \mathrm{off}_{>\mathrm{r}}(A)$.

We know by Lemma 5.3 that $\mathrm{off}_{>\mathrm{r}}(A)$ is open and thus $\mathrm{off}_{>\mathrm{r}}(A)^{*} = \overline{\mathrm{off}_{>\mathrm{r}}(A)}$. □

Table 5.1: All inclusions in Lemma 5.4 are proper in general

A	$\mathrm{off}_{>1}(\overline{A})$	$\mathrm{off}_{>1}(A)^*$	$\mathrm{off}_{>1}(A)$	$\overline{\mathrm{off}_{>1}(A)}$	$\mathrm{off}_{\geqslant 1}(\overline{A})$
$]-\infty, 0[$	$]1, \infty[$	$[1, \infty[$			
$\sim [-1, 1]$		$\emptyset$	$\{0\}$		
$]-\infty, 0]$			$]1, \infty[$	$[1, \infty[$	
$\sim]-1, 1[$				$\emptyset$	$\{0\}$

A similar result for $<$-offsets and $\leqslant$-offsets is derived in the following way: We take the complement of each offset in Lemma 5.4, which inverts all inclusion relations. Then the complements are encoded into offsets using rules already discussed above together with the topological rules

$$\sim \overline{X} = (\sim X)^\circ \quad \text{and} \quad \sim (X^\circ) = \overline{\sim X}.$$

Corollary 5.5 1. Let $A \subseteq \mathbb{R}^n$, and let $0 < r \in \mathbb{R}$. Then

$$\begin{aligned}
\mathrm{off}_{<\mathrm{r}}(A)^\circ &= \mathrm{off}_{<\mathrm{r}}(A) = \mathrm{off}_{<\mathrm{r}}(\overline{A}) \\
&\subseteq \mathrm{off}_{\leqslant\mathrm{r}}(A)^\circ \\
&\subseteq \mathrm{off}_{\leqslant\mathrm{r}}(A) \\
&\subseteq \mathrm{off}_{\leqslant\mathrm{r}}(\overline{A}) = \mathrm{off}_{<\mathrm{r}}(A)^* = \overline{\mathrm{off}_{<\mathrm{r}}(A)} = \mathrm{off}_{\leqslant\mathrm{r}}(A)^* = \overline{\mathrm{off}_{\leqslant\mathrm{r}}(A)}, \\
\overset{\circ}{\overline{\mathrm{off}_{<\mathrm{r}}(A)}} &= \overset{\circ}{\overline{\mathrm{off}_{\leqslant\mathrm{r}}(A)}}.
\end{aligned}$$

2. If A is open, then

$$\begin{aligned}
\mathrm{off}_{<\mathrm{r}}(A)^\circ &= \mathrm{off}_{<\mathrm{r}}(A) = \mathrm{off}_{<\mathrm{r}}(\overline{A}) = \mathrm{off}_{\leqslant\mathrm{r}}(A)^\circ = \mathrm{off}_{\leqslant\mathrm{r}}(A) \\
&\subseteq \overset{\circ}{\overline{\mathrm{off}_{<\mathrm{r}}(A)}} = \overset{\circ}{\overline{\mathrm{off}_{\leqslant\mathrm{r}}(A)}} \\
&\subseteq \mathrm{off}_{\leqslant\mathrm{r}}(\overline{A}) = \mathrm{off}_{<\mathrm{r}}(A)^* = \overline{\mathrm{off}_{<\mathrm{r}}(A)} = \mathrm{off}_{\leqslant\mathrm{r}}(A)^* = \overline{\mathrm{off}_{\leqslant\mathrm{r}}(A)}.
\end{aligned}$$

3. If A is closed, then

$$\begin{aligned}
\mathrm{off}_{<\mathrm{r}}(A)^\circ &= \mathrm{off}_{<\mathrm{r}}(A) = \mathrm{off}_{<\mathrm{r}}(\overline{A}) \\
&\subseteq \overset{\circ}{\overline{\mathrm{off}_{<\mathrm{r}}(A)}} = \mathrm{off}_{\leqslant\mathrm{r}}(A)^\circ = \overset{\circ}{\overline{\mathrm{off}_{\leqslant\mathrm{r}}(A)}} \\
&\subseteq \mathrm{off}_{\leqslant\mathrm{r}}(A) = \mathrm{off}_{\leqslant\mathrm{r}}(\overline{A}) = \mathrm{off}_{<\mathrm{r}}(A)^* = \overline{\mathrm{off}_{<\mathrm{r}}(A)} = \mathrm{off}_{\leqslant\mathrm{r}}(A)^* \\
&= \overline{\mathrm{off}_{\leqslant\mathrm{r}}(A)}.
\end{aligned}$$

Table 5.1 illustrates by example that already in $\mathbb{R}$ all inclusions in Lemma 5.4 are proper in general. The examples can be transformed into corresponding examples for $<$/$\leqslant$-offsets by taking complements. Replacing the intervals by corresponding balls, they can be lifted to any dimension.

Since all the inclusions are proper, it follows that $\mathrm{off}_{>\mathrm{r}}(A)^*$ cannot be positioned into the given chain of inclusions for the general case. The following example illustrates this fact once more: We set

$$A =]-\infty, 0] \cup [3, 4[\cup]6, \infty[\,,$$

which is neither open nor closed. Then $\mathrm{off}_{>1}(A)^* = [1,2]$ is neither a subset nor a superset of $\mathrm{off}_{>1}(A) = \,]1,2[\, \cup \{5\}$. The same observation applies to $\mathrm{off}_{\geqslant \mathrm{r}}(A)^* = \mathrm{off}_{>\mathrm{r}}(A)^*$.

5.4 Blendings

Among the numerous types of blendings of solids that have been discussed in the literature [HH86, VMV94] we consider global fixed-radius rolling ball roundings in the classification of Vida et al. [VMV94]. We will define this notion precisely below.

Our blending operations are by definition *global* on the involved objects. In practice, however, one often wishes to perform *local* blendings, i.e., blendings restricted to certain edges or corners. This can be achieved by equivalently modifying the boolean structure of the quantifier-free representation of a given solid in such a way that the target parts are isolated. Then the operations are applied only to the subformulas describing these parts. Afterwards, the boolean structure can once more be rearranged to a more canonical form.

5.4.1 Roundings from Inside

Definition 5.6 Let A be a closed set in $\mathbb{R}^n$, and let $0 \leqslant r \in \mathbb{R}$. Then the *$r$-rounding from inside* of A is the union of all closed balls completely contained in A:

$$\mathrm{round}_r(A) = \bigcup_{\mathrm{B}[\mathbf{a},r]\subseteq A} \mathrm{B}[\mathbf{a},r] = \{\, \mathbf{x} \in \mathbb{R}^n \mid \exists(\mathbf{a} \in A) : \mathbf{x} \in \mathrm{B}[\mathbf{a},r] \subseteq A \,\}.$$

Accordingly, A is called *r-rounded from inside* if $A = \mathrm{round}_r(A)$.

From the definition it follows that $\mathrm{round}_r(A) \subseteq A$, and for $A \subseteq B$ we may conclude that $\mathrm{round}_r(A) \subseteq \mathrm{round}_r(B)$. The r-rounding from inside of a semialgebraic set is again semialgebraic:

$$\begin{aligned}\mathrm{round}_r\big(S\big(\varphi(\mathbf{u})\big)\big) \quad = \quad & S\big(\exists x_1 \exists x_2 \exists x_3 (\delta_{\leqslant}(\mathbf{x},\mathbf{u},r) \wedge \\ & \forall y_1 \forall y_2 \forall y_3 (\delta_{\leqslant}(\mathbf{x},\mathbf{y},r) \longrightarrow \varphi[\mathbf{u}/\mathbf{y}]))\big).\end{aligned}$$

The following theorem characterizes roundings from inside in terms of offsets. From this characterization, it follows that r-roundings from inside are, though *infinite* unions of closed balls, again closed. In fact they are even regular closed.

Theorem 5.7 *Let $A \subseteq \mathbb{R}^n$ be closed, and let $0 \leqslant r \in \mathbb{R}$. Then*

$$\mathrm{round}_r(A) = \mathrm{off}_{\leqslant \mathrm{r}}\mathrm{off}_{>\mathrm{r}}(\sim A) = \mathrm{off}_{\leqslant \mathrm{r}}\mathrm{off}_{\geqslant \mathrm{r}}(\overline{\sim A}).$$

In particular, $\mathrm{round}_r(A)$ *is regular closed.*

Proof Let $\mathbf{x} \in \mathrm{round}_r(A)$, and pick $\mathbf{y} \in A$ with $\mathbf{x} \in \mathrm{B}[\mathbf{y},r] \subseteq A$. Then $\mathbf{y} \in \mathrm{off}_{>\mathrm{r}}(\sim A)$ and $d(\mathbf{x},\mathbf{y}) \leqslant r$, and so $\mathbf{x} \in \mathrm{off}_{\leqslant \mathrm{r}}\mathrm{off}_{>\mathrm{r}}(\sim A)$.

Conversely, let $\mathbf{x} \in \mathrm{off}_{\leqslant \mathrm{r}}\mathrm{off}_{>\mathrm{r}}(\sim A)$ and pick $\mathbf{y} \in \mathrm{off}_{>\mathrm{r}}(\sim A)$ with $d(\mathbf{x},\mathbf{y}) \leqslant r$. Then $\mathbf{x} \in \mathrm{B}[\mathbf{y},r] \subseteq A$, i.e. $\mathbf{x} \in \mathrm{round}_r(A)$. The rest follows by Lemma 5.4 and Corollary 5.5. $\square$

Let $A \subseteq \mathbb{R}^n$ be closed. We obviously have $\mathrm{round}_0(A) = A$. For $0 < r \in \mathbb{R}$ it follows immediately from Theorem 5.7 that $\mathrm{round}_r(A) \subseteq A^*$, and further

$$\bigcup_{0<r\in\mathbb{R}} \mathrm{round}_r(A) \subseteq A^*.$$

This inclusion relation is in general a proper one: Consider the first quadrant $A = \{\, (x_1, x_2) \in \mathbb{R}^2 \mid x_1 \geqslant 0 \text{ and } x_2 \geqslant 0 \,\}$ of real 2-space. It is easily seen to be regular closed, but for all $0 < r \in \mathbb{R}$ the origin $(0,0)$ is not in $\mathrm{round}_r(A)$.

We eventually have to clarify whether r-roundings are actually r-rounded. They are, and moreover they are as close to the original set as possible:

Lemma 5.8 Let $A \subseteq \mathbb{R}^n$ be closed, and let $0 \leqslant r \in \mathbb{R}$. Then $\mathrm{round}_r(A)$ is the largest subset of A that is r-rounded.

Proof We are allowed to form $\mathrm{round}_r(\mathrm{round}_r(A))$, because $\mathrm{round}_r(A)$ is closed by Theorem 5.7. Since $\mathrm{round}_r(A)$ itself is already defined as a union of closed r-balls, it follows that

$$\mathrm{round}_r\big(\mathrm{round}_r(A)\big) = \bigcup \{\, \mathrm{B}[\mathbf{b}, r] \mid \mathrm{B}[\mathbf{b}, r] \subseteq \mathrm{round}_r(A) \,\} = \mathrm{round}_r(A),$$

i.e., $\mathrm{round}_r(A)$ is r-rounded from inside. Let $B \subseteq A$ be r-rounded from inside. Then $B = \mathrm{round}_r(B) \subseteq \mathrm{round}_r(A)$. □

Finally, we examine the interplay between roundings from inside and boolean operations.

Lemma 5.9 Let A, $B \subseteq \mathbb{R}^n$ be closed, and let $0 < r \in \mathbb{R}$. Then the following assertions hold:

1. $\mathrm{round}_r(A) \cup \mathrm{round}_r(B) \subseteq \mathrm{round}_r(A \cup B)$
2. $\mathrm{round}_r(A) \cap \mathrm{round}_r(B) \supseteq \mathrm{round}_r(A \cap B)$
3. If both A and B are r-rounded from inside, then so is $A \cup B$.

Proof

1. Let $\mathbf{x} \in \mathrm{round}_r(A) \cup \mathrm{round}_r(B)$, say $\mathbf{x} \in \mathrm{round}_r(A)$. Then there exists $\mathbf{y} \in A$ such that $\mathbf{x} \in \mathrm{B}[\mathbf{y}, r] \subseteq A \subseteq A \cup B$.
2. Let $\mathbf{x} \in \mathrm{round}_r(A \cap B)$. Then there exists $\mathbf{y} \in A \cap B$ with $\mathbf{x} \in \mathrm{B}[\mathbf{y}, r] \subseteq A \cap B \subseteq A, B$.
3. Let $\mathbf{x} \in A \cup B$, say $\mathbf{x} \in A$. Then there exists $\mathbf{y} \in A$ with $\mathbf{x} \in \mathrm{B}[\mathbf{y}, r] \subseteq A \subseteq A \cup B$. □

The inclusions in Lemma 5.9 above are proper in general: Consider real 2-space and fix $r = 1$. For the first inclusion consider two adjacent squares of side lengths 2. For the second inclusion consider two overlapping closed unit disks centered at $(0,0)$ and $(1,0)$. This latter example shows also that the intersection of r-rounded sets is not necessarily r-rounded.

