General Analysis Tool Box for Controlled Perturbation Algorithms and Complexity and Computation of Θ-Guarded Regions

Dissertation

zur Erlangung des Grades des
Doktors der Ingenieurwissenschaften
der Naturwissenschaftlich-Technischen Fakultäten
der Universität des Saarlandes

vorgelegt von

Ralf Osbild

Saarbrücken
November 2012

Kolloquium

Datum
02. August 2013

Dekan
Professor Dr. Mark Groves

Prüfungsausschuss
Professor Dr. Joachim Weickert (Vorsitzender)
Professor Dr. Dr. h.c. mult. Kurt Mehlhorn (Gutachter)
Professor Dr. Raimund Seidel (Gutachter)
Dr. Tobias Mömke (Akademischer Beisitzer)

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Saarbrücken, den

Kurzzusammenfassung

Diese Dissertation auf dem Gebiet der Algorithmischen Geometrie beschäftigt sich mit den folgenden zwei Problemen.

1. Die Implementierung von verlässlichen und effizienten geometrischen Algorithmen ist eine herausfordernde Aufgabe. Controlled Perturbation verknüpft die Geschwindigkeit von Fließkomma-Arithmetik mit einem Mechanismus, der die Verlässlichkeit garantiert. Wir präsentieren einen allgemeinen ,,Werkzeugkasten" zum *Analysieren von Controlled Perturbation Algorithmen.* Dieser Werkzeugkasten ist in unabhängige Komponenten aufgeteilt. Wir präsentieren drei alternative Methoden für die Herleitung der wichtigsten Schranken. Des Weiteren haben wir alle Prädikate, die auf Polynomen und rationalen Funktionen beruhen, sowie Objekt-erhaltende Perturbationen in die Theorie miteinbezogen. Darüber hinaus wurde der Werkzeugkasten so entworfen, dass er das tatsächliche Verhalten des untersuchten Algorithmus ohne vereinfachende Annahmen widerspiegelt.

2. Illumination und Guarding Probleme stellen ein breites Gebiet der Algorithmischen und Kombinatorischen Geometrie dar. Hierzu tragen wir die *Komplexität und Berechnung von Θ-bewachten Regionen* bei. Sie stellen eine Verallgemeinerung der konvexen Hülle dar und sind mit α-hulls und Θ-maxima verwandt. Die Schwierigkeit beim Studium der Θ-bewachten Regionen ist die Abhängigkeit ihrer Form und Komplexität von Θ. Für alle Winkel Θ beweisen wir grundlegende Eigenschaften der Region, leiten untere und obere Schranken ihrer worst-case Komplexität her und präsentieren einen Algorithmus, um die Region zu berechnen.

Abstract

This thesis belongs to the field of computational geometry and addresses the following two issues.

1. The implementation of reliable and efficient geometric algorithms is a challenging task. Controlled perturbation combines the speed of floating-point arithmetic with a mechanism that guarantees reliability. We present a general tool box for the *analysis of controlled perturbation algorithms.* This tool box is separated into independent components. We present three alternative approaches for the derivation of the most important bounds. Furthermore, we have included polynomial-based predicates, rational function-based predicates, and object-preserving perturbations into the theory. Moreover, the tool box is designed such that it reflects the actual behavior of the algorithm at hand without simplifying assumptions.

2. Illumination and guarding problems are a wide field in computational and combinatorial geometry to which we contribute the *complexity and computation of* Θ-*guarded regions.* They are a generalization of the convex hull and are related to α-hulls and Θ-maxima. The difficulty in the study of Θ-guarded regions is the dependency of its shape and complexity on Θ. For all angles Θ, we prove fundamental properties of the region, derive lower and upper bounds on its worst-case complexity, and present an algorithm to compute the region.

Danksagung

Mein Dank gilt vor allem Kurt Mehlhorn, der mich als Promotionsstudent am Max-Planck-Institut für Informatik angenommen und mir eine komplexe Aufgabenstellung anvertraut hat. Insbesondere danke ich ihm für den Freiraum, den er mir, trotz seiner konkreten Vorstellung, bei der Behandlung der Themen eingeräumt hat.

Michael Sagraloff verdanke ich speziell wichtige und anregende Impulse zur Anfangszeit. Bei Domagoj Matijević möchte ich mich für die intensive und motivierende Zusammenarbeit bedanken.

Danksagen möchte ich auch meinen Eltern, die für mich stets eine unerschöpfliche Quelle der Zuversicht und Stärkung sind. Mein innigster Dank gilt schließlich Bettina Strauß für ihr tiefes Vertrauen und verständnisvolle Geduld.

Contents

1 Introduction

This thesis belongs to the field of computational geometry and addresses two issues, which we introduce briefly below.

The implementation of reliable and efficient geometric algorithms is a challenging task. The reason is the following conflict: On the one hand, computing with rounded arithmetic may call into question the reliability of programs while, on the other hand, computing with exact arithmetic may be too expensive and hence inefficient. Many schemes have been suggested to bridge over this gap. One suggestion is the implementation of controlled perturbation algorithms which combines the speed of floating-point arithmetic with a protection mechanism that guarantees reliability. The first topic of this thesis is concerned with the performance analysis of controlled perturbation algorithms in theory. We answer this question with the presentation of

a general analysis tool box for controlled perturbation algorithms.

Our tool box is separated into independent components which are presented individually with their interfaces. This way, the tool box supports alternative approaches for the derivation of the most crucial bounds: We present three approaches for this task. Even more, the framework can be applied to different perturbation policies. Furthermore, we have thoroughly reworked the concept of controlled perturbation in order to include rational function-based predicates into the theory; former research on the topic and, especially, polynomial-based predicates are included into the framework anyway. Moreover, the tool box is designed such that it reflects the floating-point behavior of the controlled perturbation algorithm at hand.

Illumination and guarding problems are another wide field in computational and combinatorial geometry. Problems of this class are often referred to for the modelling of real-world problems for two reasons: first, to find an algorithmic solution for the problem at hand and, second, to determine the complexity of the problem at hand. Our second topic contributes the

complexity and computation of Θ*-guarded regions*

to this area. Θ-guarded regions are related to well-known geometrical structures: They are a generalization of the convex hull, are equivalent to the α-hulls in certain situations, and can be derived from Θ-maxima for certain angles. The difficulty in the study of Θ-guarded regions is that their shape and complexity vary with the angle Θ. For all cases of Θ, we prove fundamental properties of the Θ-guarded region, prove lower and upper bounds on its worst-case complexity, and present an algorithm to compute the region.

1.1 General Analysis Tool Box for Controlled Perturbation Algorithms

It is a notoriously difficult task to cope with rounding errors in computing [32, 20]. In computational geometry, predicates are decided on the sign of mathematical expressions. If rounding errors cause a wrong decision of the predicate, geometric algorithms may fail in various ways: inconsistency of the data (e.g., contradictory topology), loops that do not terminate, or loops that terminate unexpectedly [50]. In addition, the thoughtful processing of degenerate cases makes the implementation of geometric algorithms laborious [8]. The meaning of degeneracy always depends on the context (e.g., three points on a line, four points on a circle). There are several ways to overcome the numerical robustness issues and to deal with degenerate inputs.

The *exact computation paradigm* [47, 48, 57, 33, 79, 58] suggests an implementation of an exact arithmetic. This is established by a number representation of variable precision (i.e., variable bit length) or by the use of symbolic values which are not evaluated (e.g., roots of integers). There are several implementations of such number types [14, 54, 64, 63, 58]. Each program must be developed carefully such that it can deal with all possible degenerate cases. The software libraries Leda and Cgal follow the exact computation paradigm [58, 51, 28]. The paradigm was also taken as a basis in [7, 73, 38].

As opposed to that, the *topology oriented approach* [74, 45, 75] is based on an arithmetic of finite precision. To avoid numerical robustness issues, the main guideline is the maintenance of the topology. This objective requires individual alterations of the algorithm at hand, and it seems that it cannot be turned into an easy-to-use general framework. Furthermore, this approach must also cope with degenerate inputs. However, the speed of floating-point arithmetic may be worth the trouble; in addition to other accelerations, Held [42] has implemented a very fast computation of the Voronoi diagram of line segments.

There are also *problem-oriented* solutions. In computational geometry, the sign of determinants decides an interesting class of predicates. For example, the side-of-line or the in-circle predicate in the plane belong to this class and are used in the computation of Delaunay diagrams. Some publications deal with the numerical issues in the evaluation of determinants directly [5, 9].

The previous approaches have in common that they primarily focus on the numerical issues. Other approaches are originated from the degeneracy issue. A slight perturbation of the input seems to solve this problem. There are different approaches based on perturbation. The *symbolic perturbation* [23, 78, 77, 25, 26, 72, 61] provides a general way to distort inputs such that degeneracies do not occur. This definitely provides a shorter route for the presentation of geometric algorithms. Practically, this approach requires exact arithmetic to avoid robustness issues. Therefore, the pitfall in this approach is that if the concept requires very small perturbations, it may implicate a high precision and a possibly slow implementation.

In this paper, we focus on *controlled perturbation.* It was introduced by Halperin et al. [41] for the computation of spherical arrangements. In the context of controlled perturbation, a perturbed input is a random point in the neighborhood of the initial input. The basic module of the algorithm is a repeating perturbation process with two objectives: finding an input that does not contain degeneracies and that leads to numerically robust floating-point evaluations. Halperin et al. have presented mechanisms to respond to inappropriate perturbations. Moreover, they have argued formally under which conditions there is a chance for a successful termination of their algorithm. *Controlled perturbation leads to numerically robust implementations of algorithms which use non-exact arithmetic and which do not need to process degenerate cases.*

This idea of controlled perturbation has since been applied to further geometric problems: the arrangement of polyhedral surfaces [40], the arrangement of circles [39], Voronoi diagrams and Delaunay triangulations [52, 35]. However, the presentation of each specific algorithm has required a specific analysis of its performance. This broaches the subject of a *general method* to analyze controlled perturbation algorithms.

We remark that controlled perturbation also has a drawback: Although it solves the problem for the perturbed input exactly, it does not necessarily solve it for the initial input. Furthermore, it is non-obvious how to derive the solution for the initial input from solutions of perturbed inputs in general. In case the input is highly degenerated, the running time of the algorithm may increase significantly after the permutation [12, 4]. In this case, the specialized treatment of degeneracies may be much faster.

Our contribution

The study of a general method to analyze controlled perturbation algorithms is a joint work with Kurt Mehlhorn and Michael Sagraloff. We first presented the idea in [59]. Then Caroli [13] studied the applicability of the method for predicates used for the computation of arrangements of circles (according to [39]) and for the computation of Voronoi diagrams of line segments (according to [10, 71]). Our significantly improved journal article contains, furthermore, a detailed discussion of the analysis of multivariate polynomials [60].

This thesis redevelops the topic completely to design a sophisticated tool box for the analysis of controlled perturbation algorithms. The tool box is valid for floating-point arithmetic, guides the user step by step through the analysis, and allows alternative components. We want to emphasize that former research fits into this refined framework. Moreover, the solution to an open problem is integrated into the theory. We briefly present our achievements below.

We present a general tool box to analyze algorithms and their predicates. The tool box is subdivided into independent components and their interfaces. Step-by-step instructions for the analysis are associated with each component. Interfaces represent the bounds that are used in the analysis. The components of the tool box can be chosen such that the result is a precision function or a probability function.

Furthermore, necessary conditions for the analysis are derived from the interfaces.

We present alternative approaches to derive necessary bounds. Because we have subdivided the tool box into independent components and their interfaces, it is possible to make alternative components available in the most crucial step of the analysis. The *direct approach* is based on the geometric meaning of predicates, the *bottom-up approach* is based on the composition of functions, and the *top-down approach* is a coordinate-wise analysis of functions. Similar direct and top-down approaches are presented in [59, 60]. This is the first time that a bottom-up approach is presented for this task.

The result of the analysis is valid for floating-point arithmetic. A random floating-point number generator that guarantees a uniform distribution was introduced in [60]. However, the result of the analysis has not yet been proven valid for the finite set of floating-point numbers since the Lebesgue measure cannot take sets of measure zero into account. To overcome this problem, we define a specialized perturbation generator and pay attention to the finiteness in the analysis, namely, in the success probability, in the (non-)exclusion of points, and in the usage of the Lebesgue measure.

We present an alternative analysis of multivariate polynomials. An analysis of multivariate polynomials, which resembles the top-down approach, is presented in [60]. We present an alternative analysis here which makes use of the bottom-up approach.

We solve the open problem of analyzing rational functions. We include poles of rational functions into the theory and describe the treatment of floating-point range errors in the analysis. We suggest a general way to guard rational functions in practice, and we describe how to analyze the behavior of these guards in theory.

We integrate different perturbation policies into the analysis. We present a perturbation generator that makes it possible to perturb the location of input objects with or without deforming the objects themselves. To achieve this goal, we have designed the perturbation process such that the relative floating-point input specifications of the objects can be preserved despite using rounded arithmetic.

We suggest an implementation that is in accordance with the analysis tool box. We define a fixed-precision perturbation generator and extend it to be object-preserving. We explain the particularities in the practical treatment of range errors that occur especially in the case of rational functions. Finally, we show how to realize guards for rational functions.

1.2 Complexity and Computation of Θ-Guarded Regions

The second chapter belongs to the fields of computational and combinatorial geometry. In contrast to the first chapter, some terms have totally different meanings here: The term *guard* refers to a point in the plane, the term *region* refers to the Θ-guarded region, and by π we denote Archimedes' constant.

Illumination and guarding problems are another wide field in computational and combinatorial geometry. One instance in this class of problems is the classical one posed by Victor Klee [66]: *How many guards are necessary, and how many are sufficient, to patrol the paintings and works of art in an art gallery with n walls?* While this particular problem was solved shortly thereafter by Chvátal [16], proving a tight $\lfloor \frac{n}{3} \rfloor$ bound, many variants of guarding problems appeared subsequently. A survey on the topic is contained in [76].

In this chapter of the thesis, we consider the following planar guarding problem. We are given a finite point set G in the plane, whose points we call *guards*, and an angle $\Theta \in [0, 2\pi]$. A Θ-cone is a cone with the apex angle Θ. We say that the Θ-cone is *empty (with respect to G)* if it does not contain any point of G in its interior. A point $p \in \mathbb{R}^2$ is Θ-*guarded* if every Θ-cone whose apex is located at p is non-empty. Furthermore, the set of all Θ-guarded points is called the Θ-*guarded region*, or the Θ-*region* for short.