Example 5.10 (First octant) We compute the r-rounding from inside of the first octant $O_1 = S(u_1 \geqslant 0 \wedge u_2 \geqslant 0 \wedge u_3 \geqslant 0)$. For this, we compute the $\leqslant$-offset of the obvious quantifier-free description $u_1 - r \geqslant 0 \wedge u_2 - r \geqslant 0 \wedge u_3 - r \geqslant 0$ of the $>$-offset of the complement of O_1.

$$\exists x_1 \exists x_2 \exists x_3 \Big(\bigwedge_{i=1}^{3} x_i - r \geqslant 0 \wedge \sum_{i=1}^{3} (u_i - x_i)^2 \leqslant r^2 \Big).$$

We automatically obtain the following equivalent quantifier-free description of the parametric r-rounding from inside. It contains 19 atomic formulas:

$$\begin{aligned}
\mathrm{round}_r(O_1) \quad \equiv \quad & S\big(2r^2 - 2ru_1 - 2ru_2 - 2ru_3 + u_1^2 + u_2^2 + u_3^2 \leqslant 0 \vee \\
& (r - u_1 \leqslant 0 \wedge r - u_2 \leqslant 0 \wedge r - u_3 \leqslant 0) \vee \\
& (2ru_1 - u_1^2 \geqslant 0 \wedge r - u_2 \leqslant 0 \wedge r - u_3 \leqslant 0) \vee \\
& (2ru_2 - u_2^2 \geqslant 0 \wedge r - u_1 \leqslant 0 \wedge r - u_3 \leqslant 0) \vee \\
& (r^2 - 2ru_1 - 2ru_2 + u_1^2 + u_2^2 \leqslant 0 \wedge r - u_3 \leqslant 0) \vee \\
& (2ru_3 - u_3^2 \geqslant 0 \wedge r - u_1 \leqslant 0 \wedge r - u_2 \leqslant 0) \vee \\
& (r^2 - 2ru_1 - 2ru_3 + u_1^2 + u_3^2 \leqslant 0 \wedge r - u_2 \leqslant 0) \vee \\
& (r^2 - 2ru_2 - 2ru_3 + u_2^2 + u_3^2 \leqslant 0 \wedge r - u_1 \leqslant 0)\big).
\end{aligned}$$

This computation takes 60 ms. The first inequality in the above result formula is the ball

$$\sum_{i=1}^{3} (u_i - r)^2 \leqslant r^2.$$

at (r, r, r) with radius r including the rounded corner.

The conjunction following this inequality is the first octant shifted from the origin to the center (r, r, r) of our ball. The next conjunction can be rewritten as $(u_1 - r)^2 \leqslant r^2 \wedge r \leqslant u_2 \wedge r \leqslant u_3$, which for positive r amounts to

$$0 \leqslant u_1 \leqslant 2r \wedge r \leqslant u_2 \wedge r \leqslant u_3,$$

i.e. a slice of width $2r$ parallel to the Y-Z-plane. The two other conjunctions that look similar describe corresponding slices in the other coordinate directions.

The conjunction $r^2 - 2ru_1 - 2ru_2 + u_1^2 + u_2^2 \leqslant 0 \wedge r - u_3 \leqslant 0$ describes the semiinfinite cylinder

$$\sum_{i=1}^{2} (u_i - r)^2 \leqslant r^2 \wedge u_3 \geqslant r$$

continuing our ball to infinity in direction u_3. Again, the two similar conjunctions describe the corresponding cylinders in direction u_1 and u_2, respectively. □

Example 5.11 (Full box) Rounding the full box

$$B = S(0 \leqslant u_1 \leqslant 2 \wedge 0 \leqslant u_2 \leqslant 3 \wedge 0 \leqslant u_3 \leqslant 4)$$

yields within 3 s a quantifier-free description of $\mathrm{round}_r(B)$ containing 106 atomic formulas. Substituting into this description $r = 2$ and simplifying yields "false" describing the empty set. Substituting $r = 1$ yields a description of the 1-rounded box, which contains only 50 atomic formulas. The time for substitution and simplification can be neglected here (10 ms and 40 ms of CPU time, respectively). □

5.4.2 Roundings from Outside

Recall that we have defined roundings from inside of a solid as the union of all closed balls contained in it. This removes certain parts of the solid such as convex edges and corners. Roundings from outside, in contrast, will add certain parts to the solid thus smoothing concave edges and corners. Roughly speaking, the rounding from outside of a solid is obtained by rounding its complement from inside. In other words, the solid rounded from outside is the complement of all balls not intersecting it. Note, however, that both the input solid and the result of the rounding operation should be closed, i.e. the corresponding complements are open. We hence operate with open balls here, and, as a consequence, the notion of rounding from outside is not exactly dual to that of rounding from inside.

Definition 5.12 Let A be a closed set in $\mathbb{R}^n$, and let $0 \leqslant r \in \mathbb{R}$. Then the *$r$-rounding from outside* of A is defined as the following set:

$$\begin{aligned} \mathrm{Round}_r(A) &= \bigcap \{ \sim \mathrm{B}(\mathbf{x}, r) \mid \mathrm{B}(\mathbf{x}, r) \cap A = \emptyset \} \\ &= \{ \mathbf{x} \in \mathbb{R}^n \mid \forall (\mathbf{y} \in \mathbb{R}^n) : \mathrm{B}(\mathbf{y}, r) \cap A = \emptyset \Longrightarrow \mathbf{x} \notin \mathrm{B}(\mathbf{y}, r) \}. \end{aligned}$$

A is called *r-rounded from outside* if $A = \mathrm{Round}_r(A)$.

From the definition it follows that $A \subseteq \mathrm{Round}_r(A)$, and for $A \subseteq B$, we can conclude that $\mathrm{Round}_r(A) \subseteq \mathrm{Round}_r(B)$. As an intersection of closed sets $\mathrm{Round}_r(A)$ is closed. Roundings from outside of semialgebraic sets are semialgebraic:

$$\begin{aligned} \mathrm{Round}_r\big(S(\varphi(\mathbf{u}))\big) &= S\big(\forall x_1 \forall x_2 \forall x_3 (\neg \exists y_1 \exists y_2 \exists y_3 (\delta_{<}(\mathbf{x}, \mathbf{y}, r) \wedge \varphi(\mathbf{y})) \\ &\quad \longrightarrow \delta_{\geqslant}(\mathbf{x}, \mathbf{u}, r))\big). \end{aligned}$$

Similar to roundings from inside, roundings from outside can be characterized as iterated offset computations:

Theorem 5.13 *Let $A \subseteq \mathbb{R}^n$ be closed, and let $0 \leqslant r \in \mathbb{R}$. Then*

$$\mathrm{Round}_r(A) = \mathrm{off}_{\geqslant \mathrm{r}} \mathrm{off}_{\geqslant \mathrm{r}}(A).$$

Proof Let $\mathbf{x} \in \mathrm{Round}_r(A)$, and let $\mathbf{y} \in \mathrm{off}_{\geqslant \mathrm{r}}(A)$. Then $\mathrm{B}(\mathbf{y}, r) \cap A = \emptyset$, thus $\mathbf{x} \notin \mathrm{B}(\mathbf{y}, r)$, and so $d(\mathbf{x}, \mathbf{y}) \geqslant r$. This shows that $\mathrm{Round}_r(A) \subseteq \mathrm{off}_{\geqslant \mathrm{r}} \mathrm{off}_{\geqslant \mathrm{r}}(A)$.

Conversely, let $\mathbf{x} \in \mathrm{off}_{\geqslant \mathrm{r}} \mathrm{off}_{\geqslant \mathrm{r}}(A)$, and let $\mathbf{y} \in \mathbb{R}^n$ be such that $\mathrm{B}(\mathbf{y}, r) \cap A = \emptyset$. Then $\mathbf{y} \in \mathrm{off}_{\geqslant \mathrm{r}}(A)$, thus $d(\mathbf{x}, \mathbf{y}) \geqslant r$, and so $\mathbf{x} \notin \mathrm{B}(\mathbf{y}, r)$. This shows the converse inclusion. □

Also, rounding from outside yields rounded sets as one would expect:

Lemma 5.14 Let $A \subseteq \mathbb{R}^n$ be closed, and let $0 \leqslant r \in \mathbb{R}$. Then $\mathrm{Round}_r(A)$ is the smallest superset of A that is r-rounded from outside.

Proof We already know that $\mathrm{Round}_r(A) \subseteq \mathrm{Round}_r\big(\mathrm{Round}_r(A)\big)$. Let, conversely, $\mathbf{x} \in \mathrm{Round}_r\big(\mathrm{Round}_r(A)\big)$, and let furthermore $\mathbf{y} \in \mathbb{R}^n$ be such that $\mathrm{B}(\mathbf{y}, r) \cap A = \emptyset$. Then for all $\mathbf{z} \in \mathrm{Round}_r(A)$ we have $\mathbf{z} \notin \mathrm{B}(\mathbf{y}, r)$, i.e.,

$\mathrm{B}(\mathbf{y},r) \cap \mathrm{Round}_r(A) = \emptyset$. Hence $\mathbf{x} \notin \mathrm{B}(\mathbf{y},r)$, and we have shown that $\mathrm{Round}_r\big(\mathrm{Round}_r(A)\big) \subseteq \mathrm{Round}_r(A)$. Let $B \supseteq A$ be r-rounded from outside. Then $\mathrm{Round}_r(A) \subseteq \mathrm{Round}_r(B) = B$. $\square$

Compared to roundings from inside, unions and intersections of roundings from outside obey the opposite inclusions. Accordingly, roundings from outside are closed under intersection instead of union.

Lemma 5.15 Let A, $B \subseteq \mathbb{R}^n$ be closed, and let $0 < r \in \mathbb{R}$. Then the following assertions hold:

1. $\mathrm{Round}_r(A) \cup \mathrm{Round}_r(B) \subseteq \mathrm{Round}_r(A \cup B)$
2. $\mathrm{Round}_r(A) \cap \mathrm{Round}_r(B) \supseteq \mathrm{Round}_r(A \cap B)$
3. If both A and B are r-rounded from outside, then so is $A \cap B$.

Proof 1. Let $\mathbf{x} \in \mathrm{Round}_r(A) \cup \mathrm{Round}_r(B)$, and let $\mathbf{z} \in \mathbb{R}^n$ with $\mathrm{B}(\mathbf{z},r) \subseteq \sim(A \cup B) = \sim A \cap \sim B$. Then $\mathbf{x} \notin \mathrm{B}(\mathbf{z},r)$.

2. Let $\mathbf{x} \in \mathrm{Round}_r(A \cap B)$, and let $\mathbf{z} \in \mathbb{R}^n$ such that $\mathrm{B}(\mathbf{z},r) \cap A = \emptyset$, respectively $\mathrm{B}(\mathbf{z},r) \cap B = \emptyset$. Then $\mathrm{B}(\mathbf{z},r) \cap (A \cap B) = \emptyset$, and thus $\mathbf{x} \notin \mathrm{B}(\mathbf{z},r)$.

3. Let $\mathbf{x} \in \mathrm{Round}_r(A \cap B)$, and assume $\mathbf{x} \notin A \cap B$, say $\mathbf{x} \notin A$. Then $\mathbf{x} \notin \mathrm{Round}_r(A)$, and so there exists $\mathbf{z} \in \mathbb{R}^n$ with $\mathbf{x} \in \mathrm{B}(\mathbf{z},r) \subseteq \sim A \subseteq \sim A \cup \sim B = \sim(A \cap B)$, a contradiction. $\square$

Again, the inclusions are proper in general already in real 2-space: For the first inclusion, consider the two half spaces given by $x_1 \leqslant 0$ and by $x_2 \leqslant 0$. For the second inclusion, consider the complements of the first and of the second quadrant. The first example shows also that roundings from outside are not closed under union.

There appear to be no relations between r-roundings from outside and regular closed or open sets. In fact, there is a regular closed, simply connected set $A \subseteq \mathbb{R}^3$ such that $\mathrm{Round}_r(A)$ is disconnected and not regular closed.

The relation between r-roundings from inside and r-roundings from outside works via the operation of taking closures of complements.

Theorem 5.16 *Let $A \subseteq \mathbb{R}^n$ be closed, and let $0 < r \in \mathbb{R}$. Then the following assertions hold:*

1. $\mathrm{round}_r(A) = \overline{\sim \mathrm{Round}_r(\overline{\sim A})}$
2. $\mathrm{Round}_r(A^*)^* = \overline{\sim \mathrm{round}_r(\overline{\sim A})}$.

Proof 1. $\mathrm{round}_r(A) = \mathrm{off}_{\leqslant \mathrm{r}}\mathrm{off}_{\geqslant \mathrm{r}}(\overline{\sim A}) = \overline{\mathrm{off}_{< \mathrm{r}}\mathrm{off}_{\geqslant \mathrm{r}}(\overline{\sim A})} = \overline{\sim \mathrm{off}_{\geqslant \mathrm{r}}\mathrm{off}_{\geqslant \mathrm{r}}(\overline{\sim A})} = \overline{\sim \mathrm{Round}_r(\overline{\sim A})}$

2. We rewrite $\overline{\sim \mathrm{round}_r(\overline{\sim A})}$ according to part (1), and transform it in the following way:

$$\overline{\sim \overline{\sim \mathrm{Round}_r(\overline{\sim \overline{\sim A}})}} = \overline{\sim \overline{\sim \mathrm{Round}_r(\overline{\sim \sim (A^\circ)})}} = \overline{\sim \overline{\sim \mathrm{Round}_r(A^*)}}.$$

By the same transformation we eventually obtain $\mathrm{Round}_r(A^*)^*$. $\square$

Example 5.17 (Closed complement of first octant) For the parametric r-rounding from outside of the closed complement of the first octant $S(u_1 \leqslant 0 \vee u_2 \leqslant 0 \vee u_3 \leqslant 0)$ we obtain, similar to rounding the octant itself from inside in Example 5.10, 19 atomic formulas in less than one second. □

5.4.3 Comparison to Other Work

Rossignac and Requicha [RR86] define solids as regular closed sets $A \subseteq \mathbb{R}^n$. Rounding is defined in terms of the offsetting operations of *expanding* $A \uparrow^* r$ and (regularized) *shrinking* $A \downarrow^* r$. The notion of expanding is exactly our $\leqslant$-r-offset. Shrinking is defined in terms of expanding:

$$A \uparrow^* r = \mathrm{off}_{\leqslant \mathrm{r}}(A), \quad A \downarrow^* r = \overline{\sim (\overline{\sim A} \uparrow^* r)}.$$

Recall from Corollary 5.5 that $\mathrm{off}_{\leqslant \mathrm{r}}(A)$ is regular closed for closed A. In the definition of shrinking, both applications of the closure operator are to open sets. The closures obtained are thus in fact regular closures, i.e., the result of shrinking applied to closed sets is regular closed. We translate the definition of shrinking to our framework:

$$A{\downarrow}^* r = \overline{\sim (\overline{\sim A} \uparrow^* r)} = \overline{\sim \mathrm{off}_{\leqslant \mathrm{r}}(\overline{\sim A})} = \overline{\mathrm{off}_{>\mathrm{r}}(\overline{\sim A})} = \mathrm{off}_{>\mathrm{r}}(\overline{\sim A})^* = \mathrm{off}_{\geqslant \mathrm{r}}(\overline{\sim A})^*.$$

Rounding from inside and rounding from outside are defined by Rossignac and Requicha as

$$\mathrm{round}'_r(A) = (A \downarrow^* r) \uparrow^* r \quad \text{and} \quad \mathrm{Round}'_r(A) = (A \uparrow^* r) \downarrow^* r,$$

respectively. As we have seen above both expanding and shrinking yield regular closed sets, and so do hence rounding from inside and from outside.

This means that at least Round'_r must differ in some way from our notion Round_r: Consider, e.g., $\sim]{-r}, r[\subseteq \mathbb{R}$, which is regular closed. In our sense, it is r-rounded from outside:

$$\mathrm{Round}_r(\sim]{-r}, r[) = \mathrm{off}_{\geqslant \mathrm{r}} \mathrm{off}_{\geqslant \mathrm{r}}(\sim]{-r}, r[) = \mathrm{off}_{\geqslant \mathrm{r}}(\{0\}) = \sim]{-r}, r[\,.$$

In the framework of [RR86] we obtain in contrast:

$$\mathrm{Round}'_r(\sim]{-r}, r[) = \mathrm{off}_{\geqslant \mathrm{r}}\big(\overline{\sim \mathrm{off}_{\leqslant \mathrm{r}}(\sim]{-r}, r[)}\big)^* = \mathrm{off}_{\geqslant \mathrm{r}}(\emptyset)^* = \mathbb{R}.$$

Similar effects occur with roundings from inside. In our framework, closed balls of radius r are r-rounded from inside:

$$\mathrm{round}_r([-r, r]) = \mathrm{off}_{\leqslant \mathrm{r}} \mathrm{off}_{>\mathrm{r}}(\sim [-r, r]) = \mathrm{off}_{\leqslant \mathrm{r}}(\{0\}) = [-r, r].$$

Using round'_r we obtain in contrast the empty set as inner rounding:

$$\mathrm{round}'_r([-r, r]) = \mathrm{off}_{\leqslant \mathrm{r}}\big(\mathrm{off}_{\geqslant \mathrm{r}}(\overline{\sim [-r, r]})^*\big) = \mathrm{off}_{\leqslant \mathrm{r}}(\emptyset^*) = \mathrm{off}_{\leqslant \mathrm{r}}(\emptyset) = \emptyset.$$

The examples can be lifted to any dimension using corresponding balls instead of intervals. Moreover, for a dimension greater than or equal to two, $\sim \mathrm{B}(\mathbf{0}, r)$ used in the example for roundings from outside becomes connected. Note also that the 1-rounding of the full box in our Example 5.11 would be the empty set with the notion of regularized rounding.

The counterintuitive behavior of regularized roundings has already been observed and illustrated by Rossignac [Ros85] by means of examples similar to those given in Figure 5.1.

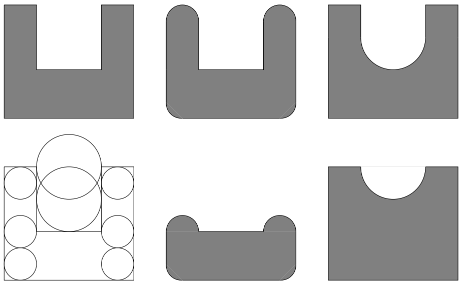

Figure 5.1: Given the solid top left, our notions of rounding from inside (center) and outside (right) lead to the results in the top row. Regularized blending yields the results in the bottom row for these limit cases.

Example 5.18 (Rossignac's limit case) We compute the rounding from inside of Rossignac's solid pictured in Figure 5.1. it is described by

$$\begin{aligned}&(0 \leqslant x_1 \leqslant 8 \wedge 0 \leqslant x_2 \leqslant 3) \vee \\ &\qquad (0 \leqslant x_1 \leqslant 2 \wedge 3 \leqslant x_2 \leqslant 7) \vee (6 \leqslant x_1 \leqslant 8 \wedge 3 \leqslant x_2 \leqslant 7).\end{aligned}$$

After 182 s we obtain a quantifier-free description containing 2063 atomic formulas. Substituting $r = 1$ and simplifying requires 16 s. The description then obtained for the 1-rounding contains 1298 atomic formulas. If we restrict to the 1-rounding from the beginning, we obtain a description with 1265 atomic formulas in 69 s. $\square$

Due to his observation, Rossignac has defined "non-regularized" variants of rounding from inside and from outside in terms of non-regularized shrinking and expanding:

$$\begin{aligned}A \uparrow r &= \mathrm{off}_{\leqslant \mathrm{r}}(A)\\ A \downarrow r &= \sim(\sim A \uparrow r) = \sim \mathrm{off}_{\leqslant \mathrm{r}}(\sim A) = \mathrm{off}_{>\mathrm{r}}(\sim A)\\ \mathrm{round}''_r(A) &= (A \downarrow r) \uparrow r = \mathrm{off}_{\leqslant \mathrm{r}}\mathrm{off}_{>\mathrm{r}}(\sim A) = \mathrm{round}_r(A)\\ \mathrm{Round}''_r(A) &= (A \uparrow r) \downarrow r = \mathrm{off}_{>\mathrm{r}}\big(\sim \mathrm{off}_{\leqslant \mathrm{r}}(A)\big) = \mathrm{off}_{>\mathrm{r}}\mathrm{off}_{>\mathrm{r}}(A).\end{aligned}$$

This rounding round''_r from inside amounts to exactly our notion. Non-regularized rounding from outside, in contrast, leads to similar problems as Round' above. Rossignac thus defines a "modified" rounding from outside:

$$\begin{aligned}\mathrm{Round}'''_r(A) &= \sim \mathrm{round}''_r((\sim A)^*)^*\\ &= \mathrm{off}_{>\mathrm{r}}\mathrm{off}_{>\mathrm{r}}(\sim(\sim A)^*)^* = \mathrm{off}_{>\mathrm{r}}\mathrm{off}_{>\mathrm{r}}\Big(\mathring{\overline{A}}\Big)^* = \mathrm{off}_{\geqslant \mathrm{r}}\mathrm{off}_{\geqslant \mathrm{r}}\Big(\mathring{\overline{A}}\Big)^*.\end{aligned}$$

For regular closed input A^*, we have $\mathring{\overline{A^*}} = A^{*\circ}$. It follows that Round''' is then the regular closure of our rounding from outside:

$$\mathrm{Round}'''_r(A^*) = \mathrm{off}_{\geqslant \mathrm{r}}\mathrm{off}_{\geqslant \mathrm{r}}(A^{*\circ})^* = \mathrm{off}_{\geqslant \mathrm{r}}\mathrm{off}_{\geqslant \mathrm{r}}(\overline{A^{*\circ}})^* = \mathrm{Round}_r(A^*)^*.$$

Both of the references discussed finally consider their initial versions round' and Round' the best solution, in order to avoid open sets even as intermediate results.

5.5 Obtaining B-REP Information

So far we have considered quantifier-free first-order formulas as an alternative to CSG. As already mentioned in the introduction to this chapter, there is a second commonly used representation in solid modeling, which is called *boundary representation* or B-REP for short. It represents solids as a finite collection of patches of their surfaces plus information on adjacency among these patches.

The solid modelers currently available are hybrid systems using both CSG and B-REP. The use of B-REP is mainly for the purpose of displaying objects, which currently cannot be satisfactorily solved on the basis of CSG. For this reason, we have to discuss how to obtain corresponding information from our quantifier-free descriptions.

In Section 4.2 on page 77, we have introduced the operator ∂ for obtaining the boundary of a geometric object. The computation of a DNF of the quantifier-free description obtained by the application of this operator will yield a disjunction of possibly overlapping patches, which is a major step towards the complete B-REP information. We apply this strategy to two toy examples. The first example looks very promising, while the second one unfolds the weakness of this naive approach for slightly more complicated input.

Example 5.19 (Box) We consider the solid rectangular box

$$\beta_1(\mathbf{u}) \equiv 0 \leqslant u_1 \leqslant 10 \wedge 0 \leqslant u_2 \leqslant 5 \wedge 0 \leqslant u_3 \leqslant 5.$$

A quantifier-free equivalent of $\partial(\beta_1)$ can be computed in 160 ms:

$$\begin{aligned}\partial(\beta_1)'(\mathbf{u}) \quad \equiv \quad & u_1 - 10 \leqslant 0 \wedge u_1 \geqslant 0 \wedge u_2 - 5 \leqslant 0 \wedge u_2 \geqslant 0 \wedge \\ & u_3 - 5 \leqslant 0 \wedge u_3 \geqslant 0 \wedge \\ & (u_1 - 10 = 0 \vee u_1 = 0 \vee u_2 - 5 = 0 \vee u_2 = 0 \vee \\ & u_3 - 5 = 0 \vee u_3 = 0).\end{aligned}$$

From this we obtain in 40 ms a Gröbner-simplified DNF describing exactly the six sides of the box:

$$\begin{aligned}&(u_1 - 10 \leqslant 0 \wedge u_1 \geqslant 0 \wedge u_2 - 5 \leqslant 0 \wedge u_2 \geqslant 0 \wedge u_3 = 0) \vee \\ &\quad (u_1 - 10 \leqslant 0 \wedge u_1 \geqslant 0 \wedge u_2 - 5 \leqslant 0 \wedge u_2 \geqslant 0 \wedge u_3 - 5 = 0) \vee \\ &\quad (u_1 - 10 \leqslant 0 \wedge u_1 \geqslant 0 \wedge u_2 = 0 \wedge u_3 - 5 \leqslant 0 \wedge u_3 \geqslant 0) \vee \\ &\quad (u_1 - 10 \leqslant 0 \wedge u_1 \geqslant 0 \wedge u_2 - 5 = 0 \wedge u_3 - 5 \leqslant 0 \wedge u_3 \geqslant 0) \vee \\ &\quad (u_1 = 0 \wedge u_2 - 5 \leqslant 0 \wedge u_2 \geqslant 0 \wedge u_3 - 5 \leqslant 0 \wedge u_3 \geqslant 0) \vee \\ &\quad (u_1 - 10 = 0 \wedge u_2 - 5 \leqslant 0 \wedge u_2 \geqslant 0 \wedge u_3 - 5 \leqslant 0 \wedge u_3 \geqslant 0). \quad \square\end{aligned}$$

Example 5.20 (Infinite cylinder) We next consider the infinite cylinder $\zeta \equiv u_1^2 + u_2^2 \leqslant 1$. After 13 s we obtain a quantifier-free description $\partial(\zeta)'$ of its boundary. This description contains 949 atomic formulas. Due to combinatorial explosion it is not possible to compute a DNF of $\partial(\zeta)'$. $\square$

It thus turns out that except for extremely simple structures we do not obtain a boundary description that is suitable for further processing. We have therefore developed a strategy, of course involving quantifier elimination, which has turned out to perform nicely on various examples of quite different style. It includes the following steps, which we are going to explain in detail:

- Computation of candidate patches.
- Isolation of proper surface patches.
- Gröbner simplification of the surface.
- Quantifier elimination recognized implication.