Previous work

For a given set G of n points in the plane, Avis et al. [3] were the first to introduce the notion of *unoriented* Θ-*maxima.* They say that some point $g \in G$ is a Θ-maxima if there exists an empty Θ-cone with its apex at g. Hence, a point g is Θ-maxima if it is not Θ-guarded with respect to G. They present an $O(\frac{n}{\Theta} \log n)$ algorithm for computing the unoriented Θ-maximum of the set G, or in other words, an algorithm to query each point in G whether it is Θ-guarded or not. A slight variation of their algorithm can in fact query any finite point set P in $O(\frac{n+|P|}{\Theta} \log(n + |P|))$ time, as we show in Lemma 3.15. They further show that the unoriented $\frac{\pi}{2}$-maxima can be computed in $O(n)$ expected time.

Abellanas et al. [1] extend the guarding model, which they refer to as *good* Θ-*illumination*, by a range r, i.e., a guard $g \in G$ can only guard points inside the circle of radius r that is centered at g. Besides other results, they show how to check if a query point p is Θ-guarded in $O(n)$ time and how to output the necessary range and guards as witnesses.

Since the π-region is equivalent to the standard convex hull of a point set, we can also regard the Θ-region as a generalization of the convex hull. Several generalizations have been proposed, like the α-hull [22], the k-th iterated hull [15], and the related concept of the k-hull (k-depth contour) [17].

Our contribution

This chapter is based on the journal article [55] which is a joint work with Domagoj Matijević. Furthermore, an improved computation of the Θ-region for $\frac{\pi}{2} \leq \Theta < \pi$ in $O(n \log n)$ time is contributed in this thesis.

The difficulty in the study of Θ-guarded regions is that their shape and complexity vary with the angle Θ. First, we consider the case $\Theta \geq \pi$. We analyze the shape of the Θ-guarded region, show its general relation to the convex hull, prove that its complexity equals the complexity of the convex hull, show its specific relation to positive α-hulls for certain guard sets and angle ranges, and develop an easy and efficient $O(n \log n)$ time algorithm to compute its boundary.

In the main part, we consider the case $\Theta < \pi$. For these angles the problem becomes much more involved, and the boundary of the Θ-region becomes more difficult to understand. We show that the boundary of the Θ-region is contained in an arrangement of circular arcs. Further, we bound this set of circular arcs by $O(\frac{n}{\Theta})$. In case $\frac{\pi}{2} \leq \Theta < \pi$, we prove that the complexity of the Θ-region is $O(n)$ and present an algorithm to compute the region in $O(n \log n)$ time. For smaller angles $\delta \leq \Theta < \frac{\pi}{2}$, where δ is a positive constant, we show that the complexity is $O(n^{1+\varepsilon})$ for any $\varepsilon > 0$. In case we consider the asymptotic complexity bound in n and $\frac{1}{\Theta}$, we prove $O(\frac{n^2}{\Theta^2})$. Furthermore, we construct a generic example for this case which has complexity $\Omega(n^2)$. Finally, we present an algorithm to compute the Θ-region for $\Theta < \frac{\pi}{2}$ and an analysis of its running time.

We would like to note that there is an independent publication by Abellanas et al. [2] on the same topic. It claims that the complexity of the Θ-region (called the α-embracing contour) is claimed to be $O(n)$ for all constant Θ, and an algorithm that runs in $O(n^2 \log n)$ time and $O(n^2)$ space is proposed. After personal communication with the authors, we agree that the claims are unfortunately not generally true for small angles.

2 General Analysis Tool Box for Controlled Perturbation Algorithms

In this chapter, we present a tool box for the general analysis of controlled perturbation algorithms. In Section 2.1, we present the basic design principles of controlled perturbation from a practical point of view. Fundamental quantities and definitions of the analysis are introduced in Section 2.2. In addition, we point to the difficulty to validate the result of the analysis for floating-point arithmetic. The *general analysis tool box* and all of its components are briefly introduced in Section 2.3. Geometric algorithms base their decisions on geometric predicates which are decided by signs of real-valued functions. Therefore, the analysis of algorithms requires a general analysis of such functions. The presentation is structured in two parts: the *function analysis* and the *algorithm analysis.*

The function analysis is performed with real arithmetic and works in two stages. The required bounds form the interface between the stages and are presented in Section 2.4. The *method of quantified relations* represents the actual analysis in the second stage and is introduced in Section 2.5. The derivation of the bounds in the first stage follows the *direct approach* from Section 2.6, the *bottom-up approach* from Section 2.7, or the *top-down approach* from Section 2.8, together with an *error analysis*, which is introduced in Section 2.9. In Section 2.10, we extend the analysis and the implementation such that both properly deal with floating-point range errors. As examples, we present the analysis of *multivariate polynomials* in Section 2.7 and the analysis of *rational functions* in Section 2.11. The function analysis is visualized in Figure 2.7 on Page 35.

The algorithm analysis is also performed in two stages. In the first stage, we make use of the introduced function analysis and derive some algorithm specific bounds. The analysis itself in the second stage is represented by the *method of distributed probability.* The algorithm analysis is presented in Section 2.12 and is visualized in Figure 2.25 on Page 89.

Furthermore, we present a general way to *implement* controlled perturbation algorithms in Section 2.13 such that our analysis tool box can be applied to them. Even more, we suggest a way to implement *object-preserving perturbations* in Section 2.14. A *quick reference* to the most important definitions of this chapter can be found in Section 2.15.

2.1 Controlled Perturbation Algorithms

This section contains an introduction to the basic principles for controlled perturbation algorithms. We already have mentioned that implementations of geometric algorithms must address degeneracy and numerical robustness issues. We review floating-point arithmetic in Section 2.1.1 and present the basic design principles of controlled perturbation algorithms in Section 2.1.2.

2.1.1 Floating-point Arithmetic

Variable precision arithmetic is necessary for a general implementation of controlled perturbation algorithms. We explain this statement with the following thought experiment,[1] which can be skipped during the first reading: Assume we compute an arrangement of n circles incrementally with a fixed precision arithmetic. Let us further assume that there is an upper bound on the radius of the circles. Then, because of the fixed precision, the number of distinguishable intersections per circle must be limited. Hence, the computation of dense arrangements cannot work in general unless we allow circles to be moved (perturbed) further away from their initial position. Asymptotically, this policy would transform a very dense arrangement into an arrangement of almost uniformly distributed circles. Therefore, we demand that the precision of the arithmetic can be chosen arbitrarily large.

A *floating-point number* is given by a sign, a mantissa, a radix, and a signed exponent. In the regular case, its value is defined as

$$\text{value} \quad := \quad \text{sign} \cdot \text{mantissa} \cdot \text{radix}^{\text{exponent}}.$$

Without loss of generality, we assume the radix to be 2. The bit length of the mantissa is called *precision L*. We denote the *bit length of the exponent* by K. The discrete set of regular floating-point numbers is a subset of the rational numbers. Furthermore, this set is finite for fixed L and K.

A *floating-point arithmetic* defines the number representation (the radix, L and K), the operations, the rounding policy, and the exception handling for floating-point numbers (see Goldberg [37]). A technical standard for fixed precision floating-point arithmetic is IEEE 754-2008 (see [44]). Nowadays, the built-in types single, double and quadruple precision are usual for radix 2.

There are several software libraries that offer *variable*[2] *precision floating-point arithmetic*. CGAL provides the multi-precision floating-point number type `MP_Float` (see the CGAL manual [14]). CORE provides the variable precision floating-point number type `CORE::BigFloat` (see [54]). And LEDA provides the variable precision floating-point number type `leda_bigfloat` (see the LEDA book [58]). Be aware that the rounding policy and exception handling of certain libraries may differ from the

[1] This consideration is absolutely conform to Halperin et al. [39]: If the augmented perturbation parameter δ exceeds a given threshold Δ, the precision is augmented and δ is reset.

[2] With variable we subsume all types of arithmetic that support arbitrarily large precisions. Some are called variable precision, multiple precision or arbitrary precision.

IEEE standard. Since our analysis partially presumes this standard,[3] we must ensure that the arithmetic in use is appropriate. The Gnu Multiple Precision Floating-Point Reliable Library, for example, "provides the four rounding modes from the IEEE 754-1985 standard, plus away-from-zero, as well as for basic operations as for other mathematical functions" (see the Gnu Mpfr manual [64]). Moreover, Gnu Mpfr is used for the multiple precision interval arithmetic, which is provided by the Multiple Precision Floating-point Interval library (see the Gnu Mpfi manual [63]).

Variable precision arithmetic is more expensive than built-in fixed precision arithmetic. In practice, we try to solve the problem at hand with built-in arithmetic first and, in addition, try to make use of floating-point filters. We use the following notations throughout the chapter.

Definition 2.1 (floating-point)**.** Let $L, K \in \mathbb{N}$. By $\mathbb{F}_{L,K}$ we denote:

1. The set of floating-point numbers with radix 2, precision L, and K-bit exponent.
2. The floating-point arithmetic that is induced by the set characterized in 1.

Furthermore, we define the suffix $|_{\mathbb{F}}$ for sets and expressions:

1. Let $k \in \mathbb{N}$, and let $X \subset \mathbb{R}^k$. Then $X|_{\mathbb{F}} := X \cap \mathbb{F}^k$.
2. $f(x)|_{\mathbb{F}}$ denotes the floating-point value of $f(x)$ evaluated with arithmetic $\mathbb{F}$.

That means, we denote by $X|_{\mathbb{F}}$ the restriction of X to its subset that can be represented with floating-point numbers in $\mathbb{F}$. To simplify the notation, we omit the indices L or K of $\mathbb{F}_{L,K}$ whenever they are given by the context. For the same reason, we have already skipped the dimension k in the suffix $|_{\mathbb{F}}$.

2.1.2 Basic Controlled Perturbation Implementations

Rounding errors of floating-point arithmetic may influence the result of predicate evaluations. Wrong predicate evaluations may cause erroneous results of the algorithm and even lead to non-robust implementations (see Kettner et al. [50]). In order to get correct and robust implementations, we introduce guards which testify the reliability of predicate evaluations (see [34, 11, 60]).

Definition 2.2 (guard)**.** Let $\mathbb{F}$ be a floating-point arithmetic, and let $f : X \to \mathbb{R}$ be a function with $X \subset \mathbb{R}^k$. We call a predicate $\mathcal{G}_f : X \to \{\text{true, false}\}$ a *guard for f on X* if

$$\mathcal{G}_f(x) \text{ is true} \qquad \Rightarrow \qquad \operatorname{sign}(f(x)|_{\mathbb{F}}) = \operatorname{sign}(f(x))$$

for all $x \in X|_{\mathbb{F}}$. Presuming that there is such a predicate $\mathcal{G}_f$, we say that an input $x \in X|_{\mathbb{F}}$ is *guarded* if $\mathcal{G}_f(x)$ is true and *unguarded* if $\mathcal{G}_f(x)$ is false.

That means, a guard confirms that the sign of the function evaluation is correct. A design of guards is presented in Section 2.9. By means of guards, we can implement geometric algorithms such that they can either verify or disprove their results.

[3] A standardized behavior of floating-point operations is presumed in Section 2.9.

Definition 2.3 (guarded algorithm)**.** We call an algorithm $\mathcal{A}_{\mathrm{G}}$ a *guarded algorithm* if there is a guard for each predicate evaluation and if the algorithm halts either with the correct combinatorial result or with the information that a guard has failed. If $\mathcal{A}_{\mathrm{G}}$ halts with the correct result, we also say that $\mathcal{A}_{\mathrm{G}}$ is *successful*, and we say that $\mathcal{A}_{\mathrm{G}}$ has *failed* if a guard has failed.

Let $\bar{y}$ be an input of $\mathcal{A}_{\mathrm{G}}$. In case $\mathcal{A}_{\mathrm{G}}(\bar{y})$ is successful, we obtain the desired result for input $\bar{y}$. Of course, the situation is unsatisfying if $\mathcal{A}_{\mathrm{G}}$ fails. Therefore, we introduce controlled perturbation (see Halperin et al. [39]): We execute $\mathcal{A}_{\mathrm{G}}$ for randomly perturbed inputs y (i.e., random points in the neighborhood of $\bar{y}$) *until* $\mathcal{A}_{\mathrm{G}}$ terminates successfully. Furthermore, we increase the precision L of the floating-point arithmetic $\mathbb{F}$ after each failure in the hope to improve the chance to succeed. (It is the task of the analysis to provide evidence.) We summarize this idea in the provisional controlled perturbation algorithm basic-$\mathcal{A}_{\mathrm{CP}}$, which is shown in Algorithm 1. The general controlled perturbation algorithm is presented on page 98, Section 2.12.

Algorithm 1 : basic-$\mathcal{A}_{\mathrm{CP}}(\mathcal{A}_{\mathrm{G}}, \bar{y}, \mathcal{U}_\delta)$

```
/* initialization */
$L \leftarrow$ precision of built-in floating-point arithmetic

repeat
  /* run guarded algorithm */
  $y \leftarrow$ random point in $\overline{\mathcal{U}}_\delta(\bar{y})|_{\mathbb{F}_L}$
  $\omega \leftarrow \mathcal{A}_{\mathrm{G}}(y, \mathbb{F}_L)$

  /* adjust parameters */
  if $\mathcal{A}_{\mathrm{G}}$ failed then
    $L \leftarrow 2L$
  end if
until $\mathcal{A}_{\mathrm{G}}$ succeeded

/* return perturbed input $y$ and result $\omega$ */
return $(y, \omega)$
```

We see that there is an implementation of basic-$\mathcal{A}_{\mathrm{CP}}(\mathcal{A}_{\mathrm{G}})$ for every guarded algorithm $\mathcal{A}_{\mathrm{G}}$, or in other words, for every algorithm that is only based on geometric predicates that can be guarded. It is important to note that this does not necessarily imply that basic-$\mathcal{A}_{\mathrm{CP}}$ performs well. It is the main objective of this chapter to develop a general method to analyze the performance of controlled perturbation algorithms $\mathcal{A}_{\mathrm{CP}}$.

2.2 Fundamental Quantities and Definitions

Before we begin with the analysis, we introduce some fundamental quantities. The situation we want to analyze is defined in Section 2.2.1. We encounter and discuss many issues during the definition of the success probability in Section 2.2.2. We see that there are two important dependencies:

1. The precision controls the floating-point error.

2. The precision controls the density of the floating-point numbers.

In Section 2.2.3, we introduce controlled perturbation-specific quantities and focus on the first dependency. In Section 2.2.4, we learn how to embed the second dependency in the analysis. The overview in Section 2.2.5 summarizes the classification of inputs in practice and in the analysis. In Section 2.2.6, we present conditions under which we may *apply* controlled perturbation to a predicate in practice and under which we can actually *justify* its application in theory.