5.5.1 Computation of Candidate Patches

Consider the rectangular box described by the following quantifier-free formula:

$$\beta_1(\mathbf{u}) \equiv 0 \leqslant u_1 \leqslant 10 \wedge 0 \leqslant u_2 \leqslant 5 \wedge 0 \leqslant u_3 \leqslant 5.$$

Each of the patches of its surface $\Sigma(\beta_1)$ can be obtained by turning exactly one of the inequalities in β_1 into an equation:

$$\begin{aligned}\Sigma(\beta_1)(\mathbf{u}) \quad \equiv \quad & \{u_1 = 0 \wedge 0 \leqslant u_2 \leqslant 5 \wedge 0 \leqslant u_3 \leqslant 5, \\ & u_1 = 10 \wedge 0 \leqslant u_2 \leqslant 5 \wedge 0 \leqslant u_3 \leqslant 5, \\ & 0 \leqslant u_1 \leqslant 10 \wedge u_2 = 0 \wedge 0 \leqslant u_3 \leqslant 5, \\ & 0 \leqslant u_1 \leqslant 10 \wedge u_2 = 5 \wedge 0 \leqslant u_3 \leqslant 5, \\ & 0 \leqslant u_1 \leqslant 10 \wedge 0 \leqslant u_2 \leqslant 5 \wedge u_3 = 0, \\ & 0 \leqslant u_1 \leqslant 10 \wedge 0 \leqslant u_2 \leqslant 5 \wedge u_3 = 5\}.\end{aligned}$$

Unfortunately, the success of this technique depends on the nice description β_1. The same solid could be described as the union of two smaller boxes:

$$\begin{aligned}\beta_1'(\mathbf{u}) \quad \equiv \quad & (0 \leqslant u_1 \leqslant 10 \wedge 0 \leqslant u_2 \leqslant 3 \wedge 0 \leqslant u_3 \leqslant 5) \vee \\ & (0 \leqslant u_1 \leqslant 10 \wedge 3 \leqslant u_2 \leqslant 5 \wedge 0 \leqslant u_3 \leqslant 5).\end{aligned}$$

Then, our naive approach would yield the following:

1. Proper surface patches such as
$$0 \leqslant u_1 \leqslant 10 \wedge u_2 = 0 \wedge 0 \leqslant u_3 \leqslant 5.$$
2. Separate surface patches, which should be combined into one, such as
$$u_1 = 0 \wedge 0 \leqslant u_2 \leqslant 3 \wedge 0 \leqslant u_3 \leqslant 5, \quad u_1 = 0 \wedge 3 \leqslant u_2 \leqslant 5 \wedge 0 \leqslant u_3 \leqslant 5.$$
3. The plane segment separating the two subboxes. This is not part of the surface of β_1:
$$0 \leqslant u_1 \leqslant 10 \wedge u_2 = 3 \wedge 0 \leqslant u_3 \leqslant 5.$$

Note, however, that this result still contains the entire surface. This will generally happen when turning inequalities into equations.

More precisely, we proceed as follows: Let σ be the description of our solid. Assume that all right hand sides of the constraints are zero, and denote by L_σ the set of left hand side terms of the constraints in σ. Then we compute the *limiting* of sigma as follows

$$\lambda(\sigma) \equiv \bigvee_{t \in L_\sigma} t = 0 \wedge \sigma.$$

Lemma 5.21 Let $S(\sigma)$ be closed, then $\partial S(\sigma) \subseteq S\big(\lambda(\sigma)\big) \subseteq S(\sigma)$.

Proof To begin with, $S\big(\lambda(\sigma)\big) \subseteq S(\sigma)$ follows immediately from the fact that $S\big(\lambda(\sigma)\big)$ is a union of subsets of $S(\sigma)$.

For proving $\partial S(\sigma) \subseteq S\big(\lambda(\sigma)\big)$, we may wlog. assume that σ is of the form

$$\bigvee_{i \in I} \bigwedge_{j \in J_i} t_{ij}(\mathbf{u})\ \varrho_{ij}\ 0, \quad \varrho_{ij} \in \{>, \geqslant\}.$$

Let now $\mathbf{x} \in \partial S(\sigma)$. Since $S(\sigma)$ is closed, it follows that even $\mathbf{x} \in S(\sigma)$. Thus $\sigma(\mathbf{x})$ holds; more precisely, we can find some $i_0 \in I$ such that

$$\bigwedge_{j \in J_{i_0}} t_{i_0 j}(\mathbf{x})\ \varrho_{i_0 j}\ 0$$

holds. Assume for a contradiction that in fact even $\bigwedge_{j \in J_{i_0}} t_{i_0 j}(\mathbf{x}) > 0$ holds. All the $t_{i_0 j}(\mathbf{u}) > 0$ describe preimages of the open interval $]0, \infty[$ under the continuous polynomial functions given by the $t_{i_0 j}(\mathbf{u})$, i.e., open sets. Consequently, the finite intersection of all these sets is open. Since the interior of a set contains all open subsets, we have found

$$\mathbf{x} \in S\Big(\bigwedge_{j \in J_{i_0}} t_{i_0 j}(\mathbf{u}) > 0\Big) = \bigcap_{j \in J_{i_0}} S\big(t_{i_0 j}(\mathbf{u}) > 0\big) \subseteq S(\sigma)^\circ,$$

which contradicts $\mathbf{x} \in \partial S(\sigma) = \overline{S(\sigma)} \setminus S(\sigma)^\circ$. It follows that there must be at least one $j_0 \in J_{i_0}$ such that $\varrho_{i_0 j_0}$ equals "$\geqslant$" and such that actually $t_{i_0 j_0}(\mathbf{x}) = 0$ holds. This finally implies that

$$\mathbf{x} \in S\Big(t_{i_0 j_0}(\mathbf{u}) = 0 \wedge \bigwedge_{j \in J_{i_0}} t_{i_0 j}(\mathbf{u})\ \varrho_{i_0 j}\ 0\Big) \subseteq S\big(\lambda(\sigma)\big). \quad \square$$

Our algorithm turns $\lambda(\sigma)$ into DNF and applies Gröbner simplification, which in turn yields a simpler but equivalent DNF. The conjunctions contained in this DNF form our list Σ_0 of candidate patches. These candidates are reduced to the pure surface in the following step.

5.5.2 Isolation of Proper Surface Patches

For isolating the surface patches, we intersect the candidates with $\overline{\sim S(\sigma)}$. A description η' of this is obtained by applying quantifier elimination to

$$\eta(\mathbf{u}) \equiv \forall\varepsilon\big(\varepsilon > 0 \longrightarrow \exists x_1 \exists x_2 \exists x_3 (\delta_{\leqslant}(\mathbf{u}, \mathbf{x}, \varepsilon) \wedge \neg\sigma[\mathbf{u}/\mathbf{x}])\big).$$

The subformula $\delta_{\leqslant}(\mathbf{u}, \mathbf{x}, \varepsilon)$ describing the metric relation $d(\mathbf{u}, \mathbf{x}, \varepsilon) \leqslant \varepsilon$ has been defined in Section 4.4. For the success of our method, it is crucial that we use here the L_∞ metric, which can, in contrast to the Euclidean one, be coded linearly. Both metrics induce the same topology.

Next, all the candidate patches are conjunctively combined with η'. The results are put into CNF, and Gröbner simplification is applied, resulting in a simplified CNF. All the CNF's obtained this way form a list Σ_1 of proper surface patches. Some of them might be, however, empty or of dimension 1 or 0. The CNF computations performed here have empirically turned out crucial for the success of the DNF computation in the following step.

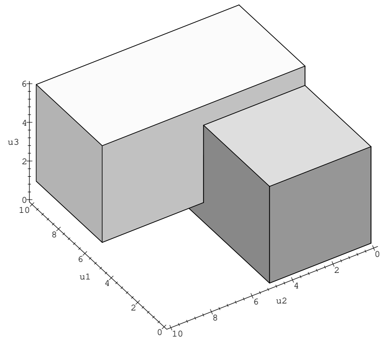

Figure 5.2: The two boxes of Example 5.22

5.5.3 Gröbner Simplification of the Surface

We now recombine the patches in Σ_1 to a formula $\bigvee \Sigma_1$, compute a DNF of this formula, and again apply Gröbner simplification, which keeps the formula in DNF. The result is split again to a list Σ_2 of patches, each of which is either a pure conjunction or a single atomic formula.

5.5.4 Quantifier Elimination Recognized Implication

We finally want to detect patches π, $\pi' \in \Sigma_2$ such that $S(\pi) \subseteq S(\pi')$. In this case, π can certainly be dropped. The inclusions are recognized by applying quantifier elimination to

$$\forall u_1 \forall u_2 \forall u_3 \big(\pi(\mathbf{u}) \longrightarrow \pi'(\mathbf{u})\big).$$

This simplification method is called *quantifier elimination recognized implication* resembling the notion of simplifier recognized implication introduced in [DS97c]. We denote the remainder of Σ_2 obtained this way by $\Sigma(\sigma)$. Note that the number of quantifier eliminations to be performed here is unfortunately quadratic in the number of patches of Σ_2.

5.5.5 Computation Examples

Example 5.22 (Two boxes) Consider the solid $\beta \equiv \beta_1 \vee \beta_2$ obtained as the union of the following two rectangular boxes:

$$\begin{aligned} \beta_1 &\equiv 0 \leqslant u_1 \leqslant 10 \wedge 0 \leqslant u_2 \leqslant 5 \wedge 0 \leqslant u_3 \leqslant 5 \\ \beta_2 &\equiv 5 \leqslant u_1 \leqslant 10 \wedge 0 \leqslant u_2 \leqslant 10 \wedge 1 \leqslant u_3 \leqslant 6. \end{aligned}$$

This solid β is pictured in Figure 5.2. Our algorithm proceeds as follows:

1. Candidate computation yields 16 candidate patches in 210 ms.

2. The computation of $\overline{\sim S(\sigma)}$ by quantifier elimination takes 140 ms. Intersections, CNF computations, and Gröbner simplifications take 710 ms altogether resulting in 16 patches.

3. DNF computation and Gröbner simplification expand this within 560 ms to 41 patches all of which are pure conjunctions.

4. Quantifier elimination recognized implication reduces this to 14 patches within 3.5 s.

The final result gives in a very nice way exactly the surface of β:

$$\begin{aligned}\Sigma(\beta) \equiv \{ & u_1 - 10 = 0 \wedge u_2 - 5 \leqslant 0 \wedge u_2 \geqslant 0 \wedge u_3 - 5 \leqslant 0 \wedge u_3 \geqslant 0, \\ & u_1 - 10 = 0 \wedge u_2 - 10 \leqslant 0 \wedge u_2 \geqslant 0 \wedge u_3 - 1 \geqslant 0 \wedge u_3 - 6 \leqslant 0, \\ & u_1 - 5 = 0 \wedge u_2 - 5 \geqslant 0 \wedge u_2 - 10 \leqslant 0 \wedge u_3 - 1 \geqslant 0 \wedge u_3 - 6 \leqslant 0, \\ & u_1 - 5 = 0 \wedge u_2 - 10 \leqslant 0 \wedge u_2 \geqslant 0 \wedge u_3 - 5 \geqslant 0 \wedge u_3 - 6 \leqslant 0, \\ & u_1 = 0 \wedge u_2 - 5 \leqslant 0 \wedge u_2 \geqslant 0 \wedge u_3 - 5 \leqslant 0 \wedge u_3 \geqslant 0, \\ & u_1 - 5 \leqslant 0 \wedge u_1 \geqslant 0 \wedge u_2 - 5 = 0 \wedge u_3 - 5 \leqslant 0 \wedge u_3 \geqslant 0, \\ & u_1 - 10 \leqslant 0 \wedge u_1 \geqslant 0 \wedge u_2 - 5 = 0 \wedge u_3 - 1 \leqslant 0 \wedge u_3 \geqslant 0, \\ & u_1 - 5 \geqslant 0 \wedge u_1 - 10 \leqslant 0 \wedge u_2 - 10 = 0 \wedge u_3 - 1 \geqslant 0 \wedge u_3 - 6 \leqslant 0, \\ & u_1 - 10 \leqslant 0 \wedge u_1 \geqslant 0 \wedge u_2 = 0 \wedge u_3 - 5 \leqslant 0 \wedge u_3 \geqslant 0, \\ & u_1 - 5 \geqslant 0 \wedge u_1 - 10 \leqslant 0 \wedge u_2 = 0 \wedge u_3 - 1 \geqslant 0 \wedge u_3 - 6 \leqslant 0, \\ & u_1 - 5 \leqslant 0 \wedge u_1 \geqslant 0 \wedge u_2 - 5 \leqslant 0 \wedge u_2 \geqslant 0 \wedge u_3 - 5 = 0, \\ & u_1 - 5 \geqslant 0 \wedge u_1 - 10 \leqslant 0 \wedge u_2 - 10 \leqslant 0 \wedge u_2 \geqslant 0 \wedge u_3 - 6 = 0, \\ & u_1 - 5 \geqslant 0 \wedge u_1 - 10 \leqslant 0 \wedge u_2 - 5 \geqslant 0 \wedge u_2 - 10 \leqslant 0 \wedge u_3 - 1 = 0, \\ & u_1 - 10 \leqslant 0 \wedge u_1 \geqslant 0 \wedge u_2 - 5 \leqslant 0 \wedge u_2 \geqslant 0 \wedge u_3 = 0\}. \quad \square\end{aligned}$$

Example 5.23 (Infinite cylinder) We apply our algorithm to the infinite cylinder $\zeta \equiv u_1^2 + u_2^2 \leqslant 1$ of Example 5.20. Of course, the correct boundary $u_1^2 + u_2^2 = 1$ is the only candidate patch and also forms the final solution. The in this case superfluous quantifier elimination for computing $\overline{\sim S(\zeta)}$ requires 1.6 s. All other steps need no measurable time. □

Example 5.24 (Bounded cylinder) For the cylinder $\zeta' \equiv u_1^2 + u_2^2 \leqslant 1 \wedge 0 \leqslant u_3 \leqslant 3$ the situation is similar at the beginning: The three correct candidates are obtained after 30 ms, $\overline{\sim S(\zeta')}$ requires 1.9 s. Now, the CNF computation and Gröbner simplification for the intersected candidates take 59 s. Finally, DNF computation plus Gröbner simplification and quantifier elimination recognized implication take 30 ms and 59 ms, respectively. □

Example 5.25 (Two boxes with hole) We revisit the two boxes of Example 5.22, and include a hole into one of the boxes:

$$\beta' \equiv \beta \wedge \neg\beta_3, \quad \text{where} \quad \beta_3 \equiv (2 < u_1 < 3 \wedge 2 < u_2 < 3 \wedge 2 < u_3 < 3).$$

A wire frame model of the obtained solid is pictured in Figure 5.3. Here, 26 candidate patches are computed in 370 ms, and the quantifier elimination for computing $\overline{\sim S(\beta')}$ takes 750 ms. Then the CNF computation for the intersection with the candidates fails due to combinatorial explosion, and consequently our entire algorithm fails. □

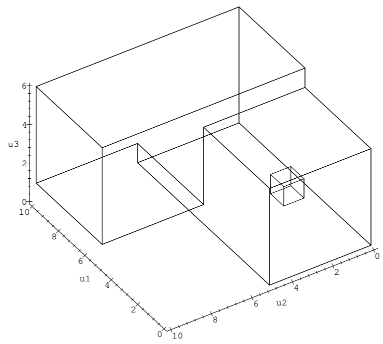

Figure 5.3: The two boxes with hole of the Examples 5.25 and 5.27

5.5.6 An Approximative Variant

Our examples have faced us with two possible problems: On one hand, the quantifier elimination for the surface computation step can be time consuming. On the other hand, the boolean normal form computations following this step can fail. In the latter case, the reason is the complexity of the quantifier elimination result. So replacing the quantifier elimination by something simpler would probably help with both problems.

Our idea is to copy the style of the candidate patch computation to the computation of the exterior of the complement. We replace in our algorithm $\overline{\sim S(\sigma)}$ by the solid $S(\omega(\neg\sigma))$ described by the *weakening* of $\neg\sigma$, which we define syntactically. The weakening $\omega(\neg\sigma)$ is obtained from $\neg\sigma$ as follows:

1. The logical negation "$\neg$"is moved inside σ, and finally eliminated by coding it into the relations of the atomic formulas.
2. Each inequality $t \neq 0$ is replaced by "true." Atomic formulas $t > 0$ and $t < 0$ are replaced by $t \geqslant 0$ and $t \leqslant 0$, respectively.

Lemma 5.26 $\overline{\sim S(\sigma)} \subseteq S\big(\omega(\neg\sigma)\big)$.

Proof We know that $\sim S(\sigma) = S(\neg\sigma)$. The first step of weakening, which moves inside and encodes "$\neg$" is an equivalence transformation leading to, say, σ'. It now suffices to show that $\overline{S(\sigma')} \subseteq S\big(\omega'(\sigma')\big)$, where ω' is the second step in our definition of weakening. It is not hard to see that ω' is compatible with the following transformations on σ':

- normalizing all right hand sides of atomic formulas to 0,
- replacing equations $t = 0$ by $t \geqslant 0 \wedge t \leqslant 0$,
- replacing inequalities $t \neq 0$ by $t > 0 \vee t < 0$,
- turning all formulas $t < 0$ and $t \leqslant 0$ into $-t > 0$ and $-t \geqslant 0$, respectively,
- computing a DNF by applying only rules of propositional logic.

We may thus assume wlog. that σ' is of the form

$$\sigma'(\mathbf{u}) \equiv \bigvee_{i\in I}\bigwedge_{j\in J_i} t_{ij}(\mathbf{u})\ \varrho_{ij}\ 0, \quad \varrho_{ij} \in \{>, \geqslant\}.$$

Accordingly, $\omega'(\sigma')(\mathbf{u}) \equiv \bigvee_{i\in I}\bigwedge_{j\in J_i} t_{ij}(\mathbf{u}) \geqslant 0$, and it is easy to see that $S(\sigma') \subseteq S\big(\omega'(\sigma')\big)$. We have to show that even $\overline{S(\sigma')} \subseteq S\big(\omega'(\sigma')\big)$.

Let $\mathbf{x} \in \overline{S(\sigma')}$, and assume for a contradiction that $\mathbf{x} \notin S\big(\omega'(\sigma')\big)$, i.e., $\mathbf{x} \in {\sim} S\big(\omega'(\sigma')\big) = S\big(\neg\omega'(\sigma')\big)$. Obviously,

$$\neg\omega'(\sigma')(\mathbf{u}) \longleftrightarrow \bigwedge_{i\in I}\bigvee_{j\in J_i} t_{ij}(\mathbf{u}) < 0,$$

and from this description, we see that ${\sim} S\big(\omega'(\sigma')\big)$ is open, because each of the $S(t_{ij} < 0)$ is the preimage of the open interval $]{-\infty}, 0[$ under the continuous polynomial function given by $t_{ij}(\mathbf{u})$, and thus open, and $S\big(\neg\omega'(\sigma')\big)$ is an intersection of finite unions of these open sets. This means, we can choose $\varepsilon > 0$ such that $\mathbf{y} \in {\sim} S\big(\omega'(\sigma')\big)$ for all $\mathbf{y}$ with $d(\mathbf{y}, \mathbf{x}) < \varepsilon$. On the other hand, we know from $\mathbf{x} \in \overline{S(\sigma')}$ that for our ε there must be some $\mathbf{y}$ with $d(\mathbf{y}, \mathbf{x}) < \varepsilon$ and $\mathbf{y} \in S(\sigma') \subseteq S\big(\omega'(\sigma')\big)$, a contradiction. □

Note that the inclusion relation in the lemma is in general a proper one. Thus the introduction of weakening leads to an approximative method that, in general, can lead to wrong results. Assume, for instance, that $\neg\sigma$ contains after moving the negation inside a contradictory strong inequality:

$$\neg\sigma \longleftrightarrow \varphi \vee u_1^2 + u_2^2 < 0.$$

weakening will then compute $\omega(\neg\sigma) \equiv \omega(\varphi) \vee u_1^2 + u_2^2 \leqslant 0$. That is, there comes into existence a cylinder $\zeta \equiv u_1^2 + u_2^2 = 0$ that possibly reaches outside the closure $\overline{{\sim} S(\sigma)}$. This cylinder ζ can in turn intersect with candidate patches of the original solid σ that are not part of its surface. During our practical experiments we have, however, never observed such cases. In particular, contradictions as obvious as $u_1^2 + u_2^2 < 0$ are immediately detected and removed by our standard simplifier.

We are going to discuss in detail Example 5.25 with weakening.

Example 5.27 (Two boxes with hole revisited) We again compute the 26 candidate patches in 330 ms. The weakening of β' does not take any measurable time. This time, the CNF computations for the intersection with the candidates succeed within 1.6 s yielding again 26 patches. DNF computation and Gröbner simplification take 8.7 s yielding 200 patches. These 200 patches are reduced by quantifier elimination recognized implication to 20 patches $\Sigma(\beta')$ within 33.6 s. These 20 patches yield in fact a beautiful description of the surface of our solid:

$$\begin{aligned}
\Sigma(\beta') \equiv \{ & u_1 - 3 = 0 \wedge u_2 - 2 \geqslant 0 \wedge u_2 - 3 \leqslant 0 \wedge u_3 - 2 \geqslant 0 \wedge u_3 - 3 \leqslant 0, \\
& u_1 - 10 = 0 \wedge u_2 - 5 \leqslant 0 \wedge u_2 \geqslant 0 \wedge u_3 - 5 \leqslant 0 \wedge u_3 \geqslant 0, \\
& u_1 - 10 = 0 \wedge u_2 - 10 \leqslant 0 \wedge u_2 \geqslant 0 \wedge u_3 - 1 \geqslant 0 \wedge u_3 - 6 \leqslant 0, \\
& u_1 - 5 = 0 \wedge u_2 - 5 \geqslant 0 \wedge u_2 - 10 \leqslant 0 \wedge u_3 - 1 \geqslant 0 \wedge u_3 - 6 \leqslant 0, \\
& u_1 - 5 = 0 \wedge u_2 - 10 \leqslant 0 \wedge u_2 \geqslant 0 \wedge u_3 - 5 \geqslant 0 \wedge u_3 - 6 \leqslant 0,
\end{aligned}$$

$$
\begin{aligned}
&u_1 - 2 = 0 \wedge u_2 - 2 \geqslant 0 \wedge u_2 - 3 \leqslant 0 \wedge u_3 - 2 \geqslant 0 \wedge u_3 - 3 \leqslant 0,\\
&u_1 = 0 \wedge u_2 - 5 \leqslant 0 \wedge u_2 \geqslant 0 \wedge u_3 - 5 \leqslant 0 \wedge u_3 \geqslant 0,\\
&u_1 - 2 \geqslant 0 \wedge u_1 - 3 \leqslant 0 \wedge u_2 - 3 = 0 \wedge u_3 - 2 \geqslant 0 \wedge u_3 - 3 \leqslant 0,\\
&u_1 - 5 \leqslant 0 \wedge u_1 \geqslant 0 \wedge u_2 - 5 = 0 \wedge u_3 - 5 \leqslant 0 \wedge u_3 \geqslant 0,\\
&u_1 - 10 \leqslant 0 \wedge u_1 \geqslant 0 \wedge u_2 - 5 = 0 \wedge u_3 - 1 \leqslant 0 \wedge u_3 \geqslant 0,\\
&u_1 - 5 \geqslant 0 \wedge u_1 - 10 \leqslant 0 \wedge u_2 - 10 = 0 \wedge u_3 - 1 \geqslant 0 \wedge u_3 - 6 \leqslant 0,\\
&u_1 - 2 \geqslant 0 \wedge u_1 - 3 \leqslant 0 \wedge u_2 - 2 = 0 \wedge u_3 - 2 \geqslant 0 \wedge u_3 - 3 \leqslant 0,\\
&u_1 - 10 \leqslant 0 \wedge u_1 \geqslant 0 \wedge u_2 = 0 \wedge u_3 - 5 \leqslant 0 \wedge u_3 \geqslant 0,\\
&u_1 - 5 \geqslant 0 \wedge u_1 - 10 \leqslant 0 \wedge u_2 = 0 \wedge u_3 - 1 \geqslant 0 \wedge u_3 - 6 \leqslant 0,\\
&u_1 - 2 \geqslant 0 \wedge u_1 - 3 \leqslant 0 \wedge u_2 - 2 \geqslant 0 \wedge u_2 - 3 \leqslant 0 \wedge u_3 - 3 = 0,\\
&u_1 - 5 \leqslant 0 \wedge u_1 \geqslant 0 \wedge u_2 - 5 \leqslant 0 \wedge u_2 \geqslant 0 \wedge u_3 - 5 = 0,\\
&u_1 - 5 \geqslant 0 \wedge u_1 - 10 \leqslant 0 \wedge u_2 - 10 \leqslant 0 \wedge u_2 \geqslant 0 \wedge u_3 - 6 = 0,\\
&u_1 - 2 \geqslant 0 \wedge u_1 - 3 \leqslant 0 \wedge u_2 - 2 \geqslant 0 \wedge u_2 - 3 \leqslant 0 \wedge u_3 - 2 = 0,\\
&u_1 - 5 \geqslant 0 \wedge u_1 - 10 \leqslant 0 \wedge u_2 - 5 \geqslant 0 \wedge u_2 - 10 \leqslant 0 \wedge u_3 - 1 = 0,\\
&u_1 - 10 \leqslant 0 \wedge u_1 \geqslant 0 \wedge u_2 - 5 \leqslant 0 \wedge u_2 \geqslant 0 \wedge u_3 = 0\}. \quad \square
\end{aligned}
$$

5.5.7 Adjacency Information

A B-REP description of a solid $S(\sigma)$ will contain besides the surface patches $\Sigma(\sigma)$ obtained by our algorithm also adjacency information on these patches. For a pair π, $\pi' \in \Sigma(\sigma)$ of patches, it can easily be checked whether π and π' are disjoint or not by applying quantifier elimination to

$$\exists u_1 \exists u_2 \exists u_3 \big(\pi(\mathbf{u}) \wedge \pi'(\mathbf{u})\big).$$

If the result is "true," this means there is a common point, which implies adjacency. Note, that in this case, it is still possible that π and π' are actually overlapping fragments of one facet, and should thus be joined. All examples discussed above do, however, not show this misbehavior, and the computation time for the quadratically many eliminations also does not cause any problem.

5.6 Geometric Feasibility

Solving geometric constraints is an important topic for the development of intelligent CAD systems [GC98a, GC98b, Hof95]. The idea is that the user inputs declarative descriptions of diagrams or engineering drawings instead of specifying how to draw these figures.

There are three approaches geometric constraint solving: a construction approach, a numerical approach, and a symbolic approach. The construction approach transforms the geometric constraints into a form that is easy to draw. One such target form is called RC-configuration. This is a list of points which can successively be constructed by ruler and compass from their respective predecessors. The numerical approach turns the constraint system into algebraic equations similar in style to the translations we used for geometry proving in Chapter 3. The obtained systems of equations are then solved numerically. The symbolic approach, in contrast, solves the obtained systems by algebraic methods such as Gröbner basis techniques or Wu–Ritt reduction.

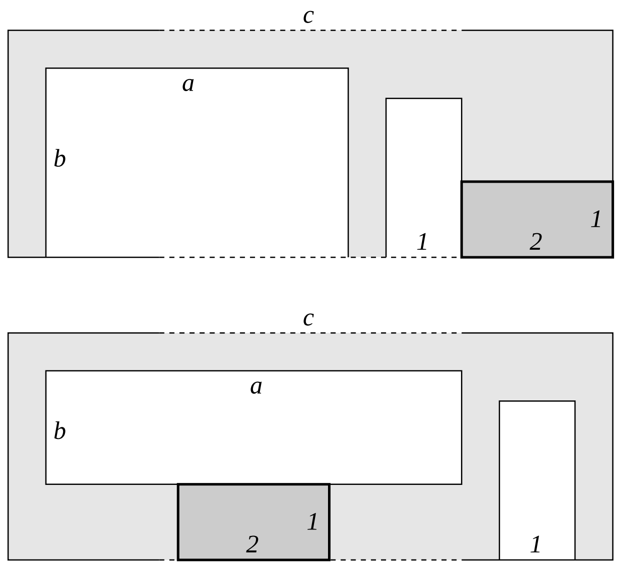

Figure 5.4: Two possible layouts for a house front (Example 5.28)

Note that both the numeric and the symbolic approach can, in contrast to real quantifier elimination, not handle ordering inequalities. We are going to discuss two examples that make intense use of such inequalities.