2.2.1 Perturbation, Predicate, Function

We next define the quantities that are needed to describe the initial situation: the original input, the perturbation area, the perturbation parameter, the perturbed input, the input value bound, functions that realize geometric predicates, and predicate descriptions.

In the analysis, we assume that the *original input* $\bar{y}$ of a controlled perturbation algorithm $\mathcal{A}_{\mathrm{CP}}$ consists of n floating-point numbers, that is, $\bar{y} \in \mathbb{F}^n$ or, as we prefer to say, $\bar{y} \in \mathbb{R}^n|_{\mathbb{F}}$. At this point, we do not care for a geometrical interpretation of the input of $\mathcal{A}_{\mathrm{CP}}$. We remark that this is no restriction: a complex number can be represented by two numbers; a vector can be represented by the sequence of its components; geometric objects can be represented by their coordinates and measures; and so on. A circle in the plain, for example, can be represented by a 6-tuple (the coordinates of three distinct points in the circle) or a 3-tuple (the coordinates of the center and the radius). And, continuing the example, an input of m circles can be interpreted as a tuple $\bar{y} \in \mathbb{R}^n|_{\mathbb{F}}$ with $n := 6m$ if we choose the first variant.

We define the *perturbation of* $\bar{y}$ as a random additive distortion of its components.[4] We call $\mathcal{U}_\delta(\bar{y}) \subset \mathbb{R}^n$ a *perturbation area* with *perturbation parameter* δ if

1. $\delta \in \mathbb{R}^n_{>0}$,
2. $y \in \mathcal{U}_\delta(\bar{y})$ implies $|y_i - \bar{y}_i| \leq \delta_i$ for $1 \leq i \leq n$, and
3. $\mathcal{U}_\delta(\bar{y})$ contains an (open) neighborhood of $\bar{y}$.

Note that $\mathcal{U}_\delta(\bar{y})$ is not a discrete set, whereas $\mathcal{U}_\delta(\bar{y})|_{\mathbb{F}}$ is finite. In our example, if we allow a circular perturbation of the $3m$ points which define the m input circles, the perturbation area is the Cartesian product of $3m$ planar discs. We make the observation that even if we consider the input as a plain sequence of numbers, the

[4] There is no unique definition of perturbation in geometry (see the introduction in [72]).

perturbation area may look very special—we cannot neglect the geometrical interpretation here. In this context, we define an *axis-parallel perturbation area* $U_\delta(\bar{y})$ as a box which is centered in $\bar{y}$ and has edge length $2\delta_i$ parallel to the i-th main axis (and always denote it by the latin letter U instead of $\mathcal{U}$). This definition significantly simplifies the shape of the perturbation area.

The perturbed input must also be a vector of floating-point numbers. For now, we denote the *perturbed input* by $y \in \mathcal{U}(\bar{y})|_{\mathbb{F}}$. (This definition is refined on page 23).

The analysis of $\mathcal{A}_{\mathrm{CP}}$ depends on the analysis of $\mathcal{A}_{\mathrm{G}}$ and its predicates (see Section 2.12). We remember that a *geometric predicate*, which is either true or false, is decided by the sign of a *real-valued function* f. Therefore, we introduce further quantities to describe such functions. We assume that f is a k-ary real-valued function and that $k \ll n$. We further assume that we evaluate f at k distinct perturbed input values, so we evaluate $f(y_{\sigma(1)}, \dots, y_{\sigma(k)})$ where $\sigma : \{1, \dots, k\} \to \{1, \dots, n\}$ is injective. The mapping σ is injective to guarantee that the variables in the formula of f are independent of each other. In order not to confuse the indices in the analysis, we change the names in the argument list of f in $x_i := y_{\sigma(i)}$ for $1 \leq i \leq k$. In the same way, we also rename the affected input values $\bar{x}_i := \bar{y}_{\sigma(i)}$. We denote the set of *valid arguments for* f by A.

In the analysis, $e_{\max}$ implicitly describes an upper-bound on the absolute value of perturbed input values where

$$e_{\max} \quad := \quad \min\left\{ e' \in \mathbb{N} \,:\, |\bar{y}_i| + \delta_i \leq 2^{e'} \text{ for all } 1 \leq i \leq n \right\}. \tag{2.1}$$

We call $e_{\max}$ the *input value parameter*. Be aware that this is just a bound on the arguments of f and not a bound on the absolute value of f. At the moment, we assume that the absolute value of f is bounded on A and that the size K of the exponent of the floating-point arithmetic $\mathbb{F}_{L,K}$ is sufficiently large to avoid overflow errors during the evaluation of f. In Section 2.10, we abandon this assumption and discuss the treatment of range issues.

Below, we summarize the basic quantities needed for the analysis of a function f.

Definition 2.4. We call $(f, k, A, \delta, e_{\max})$ a *predicate description* if:

1. $k \in \mathbb{N}$,
2. $A \subset \mathbb{R}^k$,
3. $\delta \in \mathbb{R}^k_{>0}$,
4. $e_{\max}$ is as it is defined in Formula (2.1),
5. $\bar{U}_\delta(A) \subset [-2^{e_{\max}}, 2^{e_{\max}}]^k$, and
6. $f : \bar{U}_\delta(A) \to \mathbb{R}$.

Predicate descriptions are frequently used in this chapter. We extend the notion in Definition 2.9 on page 29 and in Definition 2.10 on page 30.

2.2.2 Success Probability, Grid Points

The controlled perturbation algorithm $\mathcal{A}_{\mathrm{CP}}$ eventually terminates if there is a positive probability that $\mathcal{A}_{\mathrm{G}}$ terminates successfully. The latter condition is fulfilled if

f has the property: The probability of a successful evaluation of f gets arbitrarily close to the certain event by merely increasing the precision L. We call this property *applicability* and specify it in Section 2.2.6.

In this section we derive a definition for the success probability that is appropriate for the analysis and that is valid for floating-point evaluations. We begin with the question: What is the least probability that a guarded evaluation of f is successful in a run of $\mathcal{A}_\mathrm{G}$ under the arithmetic $\mathbb{F}$? We assume that each random point is chosen with the same probability. Then the answer is

$$\mathrm{pr}(f|_\mathbb{F}) \quad := \quad \min_{\bar{x} \in A} \frac{\left| \left\{ x \in \bar{U}_\delta(\bar{x})|_\mathbb{F} \,:\, \mathcal{G}(x) \text{ is true} \right\} \right|}{\left| \bar{U}_\delta(\bar{x})|_\mathbb{F} \right|}.$$

Indeed, the definition reflects the actual behavior of f. The probability is the number of guarded (floating-point) inputs divided by the total number of inputs and considers the worst-case for all perturbation areas.

Issue 1: Floating-point arithmetic is hard to analyze directly.

Because floating-point arithmetic and its rounding policy can hardly be analyzed directly, we aim to derive a corresponding formula for real arithmetic. In real space, we use the Lebesgue measure[5] μ to determine the volume of areas. Therefore, we are looking for a formula like

$$\mathrm{pr}(f) \quad := \quad \min_{\bar{x} \in A} \frac{\mu\left(\left\{ x \in \bar{U}_\delta(\bar{x}) \,:\, \mathcal{G}'(x) \text{ is true} \right\} \right)}{\mu(\bar{U}_\delta(\bar{x}))} \tag{2.2}$$

where the predicate $\mathcal{G}' : \bar{U}_\delta(A) \to \{\text{true, false}\}$ equals $\mathcal{G}$ at arguments with floating-point representation.

Issue 2: The set of floating-point numbers has measure zero.

It is well-known that the set $\bar{U}_\delta(\bar{x})|_\mathbb{F}$ is finite and that its superset $\bar{U}_\delta(\bar{x})|_\mathbb{Q}$ is a set of measure zero. Be aware that the fraction in Formula (2.2) does not change if we redefine f on a set of measure zero. This implies possible deviations from normal situations. For example,[6] let $f_\text{false} : \bar{U}_\delta(A) \to \mathbb{R}$ be

$$f_\text{false}(x) \quad := \quad \begin{cases} f(x) & : \quad x \notin \bar{U}_\delta(A)|_\mathbb{Q} \\ 0 & : \quad \textit{otherwise}, \end{cases}$$

and let $f_\text{true} : \bar{U}_\delta(A) \to \mathbb{R}$ be

$$f_\text{true}(x) \quad := \quad \begin{cases} f(x) & : \quad x \notin \bar{U}_\delta(A)|_\mathbb{Q} \\ B & : \quad \textit{otherwise} \end{cases}$$

[5] Measure Theory: The Lebesgue measure is defined in Forster [31].

[6] Note that there are finite sets of exceptional points that lead to similar counter-examples since every exception influences the practical behavior of the function (and L is finite).

where $B \in \mathbb{R}_{>0}$ is large enough to guarantee that the guard $\mathcal{G}$ evaluates to true in the latter case. Be aware that $\mathrm{pr}(f_{\text{false}}) = \mathrm{pr}(f_{\text{true}})$ due to Formula (2.2), whereas both implementations "$\mathcal{A}_{\mathrm{G}}$ with f_{true}" and "$\mathcal{A}_{\mathrm{G}}$ with f_{false}" are conflictive: The former is always successful, whereas the latter never succeeds. We remark that the assumption "f is (upper) continuous almost everywhere" does not solve the issue because "almost everywhere" means "with the exception of a set of measure zero." We introduce several restrictions to be able to deal with situations of this kind.

Issue 3: There is no general relation between $\mathrm{pr}(f|_{\mathbb{F}})$ and $\mathrm{pr}(f)$.

This problem already becomes visible in the 1-dimensional case.

Example 2.1. Let $\mathbb{F} = \mathbb{F}_{2,3}$ be the floating-point arithmetic with $L = 2$ and $K = 3$. In addition, let $U = [0, 2]$, $R_1 = [0, 1]$, and $R_2 = [1, 2]$ be intervals. The situation is depicted in Figure 2.1.

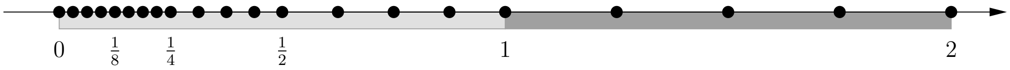


Figure 2.1: Distribution of the discrete set $\mathbb{F}_{2,3}$ within the interval $[0, 2]$.

What is the probability that a randomly chosen point $x \in U$ lies inside of R_1, respectively R_2, for points in U or $U|_{\mathbb{F}}$? Note that R_1 and R_2 have the same length. For $R_1 = [0, 1]$, we have

$$\mathrm{pr}(R_1) = \frac{1}{2} \quad < \quad \mathrm{pr}(R_1|_{\mathbb{F}}) = \frac{17}{21},$$

so that the probability is higher for floating-point arithmetic. On the other hand, for $R_2 = [1, 2]$, we have

$$\mathrm{pr}(R_2) = \frac{1}{2} \quad > \quad \mathrm{pr}(R_2|_{\mathbb{F}}) = \frac{5}{21},$$

which means that the probability is higher for real arithmetic. ○

We derive from Example 2.1 that there is no general relation between $\mathrm{pr}(f|_{\mathbb{F}})$ and $\mathrm{pr}(f)$ because of the distribution of $\mathbb{F}$.

Issue 4: The distribution of $\mathbb{F}$ is non-uniform.

Because the discrete set of floating-point numbers is non-uniformly distributed in general, we modify the perturbation policy: We restrict the random choice of floating-point numbers to selected numbers that lie on a regular grid.

Definition 2.5 (grid)**.** Let $e_{\max}$ be as it is defined in Formula (2.1), and let $\mathbb{F}_{L,K}$ be a floating-point arithmetic (with $e_{\max} \ll 2^{K-1}$). We define

$$\tau \quad := \quad 2^{e_{\max}-L-1}. \tag{2.3}$$

We call

$$\mathbb{G}_{L,K,e_{\max}} \quad := \quad \{\lambda\tau \,:\, \lambda \in \mathbb{Z} \text{ and } \lambda\tau \in [-2^{e_{\max}}, 2^{e_{\max}}]\} \tag{2.4}$$

the *grid points induced by* $e_{\max}$ *with respect to* $\mathbb{F}_{L,K}$, and we call τ the *grid unit of* $\mathbb{G}_{L,K,e_{\max}}$. Furthermore, we denote the grid points $\mathbb{G}$ inside of a set $X \subset \mathbb{R}^k$ by

$$X|_{\mathbb{G}} \quad := \quad X \cap \mathbb{G}^k.$$

We again omit the indices wherever they do not require special attention. We observe that the grid unit τ is the maximum distance between two adjacent points in $\mathbb{F} \cap [-2^{e_{\max}}, 2^{e_{\max}}]$. We further observe that the grid points $\mathbb{G}$ form a subset of $\mathbb{F}$. Be aware that the symbol $\mathbb{F}$ represents a set or an arithmetic, whereas the symbol $\mathbb{G}$ always represents a set. It is important to see that the underlying arithmetic is still $\mathbb{F}$. We have introduced $\mathbb{G}$ only to change the definition of the *original perturbation area* to $\overline{\mathcal{U}}_\delta(\bar{y})|_{\mathbb{G}}$. This leads to the *final version of the success probability of* f: The least probability that a guarded evaluation of f is successful for inputs in $\mathbb{G}$ under the arithmetic $\mathbb{F}$ is

$$\mathrm{pr}(f|_{\mathbb{G}}) \quad := \quad \min_{\bar{x} \in A} \frac{\left|\,\left\{x \in \bar{U}_\delta(\bar{x})|_{\mathbb{G}} \,:\, \mathcal{G}(x) \text{ is true}\right\}\,\right|}{\left|\,\bar{U}_\delta(\bar{x})|_{\mathbb{G}}\,\right|}. \tag{2.5}$$

Remark 2.1. How can we implement the random perturbation in $\overline{\mathcal{U}}_\delta(\bar{y})|_{\mathbb{G}}$? Because the points in $\mathbb{G}$ are uniformly distributed, the implementation of the perturbation is significantly simplified to the random choice of integer λ in Formula (2.4). This functionality is made available by most, if not all, higher programming languages. Apart from that, we generate floating-point numbers with the largest possible number of trailing zeros. This possibly reduces the rounding error in practice. ○

Issue 5: The projection of $\overline{\mathcal{U}}_\delta(\bar{y})|_{\mathbb{G}}$ is non-uniform.

The *original perturbation area* $\overline{\mathcal{U}}_\delta(\bar{y})|_{\mathbb{G}}$ is a discrete set of uniformly distributed points, of which every point is chosen with the same probability. As a consequence, the *predicate perturbation area* $\bar{U}_\delta(\bar{x})|_{\mathbb{G}}$ is also uniformly distributed. This does not imply that all points in the projected grid appear with the same probability! We illustrate, explain, and solve this issue in Section 2.12. For now, we continue our consideration under the assumption that all points in $\bar{U}_\delta(\bar{x})|_{\mathbb{G}}$ are uniformly distributed and randomly chosen with the same probability.

Issue 6: Analyses for various perturbation areas may differ.

In the determination of $\mathrm{pr}(f|_{\mathbb{G}})$ in Formula (2.5), we encounter the difficulty of finding the minimum ratio between the guarded and all possible inputs *for all possible perturbation areas*, that is, for all $\bar{x} \in A$. We can address this problem with a simple worst-case consideration if we cannot, or do not wish to, gain further insight into the behavior of f: We just expect that whatever could negatively affect the analysis of f within the total predicate perturbation area $\bar{U}_\delta(A)$ affects the perturbation area $\bar{U}_\delta(\bar{x})$ under consideration. This way, we safely obtain a lower bound on the minimum.

Issue 7: There is no general relation between $\mathrm{pr}(f|_{\mathbb{G}})$ and $\mathrm{pr}(f)$.

Example 2.2. We continue Example 2.1. In addition, let $R_3 = [\frac{1}{10}, \frac{9}{10}]$ be an interval. Because $U \subseteq [-2^1, 2^1]$, we have $e_{\max} = 1$, and $\tau = 2^{e_{\max}-L-1} = \frac{1}{4}$. The situation is depicted in Figure 2.2.

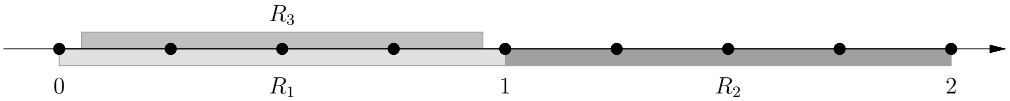


Figure 2.2: The distribution of the grid points $\mathbb{G}_{2,3,1}$ within the interval $[0, 2]$.

We again compare the continuous and the discrete case: What is the probability that a randomly chosen point $x \in U$ lies inside of R_1 (R_2 or R_3, respectively)? The probability is now higher for R_1 and R_2 in the discrete case

$$\mathrm{pr}(R_1) = \mathrm{pr}(R_2) = \frac{1}{2} \quad < \quad \mathrm{pr}(R_1|_{\mathbb{G}}) = \mathrm{pr}(R_2|_{\mathbb{G}}) = \frac{5}{9},$$

and higher for R_3

$$\mathrm{pr}(R_3) = \frac{2}{5} \quad > \quad \mathrm{pr}(R_3|_{\mathbb{G}}) = \frac{1}{3}$$

in the real case. ○

We observe that the restriction to points in $\mathbb{G}$ does not entirely solve the initial problem: We still cannot relate the probability $\mathrm{pr}(f)$ with $\mathrm{pr}(f|_{\mathbb{G}})$ in general. To improve the estimate, another modification is necessary which we indicate in Example 2.3: *If we make the interval slightly larger, we can safely determine the inequality.*

Example 2.3. Let τ be the grid unit of $\mathbb{G}$. We define three intervals $R \subset R_{\text{aug}} \subset U$. Let $U \subset \mathbb{R}$ be a closed interval of length $\lambda_0\tau$ with $\lambda_0 \in \mathbb{N}$. Let $R_{\text{aug}} \subset U$ be an interval of length at least τ that has the limits $R_{\text{aug}} := [a - \frac{\tau}{2}, b + \frac{\tau}{2}]$ for $a, b \in \mathbb{R}$. Finally, we define $R := [a, b]$. In addition, let $\lambda \in \mathbb{N}$ be such that

$$\lambda\tau \quad \leq \quad \mu(R_{\text{aug}}) \quad < \quad (\lambda + 1)\tau.$$

We observe that the number of grid points in $R|_{\mathbb{G}}$ and $R_{\text{aug}}|_{\mathbb{G}}$ is bounded by

$$\lambda - 1 \quad \leq \quad |\, R|_{\mathbb{G}} \,| \quad \leq \quad \lambda \quad \leq \quad |\, R_{\text{aug}}|_{\mathbb{G}} \,| \quad \leq \quad \lambda + 1.$$

Moreover, we see that

$$\frac{|\,R|_{\mathbb{G}}\,|}{|\,U|_{\mathbb{G}}\,|} \;\le\; \frac{\lambda}{\lambda_0+1} \;\le\; \frac{\lambda}{\lambda_0} \;=\; \frac{\lambda\tau}{\lambda_0\tau} \;\le\; \frac{\mu(R_{\mathrm{aug}})}{\mu(U)}.$$

So, it is more likely that a random point in U lies inside of R_{aug} than that a random point in $U|_{\mathbb{G}}$ lies inside of $R|_{\mathbb{G}}$. The inequality

$$\mathrm{pr}(R|_{\mathbb{G}}) \;\le\; \mathrm{pr}(R_{\mathrm{aug}})$$

is valid independently of the actual choice or location of R. ○

Issue 8: There is still no general relation between $\mathrm{pr}(f|_{\mathbb{G}})$ and $\mathrm{pr}(f)$.

The probability $\mathrm{pr}(f)$ is defined as the ratio of volumes. The definition is, in particular, independent of the location and shape of the involved sets. As an example, we consider the three different (shaded) regions in Figure 2.3 that all have the same volume.

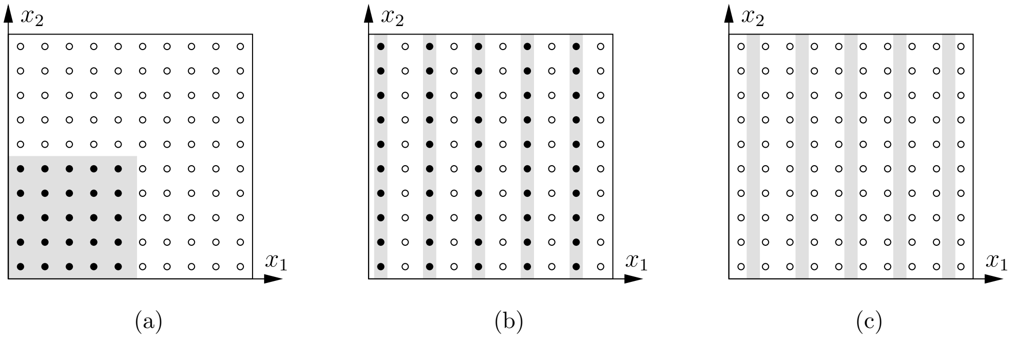


Figure 2.3: The volume of the shaded region R is the same in all three pictures. Depending on the shape and location of R, it covers various fractions of the discrete set $\mathbb{G}$. For example: (a) a quarter, (b) a half, (c) nothing.

We observe that the shape and the location matter if we derive the induced ratio for points in $\mathbb{G}$. The discrepancy between the ratios is caused by the implicit assumption that the grid unit τ is sufficiently small. (Asymptotically, the ratios approach the same limit in the three illustrated examples for $\tau \to 0$.) Making this assumption explicitly leads to a second constraint on the precision L, which we call the *grid unit condition.* To solve this issue, we need a way to adjust the grid unit τ to the shape of R. We solve this problem in Section 2.2.4.

Summary and validation of $\mathrm{pr}(f|_{\mathbb{G}})$

We summarize our considerations so far. The analysis of a guarded algorithm must reflect its actual behavior. Therefore, we have defined the success probability of a floating-point evaluation of f in Formula (2.5) such that it is based on the behavior of guards. Furthermore, we have studied the interrelationship between the success probability for floating-point and real arithmetic to prepare the analysis in real space. Keep in mind that we have introduced a specialized perturbation on a regular grid $\mathbb{G}$ (in practice and in analysis), which is necessary for the derivation of the interrelationship. Moreover, we now make this relationship explicit for a single interval. (The general relationship is formulated in Section 2.2.4.)

Example 2.4. (Continuation of Example 2.3). Let $f : U \to \mathbb{R}$. We assume the following property of R: If $x \in U|_{\mathbb{G}}$ lies outside of R, then the guard $\mathcal{G}(x)$ is true. Then we have

$$\begin{aligned} \mathrm{pr}(f|_{\mathbb{G}}) &= \frac{|\,\{x \in U|_{\mathbb{G}} \,:\, \mathcal{G}(x) \text{ is true}\}\,|}{|\,U|_{\mathbb{G}}\,|} \\ &\geq 1 - \frac{|\,R|_{\mathbb{G}}\,|}{|\,U|_{\mathbb{G}}\,|} \\ &\geq 1 - \frac{\mu(R_{\mathrm{aug}})}{\mu(U)}. \end{aligned}$$

We conclude: *If we prove by means of abstract mathematics that*

$$1 - \frac{\mu(R_{\mathrm{aug}})}{\mu(U)} \quad \geq \quad p$$

for a probability $p \in (0,1)$*, we have implicitly proven that*

$$\mathrm{pr}(f|_{\mathbb{G}}) \quad \geq \quad p$$

for a randomly chosen grid point in $\mathbb{G}$. Be observe that $\mathrm{pr}(f|_{\mathbb{G}})$ is defined only by discrete quantities. ○

Warning: processing exceptional points

We explain in this paragraph why it is absolutely non-obvious how to process exceptional points in general. Assume that we want to exclude the set $D \subset A$ from the analysis. This changes our success probability from Formula (2.5) into

$$\begin{aligned} \mathrm{pr}(f|_{\mathbb{G}}) &= \min_{\bar{x} \in A} \frac{\left|\,\left\{x \in \bar{U}_{\delta}(\bar{x})|_{\mathbb{G}} \,:\, \mathcal{G}(x) \text{ is true}\right\} \setminus D\,\right|}{\left|\,\bar{U}_{\delta}(\bar{x})|_{\mathbb{G}}\,\right|} \\ &\geq \min_{\bar{x} \in A} \frac{\max\left\{0,\ \left|\left\{x \in \bar{U}_{\delta}(\bar{x})|_{\mathbb{G}} \,:\, \mathcal{G}(x) \text{ is true}\right\}\right| - |D|\right\}}{\left|\,\bar{U}_{\delta}(\bar{x})|_{\mathbb{G}}\,\right|}. \end{aligned}$$

To obtain a practicable solution, it is reasonable to assume that D is finite and, moreover, that $|D| \ll \left|\,\bar{U}_{\delta}(\bar{x})|_{\mathbb{G}}\,\right|$. This changes the relation in Example 2.4 into:

$$\mathrm{pr}(f|_{\mathbb{G}}) \quad \geq \quad \max\left\{0,\ 1 - \frac{\mu(R_{\mathrm{aug}})}{\mu(U)} - \frac{|D|}{\left|\,\bar{U}_{\delta}(\bar{x})|_{\mathbb{G}}\,\right|}\right\}.$$

This estimate still contains two quantities that depend on the floating-point arithmetic. But our plan was to eliminate this dependency. In spite of the simplifying assumptions, it is non-obvious how to perform the analysis in real space in general. *Our suggested solution to this problem is to avoid exceptional points. Alternatively, we declare them critical (see next section), which triggers an exclusion of their environment.*

2.2.3 Fp-safety Bound, Critical Set, Region of Uncertainty

The fp-safety bound

The sign of a floating-point evaluation is verified by a guard $\mathcal{G}$. The essential part of its realization is the fp-safety bound. We show in Section 2.9 that there are fp-safety bounds for a wide class of functions.

Definition 2.6 (lower fp-safety bound)**.** Let $(f, k, A, \delta, e_{\max})$ be a predicate description. Let $S_{\inf f} : \mathbb{N} \to \mathbb{R}_{\geq 0}$ be a monotonically decreasing function that maps a precision L to a non-negative value. We call $S_{\inf f}$ a *(lower) fp-safety bound for f on A* if the statement

$$|f(x)| > S_{\inf f}(L) \qquad \Rightarrow \qquad \mathrm{sign}(f(x)|_{\mathbb{F}_L}) = \mathrm{sign}(f(x)) \tag{2.6}$$

is true for every precision $L \in \mathbb{N}$ and for all $x \in \bar{U}_\delta(A)|_{\mathbb{F}_L}$.

For the time being, we consider K to be a constant. We abandon this assumption in Section 2.10 where we introduce *upper* fp-safety bounds. Until then, we only consider *lower* fp-safety bounds.

The critical set

We next introduce a classification of the points in $\bar{U}_\delta(A)$ in dependence on their neighborhood. (We refine the definition on Page 85.)

Definition 2.7 (critical)**.** Let $(f, k, A, \delta, e_{\max})$ be a predicate description. We call a point $c \in \bar{U}_\delta(\bar{x})$ *critical* if

$$\inf_{x \in U_\varepsilon(c) \setminus \{c\}} |f(x)| \quad = \quad 0 \tag{2.7}$$

on a neighborhood $U_\varepsilon(c)$ for infinitesimal small $\varepsilon > 0$. Furthermore, we call zeros of f that are not critical *less-critical.* Points that are neither critical nor less-critical are called *non-critical.* We define the *critical set $C_{f,\delta}$ of f at $\bar{x} \in A$ with respect to* δ as the union of critical and less-critical points within $\bar{U}_\delta(\bar{x})$.

In other words, we call c critical if there is a Cauchy sequence[7] $(a_i)_{i \in \mathbb{N}}$ in $\bar{U}_\delta(\bar{x}) \setminus \{c\}$ where $\lim_{i\to\infty} a_i = c$ and $\lim_{i\to\infty} f(a_i) = 0$. We remember that the metric space[8] $\mathbb{R}^k$ is complete, that is, the limit of the sequence (a_i) lies inside of the closure $\bar{U}_\delta(\bar{x})$. We sometimes omit the indices of the critical set C if they are given by the context.

[7] Analysis: The Cauchy sequence is defined in Forster [30].

[8] Topology: Metric space and completeness are defined in Jänich [46].