Example 5.28 (House front) We wish to place 3 objects within a wall: a window, a door, and a heater. Our layout has to obey the following design constraints (compare Figure 5.4): The room is 2.5 m high, we leave the length of our wall parametric, say c m. The window is of parametric size $a \times b$ m. It has to placed at least 0.5 m from the end of the wall, from the ceiling, and from the door. The door is 1 m wide, its height does not matter. It also has to be placed at least 0.5 m from the wall to the right of the window. The heater is 2 m wide and 1 m high. It has to be placed on the floor either to the right of the door or centered below the window. In the latter case, the window also must be at least 2 m wide. Denoting the lower left corner of the window by (w_x, w_y), and the right borders of door and heater by d_x and h_x, respectively, we obtain the following subsystems of constraints:

$$\begin{aligned}
\pi &\equiv a > 0 \land b > 0 \land c > 0 \\
\text{window} &\equiv 0.5 \leqslant w_x \land w_x + a \leqslant d_x - 0.5 \land w_y \geqslant 0 \land w_y + b \leqslant 2.5 - 0.5 \\
\text{door} &\equiv d_x + 1 \leqslant c - 0.5 \\
\text{w-heater} &\equiv h_x + 1 = w_x + \frac{a-2}{2} \land w_y \geqslant 1 \land a \geqslant 2 \land F = 1 \\
\text{d-heater} &\equiv h_x = c - 2 \land d_x \leqslant h_x - 1 \land F = 0.
\end{aligned}$$

Using these constraints, our layout problem can be described by the following formula with parameters a, b, c, and F:

$$\Delta(a, b, c, F) \equiv \exists d_x \exists h_x \exists w_x \exists w_y \big(\pi \land \text{window} \land \text{door} \land (\text{w-heater} \lor \text{d-heater})\big).$$

Table 5.2: Minimal wall lengths for given window areas (Example 5.28)

A (in m^2)	$c_{\min}$ (in m) for ω_A	$c_{\min}$ (in m) for δ_A
0.5	4.5	4.25
1	4.5	4.5
1.5	4.5	4.75
2	4.5	5
2.5	5	5.25
3	5.5	5.5
3.5	6	5.75
4	6.5	6
4.5	7	6.25
5	7.5	6.5

The parameter F is a flag that will indicate the chosen design variant in the result. This technique has been firstly used by A. Dolzmann for solving sophisticated scheduling problems by quantifier elimination.

Quantifier elimination on Δ plus subsequent Gröbner simplification yields after only 10 ms the following quantifier-free equivalent $\Delta'(a, b, c, F)$ for Δ:

$$(2a - 2c + 5 \leqslant 0 \land a - 2 \geqslant 0 \land b - 1 \leqslant 0 \land b > 0 \land c > 0 \land F - 1 = 0) \lor$$
$$(a - c + 4 \leqslant 0 \land a > 0 \land b - 2 \leqslant 0 \land b > 0 \land c > 0 \land F = 0).$$

This Δ' precisely describes the relations that must hold between the parameters a, b, and c for our two possible designs. If we substitute concrete window dimensions, say $a = 2$ and $b = 1$, into Δ' and apply the standard simplifier, we immediately obtain

$$(2c - 9 \geqslant 0 \land F - 1 = 0) \lor (c - 6 \geqslant 0 \land F = 0).$$

So for the chosen window dimensions, with the heater under the window the wall must be at least 4.5 m long, and with the heater on the right hand side of the door the wall must be even 6 m long.

Next, we want to decide which layout to use with windows of a given minimal area. For areas $A \in \{0.5, 1, 1.5, \ldots, 5\}$ we compute quantifier-free equivalents for the formulas

$$\begin{aligned}
\omega_A(c) &\equiv \exists d_x \exists h_x \exists w_x \exists w_y \exists a \exists b \exists F(\pi \land \text{window} \land \text{door} \land \text{w-heater} \land ab \geqslant A)\\
\delta_A(c) &\equiv \exists d_x \exists h_x \exists w_x \exists w_y \exists a \exists b \exists F(\pi \land \text{window} \land \text{door} \land \text{d-heater} \land ab \geqslant A).
\end{aligned}$$

These yield, separately for each layout variant, straightforward descriptions of the minimal wall lengths for the chosen window sizes. For instance,

$$\omega'_{0.5}(c) \equiv 2c - 9 \geqslant 0 \quad \text{and} \quad \delta'_{0.5}(c) \equiv 4c - 17 \geqslant 0.$$

The eliminations for all the A yield Table 5.2, the computation of which takes 730 ms altogether. □

Our house front example above is in a certain way very well-behaved: We are dealing exclusively with rectangles that are positioned in parallel to the

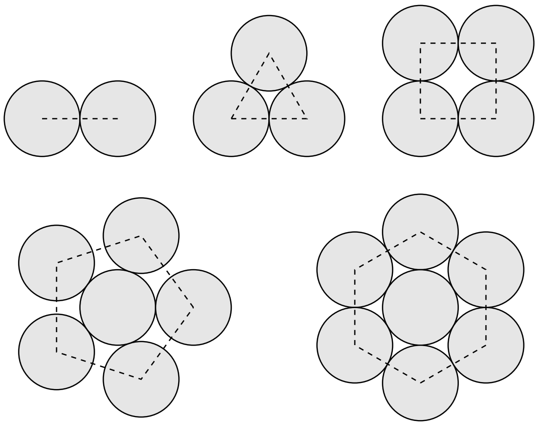

Figure 5.5: Packing disks (Example 5.29)

axes of our coordinate system. Consequently, we do not need quadratic terms for encoding Euclidean distances. This changes in the following example, which exclusively deals with circles.

Example 5.29 (Packing disks) For $n \in \mathbb{N}$, we want to arrange n unit disks in the plane such that the maximal distance between any two if them is minimized. This is described by the following formula, where we use the Euclidean metric for $\delta_\geqslant$ and $\delta_\leqslant$:

$$\beta_n(z) \equiv \exists x_1^{(1)} \exists x_2^{(1)} \ldots \exists x_1^{(n)} \exists x_2^{(n)} \Big(\bigwedge_{i=1}^{n} \bigwedge_{j=1}^{i-1} \big(\delta_\geqslant(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}, 1) \land \delta_\leqslant(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}, z)\big)\Big).$$

If we apply extended quantifier elimination to some β_n, the quantifier-free equivalent will describe the minimal maximal distance in terms of z, and the answer will yield one possible list of centers of the unit disks such that this distance is realized. See Figure 5.5 for solutions for $n \leqslant 6$, which are quite easy to guess. The problem starts to get interesting for $n = 7$. For our practical computations, we normalize

$$\mathbf{x}^{(1)} = (0, 0) \quad \text{and} \quad \mathbf{x}^{(2)} = (1, 0),$$

and drop the corresponding quantifiers. Then β_1 and β_2 are quantifier-free. For β_3 we obtain within 60 ms the quantifier-free equivalent $\beta_3' \equiv z^2 - 1 \geqslant 0$, and the answer

$$\mathbf{x}^{(3)} = \Big(\frac{1}{2}, \frac{\sqrt{4z^2 - 1}}{2}\Big),$$

i.e, the distance z is 1 and the center of the third unit disk can be located at $\mathbf{x}^{(3)} = \big(\frac{1}{2}, \frac{\sqrt{3}}{2}\big)$. The elimination for β_4 yields after approximately 10 minutes a formula with 4911 atomic formulas, where the quantifiers "$\exists x_1^{(3)}$" and "$\exists x_2^{(3)}$" are still present. They cannot be eliminated due to the degree restrictions of the virtual substitution method. The β_n for $n > 4$ will certainly also fail. □

Note that the packing problem is actually solved even for balls in 3-space. This solution plays an important role for understanding the structure of metal lattices in chemistry [HW85].

5.7 Conclusions

We have proposed to use quantifier-free formulas for representing solids in CAD systems. The standard operations commonly encountered in solid modelers and, in addition, offsetting and blending can then be performed by quantifier elimination. In particular blending has been extensively discussed in the literature during the last years.

Our approach to blending uses in a natural way open sets as intermediate results. This allows us to define very adequate and convenient notions of blending from inside and from outside in terms of some newly developed offsetting operations. We have carefully analyzed the mathematical properties of our various offsets. This has finally enabled us to straightforwardly prove the relevant properties of our notions of blending from inside and from outside. Moreover, we could use our offsets as a tool for analyzing, comparing, and systemizing also the notions of blending discussed elsewhere.

We have demonstrated that one can obtain the information provided by common B-REP descriptions from our formula descriptions. This information allows to render the modeled geometric configurations on a screen or printer. For this purpose, we have changed for the first and only time in this thesis our strategy from the pure, though often quite sophisticated, modeling of problems by first-order formulas to the development of new algorithms, which certainly involve quantifier elimination for several subtasks. By suggesting an approximate method for computing surface patches of a given formula-described solid, we have demonstrated that the quantifier elimination approach can be feasible and reasonable even in situations where we are faced with the problem that pure quantifier elimination approaches fail due to the poor inherent worst-case complexity of this method.

Finally, we have shown that our methods are also straightforwardly applicable to geometric constraint solving, which is of increasing importance in the scientific literature on the design of graphics systems, where it is considered to be the basic tool for the next generation of modeling systems. Here we have, similar to the situation with geometric theorem proving, again to compete with established algebraic approaches. Again, our real quantifier elimination approach has the great advantage of allowing ordering constraints in the descriptions of the modeled sceneries.

Chapter 6

Concluding Remarks

We are finally going to summarize and evaluate our work.

After an introduction chapter containing some historical notes, giving an overview over the content of this thesis, and explicitly stating the main results, the work starts in Chapter 2 by introducing the reader to the scientific framework of real quantifier elimination. For this, we haven given a historical overview over the development of this field of science. Then we have summarized a large part of the known results and methods, where we focussed on material that is in some way relevant for applications.

The new contributions provided by this thesis start with the introduction of generic quantifier elimination and the related concepts in Section 2.4. This is followed by a brief summary of simplification strategies, which have in the past turned out crucial for the success of applied quantifier elimination. At this point, the major part of the algorithmic tools used throughout the thesis has been introduced.

Our focus is naturally on quantifier elimination by virtual substitution, which has been implemented within the computer logic system REDLOG by the author et al. that, as a part of the computer algebra system REDUCE, is widely accepted and used inside the scientific community. In fact, REDUCE provides with REDLOG the currently only professionally maintained and marketed implementation of real quantifier elimination. We have devoted an entire section to the design of REDLOG, which has been our major computation tool throughout this thesis. All our implementations of quantifier elimination variants by virtual substitution are restricted to formulas, where the quantified variables occur with low degree only. In theory, the method can be extended to arbitrary degrees. It is, however, not clear if a corresponding implementation would be applicable to problems of reasonable sizes. In some cases discussed throughout this thesis, the degree restriction was actually relevant in that problems could be formulated but not straightforwardly solved.

A summary of a variety of application areas has given an idea of the scope of the methods used here. They provide an extremely general tool that is by no means restricted to geometric problems, which we have focussed on. Finally, we have clarified the difference between real quantifier elimination and apparently closely related areas such as automatic theorem proving and constraint logic programming.

In Chapter 3, we turned to the first major geometric application area: geometric theorem proving. By automatically proving some well-known benchmark examples with our REDLOG implementation, we have demonstrated that our method can compete with the algebraic methods discussed in the literature so far. Since our method is a genuinely real, in contrast to complex, proof method, its application range exceeds that of the traditional methods in various ways: First, it can handle ordering inequalities; second, it can prove theorems, which do not hold over the complex numbers but only over the reals; third, it can not only prove but also disprove theorems.

After specification of the independent parameters in the given problem, the assumptions made by the generic quantifier elimination for reducing the complexity of the elimination algorithm turn out to specify mostly the non-degeneracy conditions that are actually necessary for the input conjecture to hold. The apparent restriction of generic quantifier elimination in contrast to regular quantifier elimination has thus turned out as a valuable tool. In fact, generic quantifier elimination is, at least in the application area considered here, superior to regular quantifier elimination, which cannot straightforwardly be applied in such a way that non-degeneracy conditions are detected.

There are geometric assertions that do not generally hold without an additional assumption that in turn cannot be characterized as non-degeneracy condition. Such an assumption may, e.g., describe the locus of certain points to fulfill the geometrical requirements of a theorem, or it can contain information on the ordering between certain values. For such examples, our method supplies by means of the actual elimination result, which is an arbitrary boolean combination of equations and inequalities, exactly the necessary and sufficient conditions for the theorem to hold wrt. the classical non-degeneracy conditions obtained independently. This extends the application range of our method from pure theorem proving to the much more sophisticated concept of theorem finding.

In Chapter 4 we have discussed the formulation of various standard problems of computational geometry as quantifier elimination problems or extended quantifier elimination problems. Instead of competing with existent approaches, we have focussed on the fact that we can handle these problems parametrically. Such a parametric solution actually provides the solutions for an entire class of problems given by all possible interpretations for the parameters. The single or even double exponential worst-case complexity of our method has to be viewed and judged on the basis of this fact. Anyway, we have encountered examples, where the elimination time exceeded several hours. In such cases the application of quantifier elimination techniques is obviously not reasonable.