Example 2.5. We consider the three functions that are depicted in Figure (2.4). Let $f_1(x) = x^2$. Let $f_2(x) = x^2$ for $x \neq 0$, and $f_2(0) = 2$. Let $f_3(x) = x^2 + 1$ for $x \notin \{-2, 1\}$, and $f_3(-2) = 0$, and $f_3(1) = 0.2$.

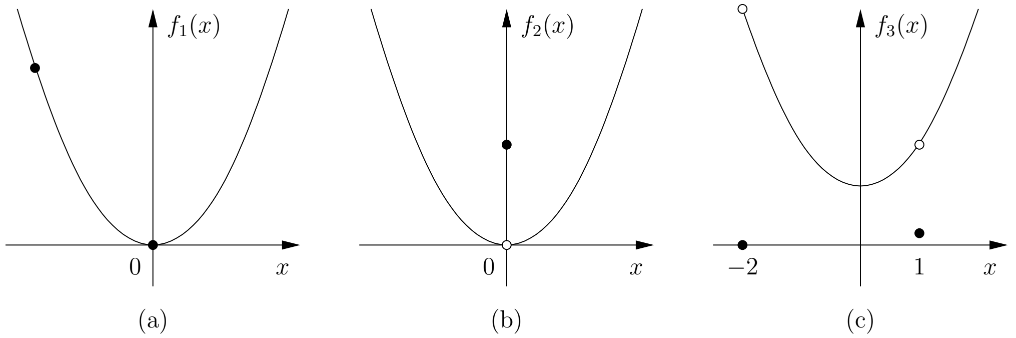


Figure 2.4: Examples of critical, less-critical and non-critical points.

The point $x = 0$ in Picture (a) is a zero and a critical point for f_1. In (a), every argument $x \neq 0$ is non-critical for f_1. In (b), f_2 is non-zero at $x = 0$, but $x = 0$ is a critical point for f_2. In (c), the argument $x = -2$ is less-critical for f_3 and the argument $x = 1$ is non-critical for f_3. ○

What is the difference between critical and less-critical points? We observe that the point c is excluded from its neighborhood in Formula (2.7). Zeros of f would trivially be critical otherwise. Furthermore, we observe that zeros of continuous functions are always critical. For our purpose, it is important to see that the infimum of $|f|$ is positive if we exclude the less-critical points *itself* and *neighborhoods* of critical points. We could technically treat both kinds differently in the analysis and still ensure that the result of the analysis is valid for floating-point arithmetic. Only for simplicity do we deal with them in the same way by adding these points to the critical set. Only for simplicity do we also add exceptional points to the critical set.

The region of uncertainty

The next construction is a certain environment of the critical set.

Definition 2.8 (region of uncertainty)**.** Let $(f, k, A, \delta, e_{\max})$ be a predicate description. In addition, let $\gamma \in \mathbb{R}^k_{>0}$. We call

$$R_{f,\gamma}(\bar{x}) \quad := \quad \bar{U}_\delta(\bar{x}) \cap \left(\bigcup_{c \in C_{f,\delta}(\bar{x})} U_\gamma(c) \right) \qquad (2.8)$$

the *region of uncertainty for f induced by γ with respect to $\bar{x}$.*

In our presentation we use the axis-parallel boxes $U_\gamma(c)$ to define the specific γ-neighborhood of C; other shapes require adjustments, see Section 2.2.4. The sets

$U_\gamma(c)$ are open, and the complement of $R_{f,\gamma}(\bar{x})$ in $\bar{U}_\delta(\bar{x})$ is closed. We omit the indices of the region of uncertainty R if they are given by the context.

The vector γ defines the tuple of componentwise distances to c. The presentation requires a formal definition of the set of all admissible γ. This set is either a box or a line. Let $\hat{\gamma} \in \mathbb{R}^k_{>0}$. Then we define the unique open axis-parallel box with vertices 0 and $\hat{\gamma}$ as

$$\Gamma\text{-box}_{\hat{\gamma}} \quad := \quad \left\{\gamma' = (\gamma'_1, \ldots, \gamma'_k) : \gamma'_i \in (0, \hat{\gamma}_i) \text{ for all } i \in I\right\}$$

and the open diagonal from 0 to $\hat{\gamma}$ inside of $\Gamma\text{-box}_{\hat{\gamma}}$ as

$$\Gamma\text{-line}_{\hat{\gamma}} \quad := \quad \{\gamma : \gamma = \lambda\hat{\gamma} \text{ with } \lambda \in (0,1)\}\,.$$

It is important that the γ_i can be chosen arbitrarily small, whereas the upper bounds $\hat{\gamma}_i$ are only introduced for technical reasons; we assume that $\hat{\gamma}$ is "sufficiently" small.[9] Occasionally, we omit $\hat{\gamma}$.

We have already seen that there is need to augment the region of uncertainty (see Issue 7 and 8 in Section 2.2.2). This is accomplished by the mapping $\gamma \mapsto \text{aug}(\gamma) := \frac{\gamma}{t}$ for $t \in (0,1)$. Later on we use the fact that $\gamma \in \Gamma\text{-box}_{\hat{\gamma}}$ if $\text{aug}(\gamma) \in \Gamma\text{-box}_{\hat{\gamma}}$, and $\gamma \in \Gamma\text{-line}_{\hat{\gamma}}$ if $\text{aug}(\gamma) \in \Gamma\text{-line}_{\hat{\gamma}}$. We call $R_{f,\text{aug}(\gamma)}$ the *augmented region of uncertainty for f under* $\text{aug}(\gamma)$. We denote by Γ the *set of valid augmented* γ and include it in the predicate description.

Definition 2.9. We extend Definition 2.4 and call $(f, k, A, \delta, e_{\max}, \Gamma)$ a *predicate description* if: 7. $\Gamma = \Gamma\text{-line}_{\hat{\gamma}}$ or $\Gamma = \Gamma\text{-box}_{\hat{\gamma}}$ for a sufficiently small $\hat{\gamma} \in \mathbb{R}^k_{>0}$.

2.2.4 The Grid Unit Condition

We must base the analysis on a reliable relation between the success probability for floating-point arithmetic and the one for real arithmetic. We can ensure such a relation with an additional constraint on the precision. We stress that we do not consider function values here; we consider the size of the subset of floating-point numbers which are covered by the region of uncertainty. *This is the first time the precision of the floating-point arithmetic has been adjusted to the shape of the region of uncertainty.*

We adjust the distance of grid points (i.e., the grid unit τ) to the "width" of the region of uncertainty γ. As we have seen in Issue 8 in Section 2.2.2, the grid unit τ must be sufficiently small (i.e., L must be sufficiently large) to derive a reliable probability $\text{pr}(f|_{\mathbb{G}})$ from $\text{pr}(f)$. The problem is illustrated in Figure 2.3 on page 25. We call this additional constraint on L the *grid unit condition*

$$L \quad \geq \quad L_{\text{grid}} \tag{2.9}$$

for a certain $L_{\text{grid}} \in \mathbb{N}$ and expect that this constraint guarantees $\tau \ll \gamma$. We derive the threshold L_{grid} in the following. We refine the concept of the augmented region

[9] This information can be ignored in first reading. More information and the formal bound is given in Remark 2.2.2 on Page 38.

of uncertainty which we mentioned briefly in Section 2.2.2. The discussion of Issue 7 suggests an additive augmentation $\gamma = \operatorname{aug}(\gamma')$ that fulfills

$$\tau_0 \overset{(I)}{\leq} \gamma_i' \overset{(II)}{\leq} \gamma_i - \tau_0$$

for all $1 \leq i \leq k$, where τ_0 is an upper bound on the grid unit. However, in the analysis, it is easier to handle a multiplicative augmentation

$$\gamma \overset{(III)}{:=} \frac{\gamma'}{t}$$

for a factor $t \in (0,1)$, so that we define $\operatorname{aug}(\gamma') := \frac{\gamma'}{t}$. We call $\frac{1}{t}$ the *augmentation factor* for the region of uncertainty. Together, this leads to the implications

$$\begin{array}{rcrcl} (I) \text{ and } (III) & \Rightarrow & \tau_0 & \leq & t \cdot \min\limits_{1\leq i\leq k} \gamma_i, \\ (II) \text{ and } (III) & \Rightarrow & \tau_0 & \leq & (1-t) \cdot \min\limits_{1\leq i\leq k} \gamma_i, \\ \text{and consequently,} & \Rightarrow & \tau_0 & \overset{(IV)}{\leq} & \min\{t, 1-t\} \cdot \min\limits_{1\leq i\leq k} \gamma_i. \end{array}$$

Furthermore, we demand that τ_0 is a power of 2 which turns (IV) into the equality

$$\tau_0 \overset{(V)}{=} 2^{\left\lfloor \log_2\left(\min\{t,1-t\}\cdot \min_{1\leq i\leq k} \gamma_i\right)\right\rfloor}.$$

Due to Formula (2.3) in Definition 2.5, we also know that

$$\tau_0 \overset{(VI)}{=} 2^{e_{\max} - L_{\text{grid}} - 1}.$$

Therefore, we can deduce L_{grid} from (V) and (VI) as

$$L_{\text{grid}}(\gamma) := \left\lceil e_{\max} - 1 - \log_2\left(\min\{t, 1-t\} \cdot \min_{1\leq i\leq k} \gamma_i\right)\right\rceil. \tag{2.10}$$

As an example, we obtain $L_{\text{grid}}(\gamma) = \lceil e_{\max} - \log_2 \min_{1\leq i\leq k} \gamma_i \rceil$ for $t = \frac{1}{2}$. Without loss of generality, we restrict the choice of the parameter t to values of at least $\frac{1}{2}$ from now on. This way, we get rid of the min expression in the Formulas. We refine the notion of a predicate description.

Definition 2.10. We extend Definition 2.9 and call $(f, k, A, \delta, e_{\max}, \Gamma, t)$ a *predicate description* if: 8. $t \in \left[\frac{1}{2}, 1\right)$.

We are now able to summarize the construction above.

Theorem 2.1. *Let $(f, k, A, \delta, e_{\max}, \Gamma, t)$ be a predicate description. Then*

$$\frac{\mu\left(R_\gamma(\bar{x})\right)}{\mu\left(U_\delta(\bar{x})\right)} \geq \frac{|\, R_{t\gamma}(\bar{x})|_{\mathbb{G}_L}\,|}{|\, U_\delta(\bar{x})|_{\mathbb{G}_L}\,|} \tag{2.11}$$

for all precisions $L \geq L_{\text{grid}}(\gamma)$, where L_{grid} is defined in Formula (2.10).

We add some remarks on the grid unit condition. First, Unequation (2.11) guarantees that the success probability for grid points is at least the success probability for real arithmetic. This justifies the analysis in real space.

Second, the grid unit condition is a fundamental constraint: It does not depend on the function that realizes the predicate, the dimension of the (projected or full) perturbation area, the perturbation parameter, or the critical set. The threshold L_{grid} mainly depends on the augmentation factor $\frac{1}{t}$ and γ. In particular, we observe that an additional bit of the precision is sufficient to fulfill the grid unit condition for $\frac{\gamma}{2}$, i.e.,

$$L_{\text{grid}}\left(\frac{\gamma}{2}\right) \quad = \quad L_{\text{grid}}(\gamma) + 1.$$

Third, we have defined the region of uncertainty R_f by means of axis-parallel boxes $U_\gamma(c)$ for $c \in C_f$ in Definition 2.8. If R_f is defined in a different way, we must appropriately adjust the derivation of L_{grid} in this section.

Finally, we observe that the grid unit condition solves Issue 8 from Section 2.2.2. We reconsider the example in Figure 2.3 on page 25 and observe that the grid unit in Picture (a) fulfills the grid unit condition, whereas the condition fails in Pictures (b) and (c). Obviously, $\tau \gg \gamma$ in the latter cases.

2.2.5 Overview: Classification of the Input

In practice and in the analysis, we deal with real-valued functions whose signs decide predicates. The arguments of these functions belong to the perturbation area. In this section we give an overview of the various characteristics for function arguments that we have introduced so far. We strictly distinguish between terms of practice and terms of the analysis.

The diagram of the practice-oriented terms is shown in Figure 2.5. We consider the discrete perturbation area $U_\delta|_{\mathbb{G}}$. Controlled perturbation algorithms $\mathcal{A}_{\text{CP}}$ are designed with the intent to avoid the implementation of degenerate cases and to compute the combinatorial correct solution. Therefore, the guards in the embedded algorithm $\mathcal{A}_{\text{G}}$ must fail for the zero set and for arguments whose evaluations lead to wrong signs. The guard is designed such that the evaluation is definitely fp-safe if the guard does not fail (light-shaded region). Unfortunately, there is no convenient way to count (or bound) the number of arguments in $U_\delta|_{\mathbb{G}}$ for which the guard fails. For this reason we perform the analysis with real arithmetic and introduce further terms.

The diagram of the analysis-oriented terms is shown in Figure 2.6. We consider the real perturbation area U_δ. Instead of the zero set, we consider the critical set (see Definition 2.7). The critical set is a superset of the zero set. Then we choose the region of uncertainty as a neighborhood of the critical set (see Definition 2.8). We augment the region of uncertainty to obtain a result that is also valid for floating-point evaluations. We intend to prove fp-safety outside of the augmented region of uncertainty (i.e., on the light-shaded region). Therefore, we design a fp-safety bound

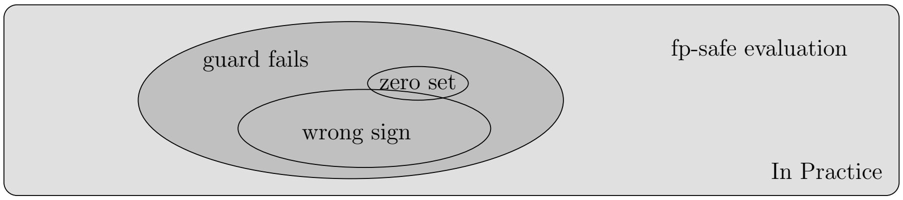


Figure 2.5: The diagram of the practice-oriented terms.