As with geometric proving, generic quantifier elimination has turned out crucial for our success. In contrast to the application to geometry proving this is, however, mainly for efficiency reasons. The effect of obtaining important extra information via the assumptions made by the generic elimination procedure has not shown here. On the other hand, the use of generic quantifier elimination did not impose any restrictions that cannot be overcome; we have demonstrated how to deal with ND-conditions that even excluded highly relevant parts of the solution. The major part of the discussed examples in the area of computational geometry cannot be satisfactorily treated by any available implementation of regular, i.e. non-extended and non-generic, quantifier elimination.

In Chapter 5, we have finally proposed to use quantifier-free formulas for representing solids in CAD systems. The standard operations commonly encountered in solid modelers and, in addition, offsetting and blending can then be performed by quantifier elimination. In particular blending has been extensively discussed in the literature during the last years.

Our approach to blending uses in a natural way open sets as intermediate results. This allows us to define very adequate and convenient notions of blending from inside and from outside in terms of some newly developed offsetting operations. We have carefully analyzed the mathematical properties of our various offsets. This has finally enabled us to straightforwardly prove the relevant properties of our notions of blending from inside and from outside. Moreover, we could use our offsets as a tool for analyzing, comparing, and systemizing also the notions of blending discussed elsewhere.

We have demonstrated that one can obtain the information provided by common B-REP descriptions from our formula descriptions. This information allows to render the modeled geometric configurations on a screen or printer. For this purpose, we have changed for the first and only time in this thesis our strategy from the pure, though often quite sophisticated, modeling of problems by first-order formulas to the development of new algorithms, which certainly involve quantifier elimination for several subtasks. By suggesting an approximate method for computing surface patches of a given formula-described solid, we have demonstrated that the quantifier elimination approach can be feasible and reasonable even in situations where we are faced with the problem that some straightforward quantifier elimination technique fails due to the poor inherent worst-case complexity of the method. Such a failure can be caused either by excessive elimination times or by large incomprehensible results that cannot be simplified automatically. It is a well-known particular drawback of the virtual substitution method to produce such redundant results.

Finally, we have shown that our methods are also straightforwardly applicable to geometric constraint solving, which is of increasing importance in the scientific literature on the design of graphics systems, where it is considered to be the basic tool for the next generation of modeling systems. Here we have, similar to the situation with geometric theorem proving, again to compete with established algebraic approaches. Again, our real quantifier elimination approach has the great advantage of allowing ordering constraints in the descriptions of the modeled sceneries.

Bibliography

[ACM84] Dennis S. Arnon, George E. Collins, and Scott McCallum. Cylindrical algebraic decomposition I: The basic algorithm. *SIAM Journal on Computing*, 13(4):865–877, November 1984.

[ADY+96] Chaouki T. Abdallah, Peter Dorato, Wei Yang, Richard Liska, and Stanly Steinberg. Applications of quantifier elimination theory to control system design. In *Proceedings of the 4th IEEE Mediterranean Symposium on Control and Automation*, pages 340–345. IEEE, 1996.

[Arn81] Dennis S. Arnon. *Algorithms for the geometry of semi-algebraic sets.* Ph.D. dissertation, Computer Sciences Department, University of Wisconsin, Madison, 1981. Technical Report No. 436.

[BCE+93] Bruno Buchberger, George E. Collins, Mark J. Encarnacion, Hoon Hong, Jeremy R. Johnson, Werner Krandick, Rüdiger Loos, Ana M. Mandache, Andreas Neubacher, and Herbert Vielhaber. Saclib 1.1 user's guide. RISC-Linz Series Technical Report 93-19, Research Institute for Symbolic Computation, Johannes Kepler University, A-4040 Linz, Austria, 1993.

[BdB07] Ferdinand François Désiré Budan de Boislaurent. Nouvelle méthode pour la résolution des équations numériques d'un degré quelconque, 1807. 2nd edition, Paris 1822.

[Bec91] Eberhard Becker. Sums of squares and quadratic forms in real algebraic geometry. In *Cahiers du Séminaire d'Historie de Mathématiques*, volume 1, pages 41–57. Université Pierre et Marie Curie, Laboratoire de Mathématiques Fondamentales, Paris, 1991.

[BKOS97] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry, Algorithms and Applications.* Springer, Berlin, Heidelberg, New York, 1997.

[BKR86] Michael Ben-Or, Dexter Kozen, and John Reif. The complexity of elementary algebra and geometry. *Journal of Computer and System Sciences*, 32:251–264, 1986.

[BPR94] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. On the combinatorial and algebraic complexity of quantifier elimination. In Shafi Goldwasser, editor, *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 632–641, Santa Fe, NM, November 1994. IEEE Computer Society Press, Los Alamitos, CA.

[Bro83] Morton Brown. Problem proposal in the *Problems and Solutions* section. *American Mathematical Monthly*, 90(8):569, October 1983.

[Buc65] Bruno Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.* Doctoral dissertation, Mathematical Institute, University of Innsbruck, Innsbruck, Austria, 1965.

[Bur90] Klaus-Dieter Burhenne. Implementierung eines Algorithmus zur Quantorenelimination für lineare reelle Probleme. Diploma thesis, Universität Passau, D-94030 Passau, Germany, December 1990.

[BW94] Eberhard Becker and Thorsten Wörmann. On the trace formula for quadratic forms. In William B. Jacob, Tsit-Yuen Lam, and Robert O. Robson, editors, *Recent Advances in Real Algebraic Geometry and Quadratic Forms*, volume 155 of *Contemporary Mathematics*, pages 271–291. American Mathematical Society, American Mathematical Society, Providence, Rhode Island, 1994. Proceedings of the RAGSQUAD Year, Berkeley, 1990–1991.

[BWK93] Thomas Becker, Volker Weispfenning, and Heinz Kredel. *Gröbner Bases, a Computational Approach to Commutative Algebra*, volume 141 of *Graduate Texts in Mathematics.* Springer, New York, 1993.

[CH91] George E. Collins and Hoon Hong. Partial cylindrical algebraic decomposition for quantifier elimination. *Journal of Symbolic Computation*, 12(3):299–328, September 1991.

[Cho88] Shang-Ching Chou. *Mechanical Geometry Theorem Proving.* Mathematics and its applications. D. Reidel Publishing Company, Dordrecht, Boston, Lancaster, Tokyo, 1988.

[Col74] George E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition—preliminary report. *ACM SIGSAM Bullitin*, 8(3):80–90, August 1974. Proceedings of the EUROSAM '74.

[Col75] George E. Collins. Quantifier elimination for the elementary theory of real closed fields by cylindrical algebraic decomposition. In H. Brakhage, editor, *Automata Theory and Formal Languages. 2nd GI Conference*, volume 33 of *Lecture Notes in Computer Science*, pages 134–183. Gesellschaft für Informatik, Springer-Verlag, Berlin, Heidelberg, New York, 1975.

[Col85] George E. Collins. The SAC-2 computer algebra system. In Bob F. Caviness, editor, *EUROCAL '85; Proceedings of the European Conference on Computer Algebra*, volume 104 of *Lecture Notes in Computer Science*, pages 34–35. Springer, Berlin, Heidelberg, New York, Tokyo, April 1985.

[Col90] Alain Colmerauer. Prolog III. *Communications of the ACM*, 33(7):70–90, July 1990.

[CT95] Pasqualina Conti and Carlo Traverso. A case of automatic theorem proving in Euclidean geometry: the MacLane 8_3 theorem. In Gérard Cohen, Marc Giusti, and Teo Mora, editors, *AAECC-11*, volume 948 of *Lecture Notes in Computer Science*, pages 183–193. Springer, Berlin, Heidelberg, 1995.

[Dan51] George B. Dantzig. Maximization of a linear function of variables subject to linear inequalities. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*, pages 339–347. John Wiley & Sons, New York, 1951.

[Des73] René Descartes. La géometrie—livre III. In Charles Adam and Paul Tannery, editors, *Discours de la Méthode & Essais*, volume 6 of *Œuvres des Descartes*, pages 442–485. Librairie Philosophique J. Vrin, Paris, new edition, 1973.

[DGS98] Andreas Dolzmann, Oliver Gloor, and Thomas Sturm. Approaches to parallel quantifier elimination. In Oliver Gloor, editor, *Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation (ISSAC 98)*, pages 88–95, Rostock, Germany, August 1998. ACM, ACM Press, New York, 1998.

[DH88] James H. Davenport and Joos Heintz. Real quantifier elimination is doubly exponential. *Journal of Symbolic Computation*, 5(1–2):29–35, February–April 1988.

[Dol94] Andreas Dolzmann. Reelle Quantorenelimination durch parametrisches Zählen von Nullstellen. Diploma thesis, Universität Passau, D-94030 Passau, Germany, November 1994.

[DS96] Andreas Dolzmann and Thomas Sturm. Redlog user manual. Technical Report MIP-9616, FMI, Universität Passau, D-94030 Passau, Germany, October 1996. Edition 1.0 for Version 1.0.

[DS97a] Andreas Dolzmann and Thomas Sturm. Guarded expressions in practice. In Wolfgang W. Küchlin, editor, *Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation (ISSAC 97)*, pages 376–383, Maui, HI, July 1997. ACM, ACM Press, New York, 1997.

[DS97b] Andreas Dolzmann and Thomas Sturm. Redlog: Computer algebra meets computer logic. *ACM SIGSAM Bulletin*, 31(2):2–9, June 1997.

[DS97c] Andreas Dolzmann and Thomas Sturm. Simplification of quantifier-free formulae over ordered fields. *Journal of Symbolic Computation*, 24(2):209–231, August 1997.

[DS99] Andreas Dolzmann and Thomas Sturm. Redlog user manual. Technical Report MIP-9905, FMI, Universität Passau, D-94030 Passau, Germany, April 1999. Edition 2.0 for Version 2.0.

[DSW98a] Andreas Dolzmann, Thomas Sturm, and Volker Weispfenning. A new approach for automatic theorem proving in real geometry. *Journal of Automated Reasoning*, 21(3):357–380, 1998.

[DSW98b] Andreas Dolzmann, Thomas Sturm, and Volker Weispfenning. Real quantifier elimination in practice. In B. H. Matzat, G.-M. Greuel, and G. Hiss, editors, *Algorithmic Algebra and Number Theory*, pages 221–247. Springer, Berlin, 1998.

[DYA97] Peter Dorato, Wei Yang, and Chaouki Abdallah. Robust multi-objective feedback design by quantifier elimination. *Journal of Symbolic Computation*, 24(2):153–159, August 1997. Special issue on applications of quantifier elimination.

[Fou31] Jean Baptiste Joseph Fourier. *Analyse des équations déterminées.* F. Didot, Paris, 1831.

[GC98a] Xiao-Shan Gao and Shang-Ching Chou. Solving geometric constraint systems. I. A global propagation approach. *Computer-Aided Design*, 30(1):47–54, January 1998.

[GC98b] Xiao-Shan Gao and Shang-Ching Chou. Solving geometric constraint systems. II. A symbolic approach and decision of Rc-constructibility. *Computer-Aided Design*, 30(2):115–122, February 1998.

[GV93] Laureano González-Vega. A combinatorial algorithm solving some quantifier elimination problems. Technical Report 11-1993, University of Cantabria, Departemento de Mathematicas, Estadistica y Computación, Facultad de Ciencias, Avenida de Los Castros, 39071 Santander, Spain, November 1993.

[Hal63] Paul R. Halmos. *Lectures on Boolean Algebras*, volume 1 of *Van Nostrand Mathematical Studies.* D. van Nostrand Company, Inc., Princeton, New Jersey, 1963.

[Her05] Charles Hermite. Remarques sur le théorème de M. Sturm. In Emile Picard, editor, *Œuvres des Charles Hermite*, volume 1, pages 284–287. Gauthier-Villars, Paris, 1905.

[HH86] Chistoph Hoffmann and John Hopcroft. Quadratic blending surfaces. *Computer-Aided design*, 18(6):301–306, July and August 1986.

[Hil87] David Hilbert. *Grundlagen der Geometrie.* Teubner Studienbücher Mathematik. Teubner, Stuttgart, Germany, 13th edition, 1987.

[HLS97] Hoon Hong, Richard Liska, and Stanly Steinberg. Testing stability by quantifier elimination. *Journal of Symbolic Computation*, 24(2):161–187, August 1997. Special issue on applications of quantifier elimination.

[Hof89] Christoph M. Hoffmann. *Geometric and Solid Modeling.* Computer Graphics and Geometric Modeling. Morgan Kaufmann, San Mateo, California, 1989.

[Hof90] Christoph M. Hoffmann. Algebraic and numerical techniques for offsets and blends. In Wolfgang Dahmen, Mariano Gasca, and Charles A. Micchelli, editors, *Computation of Curves and Surfaces*, pages 499–528. Kluwer Academic Publishers, Dordrecht, Boston, London, 1990.

[Hof95] Christoph Hoffmann. Geometric constraint solving in r^2 and r^3. In D.Z. Du and F. Huang, editors, *Computing in Euclidean Geometry*, pages 266–298. World Scientfic, 1995.

[Hof96] Christoph M. Hoffmann. How solid is solid modeling. In Ming C. Lin and Dinesh Manocha, editors, *Applied Computational Geometry*, volume 1148 of *Lecture Notes in Computer Science*, pages 1–8. Springer, Berlin, Heidelberg, 1996.