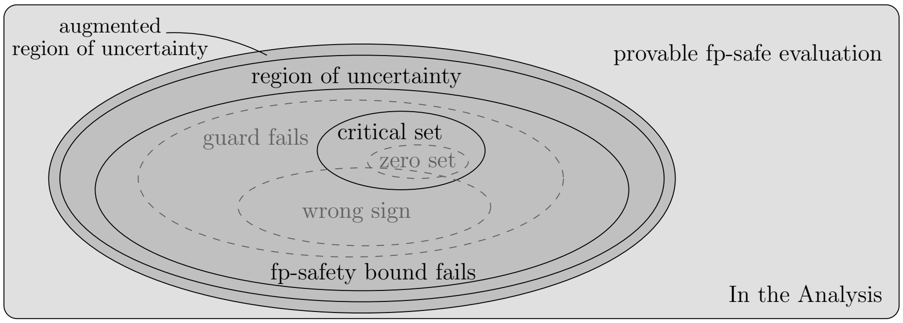


Figure 2.6: The diagram of the analysis-oriented terms (shown in black).

that is true outside of the region. This way, we can guarantee that the evaluation of a guard (in practice) only fails on a subset of the augmented region (in the analysis).

2.2.6 Applicability and Verifiability of Functions

We study the circumstances under which we may *apply* controlled perturbation to a predicate in practice and under which we can actually *verify* its application in theory. We stress that we are talking about a *qualitative* analysis here; the desired *quantitative* analysis is derived in the following sections.

Actually, *verifiability* is not necessary for the presentation of the analysis tool box. However, the distinction between applicability, verifiability, and analyzability was important during the development of the topic for this dissertation. We keep it in the presentation because it may also be helpful to the reader. Anyway, it is possible to skip this section, and even assuming equality between verifiability and analyzability will do no harm.

In practice

We specify the function property that the probability of a successful evaluation of f gets arbitrarily close to the certain event by increasing the precision.

Definition 2.11 (applicable)**.** Let $(f, k, A, \delta, e_{\max})$ be a predicate description. We call f *applicable* if for every $p \in (0, 1)$ there is $L_p \in \mathbb{N}$ such that the guarded evaluation of f is successful at a randomly perturbed input $x \in \bar{U}_\delta(\bar{x})|_{\mathbb{G}_L}$ with probability at least p for every precision $L \in \mathbb{N}$ with $L \geq L_p$ and every $\bar{x} \in A$.

Applicable functions can safely be used in guarded algorithms: Since the precision L is increased (without limit) after a predicate has failed, the success probability gets arbitrarily close to 1 for each predicate evaluation. As a consequence, the success probability of $\mathcal{A}_{\mathrm{G}}$ gets arbitrarily close to 1, too.

In the qualitative analysis

Unfortunately, we cannot check directly if f is applicable. Therefore, we introduce two properties that imply applicability.

Definition 2.12. Let $(f, k, A, \delta, e_{\max}, \Gamma\text{-line}, t)$ be a predicate description.

- (region-condition). For every $p \in (0,1)$, there is $\gamma \in \mathbb{R}^k_{>0}$ such that the geometric failure probability is bounded in the way
$$\frac{\mu(R_\gamma(\bar{x}))}{\mu(U_\delta(\bar{x}))} \quad \leq \quad (1-p) \tag{2.12}$$
for all $\bar{x} \in A$. We call this condition the *region-condition.*
- (safety-condition). There is a fp-safety bound $S_{\inf f} : \mathbb{N} \to \mathbb{R}_{>0}$ on $\bar{U}_\delta(A)$ with[10]
$$\lim_{L\to\infty} S_{\inf f}(L) \quad = \quad 0. \tag{2.13}$$
We call this condition the *safety-condition.*
- (verifiable). We call f *verifiable on* $\bar{U}_\delta(A)$ *for controlled perturbation* if f fulfills the region-condition and the safety-condition.

The region-condition guarantees the adjustability of the volume of the region of uncertainty. Note that the region-condition is actually a condition on the critical set. It states that the critical set is sufficiently "sparse".

The safety-condition guarantees the adjustability of the fp-safety bound. It states that for every $\varphi > 0$, there is a precision $L_{\text{safe}} \in \mathbb{N}$ with the property that
$$S_{\inf f}(L) \leq \varphi \tag{2.14}$$
for all $L \in \mathbb{N}$ with $L \geq L_{\text{safe}}$. We give an example of a verifiable function.

Example 2.6. Let $A \subset \mathbb{R}$ be an interval, let $\delta \in \mathbb{R}_{>0}$ and let $f : \bar{U}_\delta(A) \to \mathbb{R}$ be a univariate polynomial[11] of degree d with real coefficients, i.e.,
$$f(x) \quad = \quad a_d \cdot x^d + a_{d-1} \cdot x^{d-1} + \ldots + a_1 \cdot x + a_0.$$

[10] Technically, the assumption $S_{\inf f}(L) > 0$ is no restriction.

[11] We avoid the usual notation $f \in \mathbb{R}[x]$ to emphasize that the domain of f *must be bounded.*

We show that f is verifiable.
Part 1 (region-condition). Because of the fundamental theorem of algebra (see, e.g., Lamprecht [53]), f has at most d real roots. Therefore, the size of the critical set C_f is bounded by d, and the volume of the region of uncertainty $R_\gamma(\bar{x})$ is upper-bounded by $2d\gamma$. For a given $p \in (0,1)$, we then choose

$$\gamma \quad := \quad \frac{(1-p)\delta}{d}$$

which fulfills the region-condition because of

$$\frac{\mu(R_\gamma(\bar{x}))}{\mu(U_\delta(\bar{x}))} \quad \leq \quad \frac{2\gamma d}{2\delta} \quad = \quad 1-p.$$

Part 2 (safety-condition). Corollary 2.18 on page 82 provides the fp-safety bound

$$S_{\inf f}(L) \quad := \quad (d+2) \max_{1 \leq i \leq d} |a_i| \;\; 2^{(d+1)e_{\max}+1-L}$$

for univariate polynomials. Since $S_{\inf f}(L)$ converges to zero as L approaches infinity, the safety-condition is fulfilled. Therefore, f is verifiable. ○

We show that if a function is verifiable, it has a positive lower bound on its absolute value outside of its region of uncertainty.

Lemma 2.2. *Let $(f, k, A, \delta, e_{\max}, \Gamma\text{-line}, t)$ be a predicate description, and let f be verifiable. Then, for every $\gamma \in \mathbb{R}^k_{>0}$, there is $\varphi \in \mathbb{R}_{>0}$ with*

$$\varphi \leq |f(x)| \tag{2.15}$$

for all $x \in \bar{U}_\delta(\bar{x}) \setminus R_\gamma(\bar{x})$ and for all $\bar{x} \in A$.

Proof. We assume the opposite. That means, in particular, for every $i \in \mathbb{N}$, there is $a_i \in \bar{U}_\delta(\bar{x}) \setminus R_\gamma(\bar{x})$ such that $|f(a_i)| < \frac{1}{i}$. Then $(a_i)_{i \in \mathbb{N}}$ is a bounded sequence with accumulation points in $\bar{U}_\delta(\bar{x}) \setminus R_\gamma(\bar{x})$. These points must be critical and hence belong to $R_\gamma(\bar{x})$. This is a contradiction. □

Finally, prove that verifiability of functions implies applicability.

Lemma 2.3. *Let $(f, k, A, \delta, e_{\max}, \Gamma\text{-line}, t)$ be a predicate description, and let f be verifiable. Then f is applicable.*

Proof. Let $p \in (0,1)$. Then the geometric success probability is bounded by p. Therefore, there must be an upper bound on the volume of the region of uncertainty (see Definition 2.12). In addition, there is a precision L_{grid} such that we may interpret this region as an augmented region $R_{\text{aug}(\gamma)}$ (see Theorem 2.1). Furthermore, there must be a positive lower bound on $|f|$ outside of R_γ (see Lemma 2.2). Moreover, there must be a precision L_{safe} for which the fp-safety bound is smaller than the bound on $|f|$. This implies that the guarded evaluation of f is successful at a randomly perturbed input with probability at least p for every precision $L \geq \max\{L_{\text{safe}}, L_{\text{grid}}\}$. So, f is applicable (see Definition 2.11). □

2.3 General Analysis Tool Box

The general analysis tool box to analyze controlled perturbation algorithms is presented in the remainder of this chapter. We call the presentation a tool box because its components are strictly separated from each other and sometimes allow alternative derivations. In particular, we present three ways to analyze functions. Here, we briefly introduce the tool box and refer to the detailed presentation of its components in the subsequent sections. *The decomposition of the analysis into well-separated components and their precise description is an innovation of this presentation.*

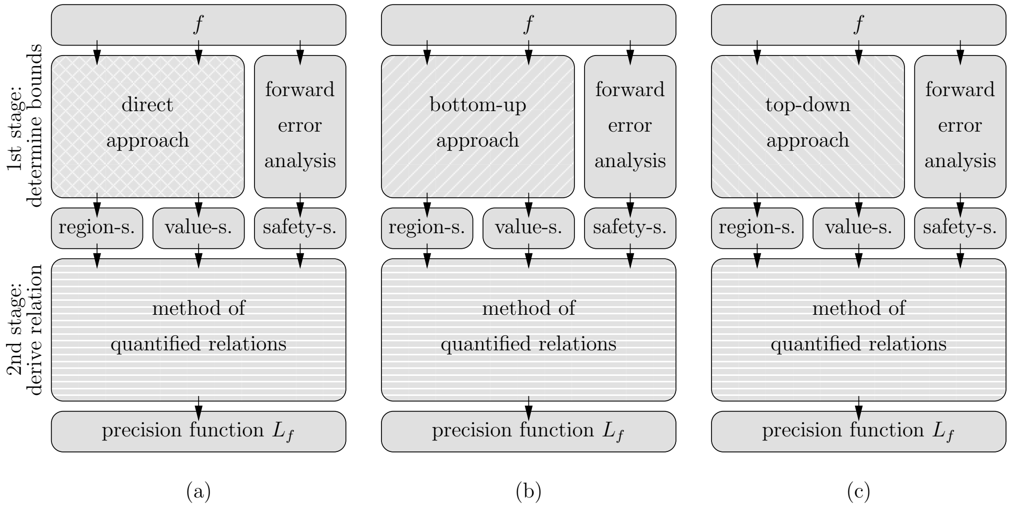


Figure 2.7: Illustration of the various ways to analyze functions.

The tool box is subdivided into components. We begin by explaining the *analysis of functions*. The diagram in Figure 2.7 illustrates three ways to analyze functions. We subdivide the function analysis in two stages. The analysis itself in the second stage requires three necessary bounds, also known as the *interface*, which are defined in Section 2.4: *region-suitability*, *value-suitability* and *safety-suitability*. In Section 2.5, we introduce the *method of quantified relations*, which represents the actual analysis in the second stage. In the first stage, we pay special attention to the derivation of two bounds of the interface and suggest three different ways to solve the task. We show in Section 2.6 how the bounds can be derived in a *direct approach* from geometric measures. Furthermore, we show how to build up the bounds for the desired function from simpler functions in a *bottom-up approach* in Section 2.7. Moreover, we present a derivation of the bounds by means of a "sequence of bounds" in a *top-down approach* in Section 2.8. Finally, we show how we can derive the third necessary bound of the interface with an *error analysis* in Section 2.9.

We deal with the *analysis of algorithms* in Section 2.12. The idea is illustrated in Figure 2.25 on page 89. We again subdivide the analysis into two stages. The actual analysis of algorithms is the *method of distributed probability*, which represents the second stage and is explained in Section 2.12.3. The *interface* between the stages is subdivided in two groups. (1) There are algorithm prerequisites (to the left of the dashed line in the figure). These bounds are defined and derived in Section 2.12.1: *evaluation-suitability*, *predicate-suitability* and *perturbation-suitability.* (2) There are predicate prerequisites (to the right of the dashed line in the figure). These are determined by means of function analyses.

2.4 Necessary Conditions for the Analysis of Functions

The method of quantified relations, which is introduced in the next section, actually performs the analysis of real-valued functions. We prepare its applicability below. In Section 2.4.1, we present three necessary conditions: the *region-, value- and safety-suitability.* Together, these conditions are also sufficient to apply the method. Because these conditions are deduced in the first stage of the function analysis (see Section 2.6–2.9) and are used in the second stage (see Section 2.5), we also refer to them as the interface between the two stages (see Figure 2.8). *This is the first time that we precisely define the prerequisites of the function analysis.* The definitions are followed by an example. In Section 2.4.2, we summarize all function properties.

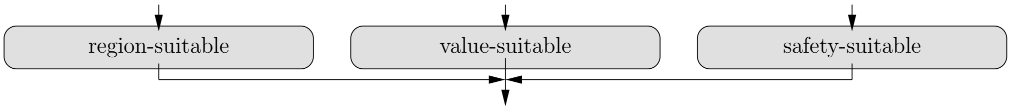


Figure 2.8: The interface between the two stages of the analysis of functions.

2.4.1 Analyzability of Functions

We define and explain the three function properties that are necessary for the analysis. Their associated bounding functions constitute the interface between the two stages. Informally, the properties have the following meanings:

- We can reduce the volume of the region of uncertainty to any arbitrarily small value (region-suitability).
- There are positive and finite limits on the absolute value of f outside of the region of uncertainty (value-suitability).
- We can reduce the rounding error in the floating-point evaluation of f to any arbitrarily small value (safety-suitability).

The region-suitability

The region-suitability is a geometric condition on the neighborhood of the critical set. We demand that we can adjust the volume of the region of uncertainty to any arbitrarily small value. For technical reasons we need an invertible bound.

Definition 2.13 (region-suitable)**.** Let $(f, k, A, \delta, e_{\max}, \Gamma\text{-line}, t)$ be a predicate description. We call f *region-suitable* if the critical set of f is either empty or if there is an invertible upper-bounding function[12]

$$\nu_f : \Gamma\text{-line} \to \mathbb{R}_{>0}$$

[12] Instead of ν_f we can also use its complement χ_f. See the following Remark 2.2.4 for details.

on the volume of the region of uncertainty with the property: For every $p \in (0,1)$, there is $\gamma \in \Gamma$-line such that

$$\frac{\mu(R_\gamma(\bar{x}))}{\mu(U_\delta(\bar{x}))} \;\leq\; \frac{\nu_f(\gamma)}{\mu(U_\delta(\bar{x}))} \;\leq\; (1-p) \tag{2.16}$$

for all $\bar{x} \in A$.

Remark 2.2. We add several remarks on the definition above.

1. Region-suitability is related to the region-condition in the following way: The criterion for region-suitability results from the replacement of $\mu(R_\gamma(\bar{x}))$ in Formula (2.12) with a function ν_f. This changes the region-condition in Definition 2.12 into a quantitative bound.

2. Of course, controlled perturbation cannot work if the region of uncertainty covers the entire perturbation area of $\bar{x}$. We have said that we consider $\gamma \in \Gamma\text{-line}_{\hat{\gamma}}$ for a "sufficiently" small $\hat{\gamma} \in \mathbb{R}^k_{>0}$. More formally, we postulate $\nu(\hat{\gamma}) \ll \mu(U_\delta(\bar{x}))$. To keep the notation as plain as possible, we are aware of this fact and do not make this condition explicit in our statements.

3. The invertibility of the bonding function ν_f is essential for the method of quantified relations as we see in the proof of Theorem 2.6. It is used there to deduce the parameter γ from the volume of the region of uncertainty—with the exception of an empty critical set that does not imply any restriction on γ.

4a. The function ν_f provides an upper bound on the volume of the region of uncertainty within the perturbation area of $\bar{x}$. It is sometimes more convenient to consider its complement

$$\chi_f(\gamma) \;:=\; \mu\left(U_\delta(\bar{x})\right) \,-\, \nu_f(\gamma). \tag{2.17}$$

The function $\chi_f(\gamma)$ provides a lower bound on the volume of the *region of provable fp-safe inputs.*

4b. The special case $\nu_f \equiv 0$ corresponds to the special case $\chi_f \equiv \mu\left(U_\delta(\bar{x})\right)$. Then the critical set is empty and there is no region of uncertainty. This implies that $\varphi_f(\gamma)$ can also be chosen as a constant function (see the value-suitability below).

4c. Based on Formula (2.17), we can demand the existence of an invertible function $\chi_f : \Gamma\text{-line} \to \mathbb{R}_{>0}$ instead of ν_f in the definition of region-suitability. That means, either $\chi_f \equiv \mu\left(U_\delta(\bar{x})\right)$ or $\chi_f : \Gamma\text{-line} \to \mathbb{R}_{>0}$ in an invertible function.

5. We make the following observations about region-suitability: (a) If the critical set is finite, f is region-suitable. (b) If the critical set contains an open set, f cannot be region-suitable. (c) If the critical set is a set of measure zero, it does not imply that f is region-suitable. Be aware that these properties are not equivalent: If f is region-suitable, the critical set is a set of measure zero. But a critical set of measure zero does not necessarily imply that f is region-suitable: In topology, we learn that $\mathbb{Q}$ is dense[13] in $\mathbb{R}$; hence, any open ε-neighborhood of $\mathbb{Q}$ equals $\mathbb{R}$. In set theory, we learn that[14] $|\mathbb{Q}| = \aleph_0 < 2^{\aleph_0} = |\mathbb{R}|$; hence, f cannot be region-suitable if the critical set is (locally) "too dense." ○

[13] Topology: "$\mathbb{Q}$ is dense in $\mathbb{R}$" means that $\overline{\mathbb{Q}} = \mathbb{R}$. For example, see Jänich [46, p. 63].

[14] Set Theory: Cardinalities of (infinite) sets are denoted by $\aleph_i$. For example, see Deiser [19, 162ff].

The inf-value-suitability

The inf-value-suitability is a condition on the behavior of the function f. We demand that there is a positive lower bound on the absolute value of f outside of the region of uncertainty.

Definition 2.14 (inf-value-suitable)**.** Let $(f, k, A, \delta, e_{\max}, \Gamma\text{-line}, t)$ be a predicate description. We call f *(inf-)value-suitable* if there is a lower-bounding function

$$\varphi_{\inf f} : \Gamma\text{-line} \to \mathbb{R}_{>0}$$

on the absolute value of f with the property: For every $\gamma \in \Gamma$-line, we have

$$\varphi_{\inf f}(\gamma) \quad \leq \quad |f(x)| \tag{2.18}$$

for all $x \in \bar{U}_\delta(\bar{x}) \setminus R_\gamma(\bar{x})$ and for all $\bar{x} \in A$.

We extend this definition by an upper bound on the absolute value of f in Section 2.10 and call this property sup-value-suitability; until then, we call the inf-value-suitability simply the value-suitability and also write φ_f instead of $\varphi_{\inf f}$. The criterion for value-suitability results from the replacement of the constant φ in Formula (2.15) with the bounding function φ_f. This changes the existence statement of Lemma 2.2 into a quantitative bound.

The inf-safety-suitability

The inf-safety-suitability is a condition on the error analysis of the floating-point evaluation of f. We demand that we can adjust the rounding error in the evaluation of f to any arbitrarily small value. For technical reasons, we demand an invertible bounding function.[15]

Definition 2.15 (inf-safety-suitable)**.** Let $(f, k, A, \delta, e_{\max})$ be a predicate description. We call f *(inf-)safety-suitable* if there is an injective fp-safety bound $S_{\inf f}(L) : \mathbb{N} \to \mathbb{R}_{>0}$ that fulfills the safety-condition in Formula (2.13) and if

$$S_{\inf f}^{-1} : (0, S_{\inf f}(1)] \to \mathbb{R}_{>0}$$

is a strictly monotonically decreasing real continuation of its inverse.

We extend the definition by sup-safety-suitability in Section 2.10; until then we call the inf-safety-suitability simply the safety-suitability.

[15] We leave the extension to non-invertible or discontinuous functions to the reader. We see in Section 2.9 that predicates of a wide class lead to continuous bounding functions.

The analyzability

Based on the definitions above, we next define analyzability, relate it to verifiability, and give an example for the definitions.

Definition 2.16 (analyzable)**.** We call f *analyzable* if it is region-, value- and safety-suitable.

Lemma 2.4. *Let f be analyzable. Then f is verifiable.*

Proof. If f is analyzable, f is especially region-suitable. Then the region-condition in Definition 2.12 is fulfilled because of the bounding function ν_f. In addition, f must also be safety-suitable. Then the safety-condition in Definition 2.12 is fulfilled because of the bounding function $S_{\inf f}$. Together, both conditions imply that f is verifiable. □

We support the definitions above with the example of univariate polynomials. Because we refer to this example later on, we formulate it as a lemma.

Lemma 2.5. *Let f be the univariate polynomial*

$$f(x) \quad = \quad a_d \cdot x^d + a_{d-1} \cdot x^{d-1} + \ldots + a_1 \cdot x + a_0 \tag{2.19}$$

of degree d and let $(f, k, A, \delta, e_{\max}, \Gamma\text{-line}, t)$ be a predicate description for f. Then f is analyzable on $\bar{U}_\delta(A)$ with the following bounding functions

$$\begin{aligned} \nu_f(\gamma) &:= 2d\gamma, & (2.20)\\ \varphi_f(\gamma) &:= |a_d| \cdot \gamma^d, \quad \textit{and} & (2.21)\\ S_{\inf f}(L) &:= (d+2) \max_{1 \le i \le d} |a_i| \ 2^{(d+1)e_{\max}+1-L}. \end{aligned}$$

Proof. For a moment, we consider the complex continuation of the polynomial, i.e., $f \in \mathbb{C}[z]$. Because of the fundamental theorem of algebra (see, e.g., Lamprecht [53]), we can factorize f in the way

$$f(z) \quad = \quad a_d \cdot \prod_{i=1}^{d} (z - \zeta_i)$$

since f has d (not necessarily distinct) roots $\zeta_i \in \mathbb{C}$. Let $\gamma \in \mathbb{R}_{>0}$. Then we can lower bound the absolute value of f by

$$|f(z)| \quad \ge \quad |a_d| \cdot \gamma^d$$

for all $z \in \mathbb{C}$ whose distance to every (complex) root of $f(z)$ is at least γ. This estimate is especially true for real arguments x whose distance to the orthogonal projection of the complex roots ζ_i onto the real axis is at least γ. So, we set the critical set to[16] $C_f(\bar{x}) := \{\mathrm{Re}(\zeta_i) : 1 \le i \le d\} \cap \bar{U}_\delta(\bar{x})$. This validates the bound φ_f and implies that f is value-suitable.

Furthermore, the size of C_f is upper-bounded by d for all $\bar{x} \in A$. This validates the bound ν_f. Because ν_f is invertible, f is region-suitable.

The bounding function $S_{\inf f}(L)$ is proven in Corollary 2.18 in Section 2.9. Because $S_{\inf f}(L)$ is invertible, f is also safety-suitable. As a consequence, f is *analyzable* with the given bounds. □

We admit that we have chosen a quite simple example; however, a more complex example would have been a waste of energy since we present *three general approaches to derive the bounding functions for the region- and value-suitability* in Sections 2.6, 2.7, and 2.8. That means, for more complex examples we use more convenient tools. A well-known approach to derive the bounding function for the safety-bound is given in Section 2.9.

2.4.2 Overview: Function Properties

At this point, we have introduced all properties that are necessary to precisely characterize functions in the context of the analysis. So, let us review what we have defined and related so far. We have summarized the most important implications in Figure 2.9. Controlled perturbation is *applicable* to a certain class of functions.

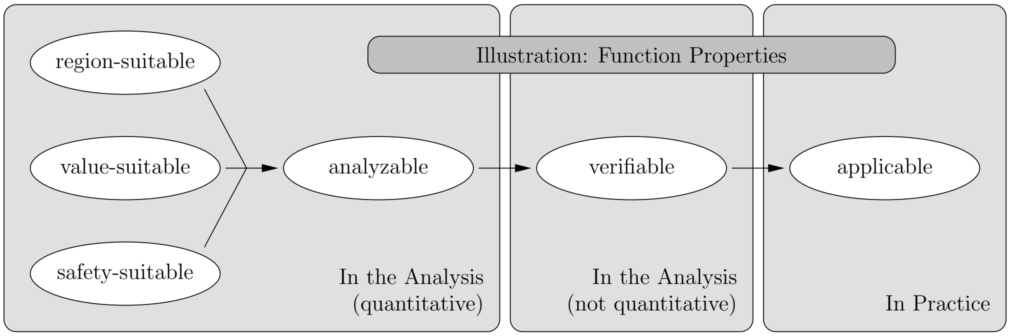


Figure 2.9: The illustration summarizes the implications of the various function properties that we have defined in this chapter. A function that is at the same time region-, value- and safety-suitable is also analyzable (see Definition 2.16). An analyzable function is also verifiable (see Lemma 2.4). A verifiable function is also applicable (see Lemma 2.3).

For a subset of those functions, we can *verify* that controlled perturbation works in practice—without the necessity, or even ability, to analyze their performance. We remember that no condition on the absolute value is needed for verifiability because it is not a quantitative property. A subset of the verifiable functions represents the

[16] Complex Analysis: The function $\mathrm{Re}(z)$ maps a complex number z to its real part. For example, see Fischer et al. [29].

set of *analyzable* functions in a quantitative sense. For those functions, there are *suitable bounds* on the maximum volume of the region of uncertainty, on the minimum absolute value outside of this region, and on the maximum rounding error. In the remaining part of the chapter, we are only interested in the class of analyzable functions.

2.5 The Method of Quantified Relations

The method of quantified relations actually performs the function analysis in the second stage. The component and its interface are illustrated in Figure 2.10. We introduce the method in Section 2.5.1. Its input consists of three bounding functions that are associated with the three suitability properties from the last section. The applicability does not depend on any other condition. The method provides general instructions to relate the three given bounds. The prime objective is to derive a relation between the probability of a successful floating-point evaluation and the precision of the floating-point arithmetic. More precisely, the method provides a precision function $L(p)$ or a probability function $p(L)$. *These are the first step-by-step instructions for the second stage of the function analysis.* An example of its application follows in Section 2.5.3.

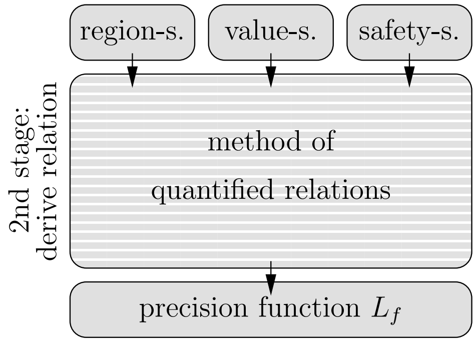


Figure 2.10: The method of quantified relations and its interface.

2.5.1 Presentation

There are no further prerequisites than the three necessary suitability properties from the last section. Therefore, we can immediately state the main theorem of this section, whose proof contains the method of quantified relations.

Theorem 2.6 (quantified relations)**.** *Let* $(f, k, A, \delta, e_{\max}, \Gamma\text{-line}, t)$ *be a predicate description, and let* f *be analyzable. Then there is a method to determine a precision function* $L_f : (0,1) \to \mathbb{N}$ *such that the guarded evaluation of* f *at a randomly perturbed input is successful with probability at least* $p \in (0,1)$ *for every precision* $L \in \mathbb{N}$ *with* $L \geq L_f(p)$*.*

Proof. We show in six steps how we can determine a precision function $L_f(p)$ that has the property: If we use a floating-point arithmetic with precision $L_f(p)$ for a given $p \in (0,1)$, the evaluation of $f(x)|_{\mathbb{G}}$ is guarded with success probability of at least p for a randomly chosen $x \in \bar{U}_\delta(\bar{x})|_{\mathbb{G}}$ and for any $\bar{x} \in A$. An overview of the steps is given in Table 2.1. We usually begin with Step 1. However, there is an exception: In the special case that $\nu_f \equiv 0$, we know that the bounding function φ is

Step 1: relate probability with volume of region of uncertainty (define ε_ν)
Step 2: relate volume of region of uncertainty with distances (define γ)
Step 3: relate distances with floating-point grid (choose t)
Step 4: relate new distances with minimum absolute value (define φ)
Step 5: relate minimum absolute value with precision (define L_{safe})
Step 6: relate L_{safe} with L_{grid} (define L_{grid} and L_f)

Table 2.1: Instructions for performing the method of quantified relations.

constant, see Remark 2.2.4 for details. Then we simply skip the first four steps and begin with Step 5.

Step 1 (define ε_ν). We derive an upper bounding function $\varepsilon_\nu(p)$ on the volume of the augmented region of uncertainty from the success probability p in the way

$$\begin{aligned} \varepsilon_\nu(p) &:= (1-p)\cdot\mu(U_\delta) \\ &= (1-p)\cdot\prod_{i=1}^{k}(2\delta_i). \end{aligned} \tag{2.22}$$

That means, a randomly chosen point $x \in U_\delta(\bar{x})$ lies inside of a given region of volume $\varepsilon_\nu(p)$ with probability at least p. The argumentation of this step is based on *real* arithmetic.