[Hon90a] Hoon Hong. An improvement of the projection operator in cylindrical algebraic decomposition. In Shunro Watanabe and Morio Nagata, editors, *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC 90)*, pages 261–264, Tokyo, Japan, August 1990. ACM, ACM Press, New York, 1990.

[Hon90b] Hoon Hong. *Improvements in CAD-based Quantifier Elimination*. Ph.D. thesis, The Ohio State University, 1990.

[Hon92] Hoon Hong. Simple solution formula construction in cylindrical algebraic decomposition based quantifier elimination. In Paul S. Wang, editor, *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC 92)*, pages 177–188, Berkeley, CA, July 1992. ACM, ACM Press, New York, 1992.

[Hon93a] Hoon Hong. Quantifier elimination for formulas constrained by quadratic equations. In Manuel Bronstein, editor, *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC 93)*, pages 264–274, Kiev, Ukraine, July 1993. ACM, ACM Press, New York, 1993.

[Hon93b] Hoon Hong. Quantifier elimination for formulas constrained by quadratic equations via slope resultants. *THE Computer Journal*, 36(5):440–449, 1993. Special issue on computational quantifier elimination.

[HW85] Arnold F. Holleman and Nils Wiberg. *Lehrbuch der anorganischen Chemie*. De Gruyter, Berlin, New York, 91st–100th edition, 1985.

[Ioa99] Nikolaos I. Ioakimidis. REDLOG-aided derivation of feasibility conditions in applied mechanics and engineering problems under simple inequality constraints. *Journal of Mechanical Engineering (Strojnícky Časopis)*, 50(1):58–69, 1999.

[Jir97] Mats Jirstrand. Nonlinear control system design by quantifier elimination. *Journal of Symbolic Computation*, 24(2):137–152, August 1997. Special issue on applications of quantifier elimination.

[Kap86] Deepak Kapur. Using Gröbner bases to reason about geometry problems. *Journal of Symbolic Computation*, 2(4):399–408, December 1986.

[KM88] Deepak Kapur and Joseph L. Mundy. Wu's method and its application to perspective viewing. *Artificial Intelligence*, 37(1-3):15–36, December 1988. Special volume on geometric reasoning.

[KS86] Bernhard A. Kutzler and Sabine Stifter. On the application of Buchberger's algorithm to automated geometry theorem proving. *Journal of Symbolic Computation*, 2(4):389–397, December 1986.

[Kut88] Bernhard A. Kutzler. *Algebraic Approaches to Automated Theorem Proving*. PhD thesis, Johannes Kepler Universität Linz, 1988. RISC-Linz series no. 88-74.0.

[KW] M'hammed el Kahoui and Andreas Weber. Deciding hopf bifurcations by quantifier elimination in a software component architecture. To appear in the Journal of Symbolic Computation.

[LPY99] Gerardo Lafferriere, George J. Pappas, and Sergio Yovine. A new class of decidable hybrid systems. In Frits W. Vaandrager and Jan H. van Schuppen, editors, *Hybrid Systems and Control. Proceedings of the Second International Workshop, HSCC'99, Berg en Dal, The Netherlands, March 1999*, volume 1569 of *Lecture Notes in Computer Science*, pages 137–151. Springer, Berlin, Germany, 1999.

[LW93] Rüdiger Loos and Volker Weispfenning. Applying linear quantifier elimination. *The Computer Journal*, 36(5):450–462, 1993. Special issue on computational quantifier elimination.

[Mac36] Saunders MacLane. Some interpretations of abstract linear dependence in terms of projective geometry. *American Journal of Mathematics*, 58:236–240, 1936.

[McC84] Scott McCallum. *An Improved Projection Operation for Cylindrical Algebraic Decomposition*. Ph.D. dissertation, Computer Sciences Department, University of Wisconsin, Madison, 1984.

[McC88] Scott McCallum. An improved projection operation for cylindrical algebraic decomposition of three-dimensional space. *Journal of Symbolic Computation*, 5(1–2):141–161, February–April 1988.

[MCG94] Nicholas Freitag McPhee, Shang-Ching Chou, and Xiao-Shan Gao. Mechanically proving geometry theorems using a combination of Wu's method and Collins' method. In Alan Bundy, editor, *Automated Deduction—CADE-12*, volume 814 of *Lecture Notes in Artificial Intelligence*, pages 401–415. Springer, Berlin, Heidelberg, New York, 1994.

[Mot36] Theodore S. Motzkin. *Beiträge zur Theorie der linearen Ungleichungen*. Doctoral dissertation, Universität Zürich, 1936.

[NT92] Masayuki Noro and Taku Takeshima. Risa/Asir—a computer algebra system. In Paul S. Wang, editor, *Proceedings of the ISSAC '92*, pages 387–396, Berkeley, CA, July 1992. ACM, ACM Press, New York, 1992.

[Ped91] Paul Pedersen. *Counting Real Zeros*. Ph.D. dissertation, Courant Institute of Mathematical Sciences, New York, 1991.

[PP97] Martin Peternell and Helmut Pottmann. Computing rational parametrizations of canal surfaces. *Journal of Symbolic Computation*, 23(2–3):255–266, February–March 1997.

[PRS93] Paul Pedersen, Marie-Françoise Roy, and Aviva Szpirglas. Counting real zeroes in the multivariate case. In F. Eysette and A. Galigo, editors, *Computational Algebraic Geometry*, volume 109 of *Progress in Mathematics*, pages 203–224. Birkhäuser, Boston, Basel; Berlin, 1993. Proceedings of the MEGA 92.

[PS85] Franco P. Preparata and Michael I. Shamos. *Computational Geometry—An Introduction.* Texts and monographs in computer science. Springer, New York, 1985.

[Ren92] James Renegar. On the computational complexity and geometry of the first-order theory of the reals. *Journal of Symbolic Computation*, 13(3):255–352, March 1992. Part I–III.

[Rit32] Joseph F. Ritt. *Differential Equations from the Algebraic Standpoint*, volume 14 of *AMS Colloquium Publications.* American Mathematical Society, New York, NY, 1932.

[Rit50] Joseph F. Ritt. *Differential Algebra*, volume 33 of *AMS Colloquium Publications.* American Mathematical Society, New York, NY, 1950.

[Ros85] Jaroslaw R. Rossignac. *Blending and Offsetting Solid Models.* Ph.D. thesis, Department of Electrical Engineering, College of Engineering and Applied Science, University of Rochester, Rochester, New York 14627, July 1985.

[RR86] Jaroslaw R. Rossignac and Aristides A. G. Requicha. Offsetting operations in solid modelling. *Computer Aided Geometric Design*, 3(2):129–148, August 1986.

[Sei56a] Abraham Seidenberg. An elimination theory for differential algebra. *Univ. California Publ. Math (N.S.)*, 3:31–66, 1956.

[Sei56b] Abraham Seidenberg. Some remarks on Hilbert's Nullstellensatz. *Arch. Math.*, 7:235–240, 1956.

[Sei69] Abraham Seidenberg. On k-constructable sets, k-elementary formulae, and elimination theory. *Journal für die reine und angewandte Mathematik*, 239/240:256–267, 1969.

[Stu35] Jacques Charles François Sturm. Mémoire sur la résolution des équations numériques. In *Mémoires présentés par divers Savants étrangers à l'Académie royale des sciences, section Sc. math. phys.*, volume 6, pages 273–318, 1835.

[Stu96] Thomas Sturm. Real quadratic quantifier elimination in Risa/Asir. Research Memorandum ISIS-RM-5E, ISIS, Fujitsu Laboratories Limited, 1-9-3, Nakase, Mihama-ku, Chiba-shi, Chiba 261, Japan, September 1996.

[Stu99] Thomas Sturm. Reasoning over networks by symbolic methods. *Applicable Algebra in Engineering, Communication and Computing*, 10(1):79–96, September 1999.

[SW97] Thomas Sturm and Volker Weispfenning. Rounding and blending of solids by a real elimination method. In Achim Sydow, editor, *Proceedings of the 15th IMACS World Congress on Scientific Computation, Modelling, and Applied Mathematics (IMACS 97)*, volume 2, pages 727–732, Berlin, August 1997. IMACS, Wissenschaft & Technik Verlag, Berlin, 1997.

[SW98] Thomas Sturm and Volker Weispfenning. Computational geometry problems in Redlog. In Dongming Wang, editor, *Automated Deduction in Geometry*, volume 1360 of *Lecture Notes in Artificial Intelligence (Subseries of LNCS)*, pages 58–86. Springer-Verlag, Berlin Heidelberg, 1998.

[Syl53] James Joseph Sylvester. On a theory of the syzegetic relations of two rational integral functions, comprising an application to the theory of Sturm's functions, and that of the greatest algebraical common measure. *Philosophical Transactions of the Royal Society London*, 143:407–548, 1853.

[Tar48] Alfred Tarski. A decision method for elementary algebra and geometry. Technical report, RAND, Santa Monica, CA, 1948.

[Tar51] Alfred Tarski. *A Decision Method for Elementary Algebra and Geometry*. University of California Press, Berkely, CA, 1951. (2nd edition, revised).

[Tar98] Alfred Tarski. A decision method for elementary algebra and geometry. In Bob F. Caviness and Jeremy R. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, Texts and Monographs in Symbolic Computation, pages 24–84. Springer-Verlag, Wien, Austria and New York, NY, 1998. Reprint.

[VMV94] Janos Vida, Ralph R. Martin, and Tamas Varady. A survey of blending methods that use parametric surfaces. *Computer-Aided Design*, 26(5):341–365, May 1994. Survey.

[Wan93a] Dongming Wang. An elimination method based on Seidenberg's theory and its applications. In F. Eyssette and A. Galligo, editors, *Computational Algebraic Geometry (MEGA '92)*, pages 301–328. Birkhäuser Verlag, 1993.

[Wan93b] Dongming Wang. An elimination method for polynomial systems. *Journal of Symbolic Computation*, 16(2):83–114, August 1993.

[Wan95a] Dongming Wang. Reasoning about geometric problems using an elimination method. In J. Pfalzgraf and D. Wang, editors, *Automated Practical Reasoning*, Texts and Monographs in Symbolic Computation, pages 147–185. Springer-Verlag, Wien, New York, 1995.

[Wan95b] Dongming Wang. Reasoning about geometric problems using an elimination method. In J. Pfalzgraf, editor, *Automatic Practical Reasoning*, pages 147–185. Springer-Verlag, Wien, 1995.

[Wei88] Volker Weispfenning. The complexity of linear problems in fields. *Journal of Symbolic Computation*, 5(1–2):3–27, February–April 1988.

[Wei90] Volker Weispfenning. The complexity of almost linear diophantine problems. *Journal of Symbolic Computation*, 10(5):395–403, November 1990.

[Wei92] Volker Weispfenning. Comprehensive Gröbner bases. *Journal of Symbolic Computation*, 14:1–29, July 1992.

[Wei93] Volker Weispfenning. A new approach to quantifier elimination for real algebra. Technical Report MIP-9305, FMI, Universität Passau, D-94030 Passau, Germany, July 1993.

[Wei94] Volker Weispfenning. Quantifier elimination for real algebra—the cubic case. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC 94)*, pages 258–263, Oxford, England, July 1994. ACM Press, New York, 1994.

[Wei97a] Volker Weispfenning. Complexity and uniformity of elimination in presburger arithmetic. In Wolfgang W. Küchlin, editor, *Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation (ISSAC 97)*, pages 48–53, Maui, HI, July 1997. ACM, ACM Press, New York, 1997.

[Wei97b] Volker Weispfenning. Quantifier elimination for real algebra—the quadratic case and beyond. *Applicable Algebra in Engineering Communication and Computing*, 8(2):85–101, February 1997.

[Wei97c] Volker Weispfenning. Simulation and optimization by quantifier elimination. *Journal of Symbolic Computation*, 24(2):189–208, August 1997. Special issue on applications of quantifier elimination.

[Wei98] Volker Weispfenning. A new approach to quantifier elimination for real algebra. In B.F. Caviness and J.R. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, Texts and Monographs in Symbolic Computation, pages 376–392. Springer, Wien, New York, 1998.

[Wei99] Volker Weispfenning. Mixed real-integer linear quantifier elimination. In Sam Dooley, editor, *Proceedings of the 1999 International Symposium on Symbolic and Algebraic Computation (ISSAC 99)*, pages 129–136, Vancouver, BC, July 1999. ACM, ACM Press, New York, 1999.

[Wol96] Steven Wolfram. *The Mathematica Book*. Wolfram Media and Cambridge University Press, 3rd edition, 1996.

[Wu84] Wen-Tsun Wu. Basic principles of mechanical theorem proving in elementary geometries. *Journal of Systems Sciences and Mathematical Sciences*, 4:207–235, 1984.

[Wu86a] Wen-Tsun Wu. Basic principles of mechanical theorem proving in elementary geometry. *Journal of Automated Reasoning*, 2:219–252, 1986.

[Wu86b] Wen-Tsun Wu. A mechanization method of geometry and its applications. Part 1: Distances, areas and volumes. *Journal of System Science and Mathematical Science*, 6:204–216, 1986.

[Wu87] Wen-Tsun Wu. A mechanization method of geometry and its applications. Part 3: Mechanical proving of polynomial inequalities and equations-solving. Mathematics-Mechanization Research Preprints 2, Institute of Systems Science, Academia Sinicia, Beijing, China, November 1987.

[Wu92] Wen-Tsun Wu. On problems involving inequalities. Mathematics-Mechanization Research Preprints 7, Institute of Systems Science, Academia Sinicia, Beijing, China, March 1992.