Step 2 (define γ). We know that there is $\gamma \in \mathbb{R}^k_{>0}$ that fulfills the region-condition in Definition 2.12 because f is verifiable. Since f is even region-suitable, we can also determine $\gamma \in \Gamma$-line by means of the inverse of the bounding function ν_f. The existence and invertibility of ν_f is guaranteed by Definition 2.13. Hence, we define the function

$$\gamma(p) := \nu_f^{-1}(\varepsilon_\nu(p)) \in \Gamma\text{-line}. \tag{2.23}$$

We remember that there is an alternative definition of the region-suitability we mentioned in Remark 2.2.4. Surely, it is also possible to use the bounding function χ_f instead of ν_f in the method of quantified relations directly; the alternative Steps $1'$ and $2'$ are introduced in Section 2.5.2.

Step 3 (choose t). We aim for a result that is valid for floating-point arithmetic although we base the analysis on real arithmetic (see Section 2.2.4). We choose[17] $t \in (0,1)$ and define $R_{t\gamma}$ as the normal sized region of uncertainty. Due to Theorem 2.1, the probability that a random point $x \in U_\delta(\bar{x})|_{\mathbb{G}}$ lies inside of $R_{t\gamma}(\bar{x})|_{\mathbb{G}}$ is smaller than the probability that a random point $x \in U_\delta(\bar{x})$ lies inside of $R_\gamma(\bar{x})$. Consequently, if a randomly chosen point lies outside of the augmented region of uncertainty with probability p, it lies outside of the normal sized region of uncertainty with probability at least p. Our next objective is to guarantee a floating-point safe evaluation outside of the *normal sized* region of uncertainty.

[17] The analysis works for any choice. However, finding the best choice is an optimization problem.

Step 4 (define φ). We now want to determine the minimum absolute value outside of the region of uncertainty $R_{t\gamma}(\bar{x})$. We have proven in Lemma 2.2 that a positive minimum exists. Because f is value-suitable, we can use the bounding function φ_f for its determination (see Definition 2.14). That means, we consider

$$\varphi(p) \quad := \quad \varphi_f(t \cdot \gamma(p)).$$

Step 5 (define L_{safe}). So far, we have fixed the region of uncertainty and have determined the minimum absolute value outside of this region. We now can use the safety-condition from Definition 2.12 to determine a precision L_{safe}, which implies fp-safe evaluations outside of $R_{t\gamma}$. That means, we want Formula (2.14) to be valid for every $L \in \mathbb{N}$ with $L \geq L_{\text{safe}}$. We again use the property that f is analyzable and use the inverse of the fp-safety bound $S^{-1}_{\inf f}$ in Definition 2.15 to deduce the precision from the minimum absolute value $\varphi(p)$ as

$$L_{\text{safe}}(p) \quad = \quad \left\lceil S^{-1}_{\inf f}\left(\varphi_f\left(t \cdot \nu_f^{-1}\left(\varepsilon_\nu\left(p\right)\right)\right)\right)\right\rceil . \tag{2.24}$$

Step 6 (define L_{grid} and L_f). We numerate the component functions of ν_f^{-1} in the way $\nu_f^{-1}(\varepsilon) = (\nu_1^{-1}(\varepsilon), \ldots, \nu_k^{-1}(\varepsilon))$. Then we deduce the bound L_{grid} from Formula (2.10) and Formula (2.23) in the way

$$L_{\text{grid}}(p) \quad := \quad \left\lceil e_{\max} - 1 - \log_2\left((1-t) \cdot \min_{1 \leq i \leq k} \nu_i^{-1}(\varepsilon_\nu(p))\right)\right\rceil . \tag{2.25}$$

Finally, we define the *precision function* $L_f(p)$ pointwise as

$$L_f(p) \quad := \quad \max\left\{L_{\text{safe}}(p), L_{\text{grid}}(p)\right\}. \tag{2.26}$$

Due to the used estimates, any precision $L \in \mathbb{N}$ with $L \geq L_f(p)$ is a solution. □

2.5.2 Properties

We focus our attention on four properties of the method of quantified relations which are important for the general analysis.

First, L_{safe} is derived from the *volume* of R_f and is based on the region- and safety condition in Definition 2.12, whereas L_{grid} is derived from the *narrowest width* of R_f and is based on the grid unit condition in Section 2.2.4. Of course, $L_f(p)$ must be large enough to fulfill both constraints.

Second, as we have seen, we can also use the function χ_f to define the region-suitability in Definition 2.13. Therefore, we can modify the first two steps of the method of quantified relations as follows:

Step 1′ (define ε_χ). Instead of Step 1, we define a bounding function $\varepsilon_\chi(p)$ on the volume of the complement of R_f from the given success probability p. That means, we replace Formula (2.22) with

$$\begin{aligned} \varepsilon_\chi(p) \quad &:= \quad p \cdot \mu(U_\delta) \\ &= \quad p \cdot \prod_{i=1}^{k}(2\delta_i)\,. \end{aligned}$$

Step 2′ (define γ). Then we can determine $\gamma(p)$ with the inverse of the bounding function χ_f. That means, we replace Formula (2.23) with

$$\gamma(p) \quad := \quad \chi_f^{-1}(\varepsilon_\chi(p)) \in \Gamma\text{-line},$$

which finally changes Formula (2.24) into

$$L_{\text{safe}}(p) \quad = \quad \left\lceil S_{\inf f}^{-1}\left(\varphi_f\left(t \cdot \chi_f^{-1}\left(\varepsilon_\chi\left(p\right)\right)\right)\right)\right\rceil$$

and Formula (2.25) into

$$L_{\text{grid}}(p) \quad := \quad \left\lceil e_{\max} - 1 - \log_2\left((1-t) \cdot \min_{1 \leq i \leq k} \chi_i^{-1}(\varepsilon_\chi(p))\right)\right\rceil . \qquad (2.27)$$

All of these changes do not affect the correctness of the method of quantified relations.

Third, the method of quantified relations is absolutely independent of the derivation of the bounding functions which are associated with the necessary suitability properties. Especially in Step 2, γ is determined solely by means of the function ν^{-1}. We illustrate this generality with the examples in Figure 2.11. The three pictures show different regions of uncertainty for the *same* critical set and the *same* volume ε_ν. This is because the region of uncertainties result from different functions ν^{-1}. We could say that the function ν^{-1} “knows” how to distribute the region of uncertainty around the critical set because of its definition in the first stage of the analysis, for example: (a) as local neighborhoods, (b) as axis-parallel stripes, or (c) as neighborhoods of local minima of f (the dashed line). (We remark that case (c) presumes that f is continuous.) Different functions ν^{-1} naturally lead to different values of γ as is illustrated in the pictures. Be aware that the method of quantified relations itself is absolutely independent of the *derivation* of ν and especially independent of the *approach* by which ν is derived. (We will soon present three different approaches.)

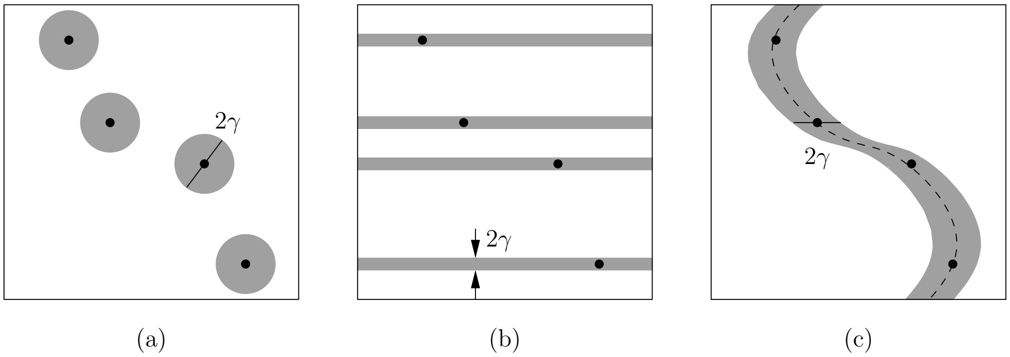


Figure 2.11: Visualization of $\nu^{-1}(\varepsilon_\nu)$ in Step 2 of the method of quantified relations.

Forth, the components of the analysis framework can often be replaced by alternative components. This is why we call the framework analysis *tool box*. Consider, for example, the variation of the method of quantified relations that derives the success probability p from a precision L. In addition to the analyzability of f, we merely require that φ_f is invertible. We observe that the function ε_ν in Formula (2.22) is always invertible. Therefore, we can transform Formula (2.24) and (2.25) into

$$\begin{aligned} p_{\text{inf}}(L) &:= \varepsilon_\nu^{-1}\left(\nu_f\left(\frac{1}{t}\cdot\varphi_f^{-1}\left(S_{\inf f}(L)\right)\right)\right) \\ p_{\text{grid}}(L) &:= \varepsilon_\nu^{-1}\left(\nu_*\left(\frac{2^{-L+e_{\max}-1}}{1-t}\right)\right), \end{aligned}$$

respectively, where ν_*^{-1} is the least growing component function of ν_f^{-1} and ν_* is the inversion of ν_*^{-1}. This leads to the (preliminary) *probability function* $p_f : \mathbb{N} \to (0,1)$,

$$p_f(L) \quad := \quad \min\{p_{\text{inf}}(L), p_{\text{grid}}(L)\}$$

for parameter $t \in (0,1)$. We develop the final version of the probability function in Section 2.10.2. Of course, we can also derive appropriate bounding functions for χ instead of ν as we have explained in the second remark.

2.5.3 Example

In order to become familiar with the usage of the method of quantified relations, we give a detailed application in the proof of the following lemma.

Lemma 2.7. *Let f be a univariate polynomial of degree d as shown in Formula (2.19), and let $(f, k, A, \delta, e_{\max}, \Gamma\text{-line}, t)$ be a predicate description. Then we obtain for f:*

$$L_{\text{safe}}(p) \quad := \quad \lceil -d\log_2(1-p) \;+\; c_{\text{u}} \rceil \tag{2.28}$$

where

$$c_{\text{u}} \quad := \quad \log_2 \frac{(d+2)\cdot\max_{1\le i\le d}|a_i|\cdot 2^{(d+1)e_{\max}+1}}{|a_d|\cdot(t\delta/d)^d}.$$

Proof. The polynomial f is analyzable because of Lemma 2.5. Therefore, we can determine L_{safe} with the first 5 steps of the method of quantified relations (see Theorem 2.6).

Step 1: Since the perturbation area $U_\delta(\bar{x})$ is an interval of length 2δ, the region of uncertainty has a volume of at most

$$\varepsilon_\nu(p) \quad := \quad 2\delta(1-p).$$

Step 2: We next deduce γ from the inverse of the function in Formula (2.20), which means, from $\nu_f^{-1}(\varepsilon) = \frac{\varepsilon}{2d}$. We obtain

$$\gamma(p) \quad := \quad \nu_f^{-1}(\varepsilon_\nu(p)) \quad = \quad \frac{\varepsilon_\nu(p)}{2d} \quad = \quad \frac{\delta(1-p)}{d}.$$

Step 3: We choose $t \in (0,1)$.
Step 4: Due to Formula (2.21), the absolute value of f outside of the region of uncertainty is lower-bounded by the function

$$\varphi(p) \quad := \quad |a_d| \cdot (t \cdot \gamma(p))^d \quad = \quad |a_d| \cdot \left(\frac{t\delta(1-p)}{d} \right)^d .$$

Step 5: A fp-safety bound $S_{\inf f}$ is provided by Corollary 2.18 on page 82. The inverse of this function at $\varphi(p)$ is

$$S^{-1}_{\inf f}(\varphi(p)) \quad = \quad \log_2 \frac{(d+2) \cdot \max_{1 \le i \le d} |a_i| \cdot 2^{(d+1)e_{\max}+1}}{\varphi(p)}.$$

Due to Formula (2.24), this leads to

$$\begin{aligned} L_{\mathrm{safe}}(p) \quad &:= \quad \left\lceil S^{-1}_{\inf f}(\varphi(p)) \right\rceil \\ &= \quad \left\lceil \log_2 \frac{(d+2) \cdot \max_{1 \le i \le d} |a_i| \cdot 2^{(d+1)e_{\max}+1}}{|a_d| \cdot (t\delta(1-p)/d)^d} \right\rceil \\ &= \quad \left\lceil -d \log_2(1-p) \ + \ \log_2 \frac{(d+2) \cdot \max_{1 \le i \le d} |a_i| \cdot 2^{(d+1)e_{\max}+1}}{|a_d| \cdot (t\delta/d)^d} \right\rceil \end{aligned}$$

as was claimed in the lemma. □

Since the formula for $L_{\mathrm{safe}}(p)$ in the lemma above seems rather complicated, we interpret it here. We observe that c_u is a constant because it is defined only by constants: The degree d and the coefficients a_i are defined by f, and the parameters $e_{\max}$ and δ are given by the input. We make the asymptotic behavior $L_{\mathrm{safe}}(p) = O(-d \log_2(1-p))$ for $p \to 1$ explicit in the following corollary: We show that d additional bits of the precision are sufficient to halve the failure probability.

Corollary 2.8. *Let f be a univariate polynomial of degree d and let $L_{\mathrm{safe}} : (0,1) \to \mathbb{N}$ be the precision function in Formula (2.28). Then*

$$L_{\mathrm{safe}} \left(\frac{1+p}{2} \right) \quad = \quad L_{\mathrm{safe}}(p) + d.$$

Proof. Due to Formula (2.28), we have:

$$\begin{aligned} L_{\mathrm{safe}} \left(\frac{1+p}{2} \right) \quad &= \quad \left\lceil -d \log_2 \left(1 - \left(\frac{1+p}{2} \right) \right) + c_u \right\rceil \\ &= \quad \left\lceil -d \log_2 \left(\frac{1-p}{2} \right) + c_u \right\rceil \\ &= \quad \lceil -d \left(\log_2(1-p) - \log_2(2) \right) + c_u \rceil \\ &= \quad \lceil -d \log_2(1-p) + d + c_u \rceil \\ &= \quad L_{\mathrm{safe}}(p) + d. \end{aligned}$$

Because d is a natural number, we can pull it out of the brackets. □

2.6 The Direct Approach Using Estimates

This approach derives the bounding functions that are associated with region- and value-suitability in the first stage of the analysis (see Figure 2.12). It is partially based on the geometric interpretation of the function f at hand. More precisely, it presumes that the critical set of f is embedded in geometric objects for which we know simple mathematical descriptions (e.g., lines, circles, etc.). The derivation of bounds from geometric interpretations is also presented in [59, 60]. In Section 2.6.1, we explain the derivation of the bounds. In Section 2.6.2, we show some examples.

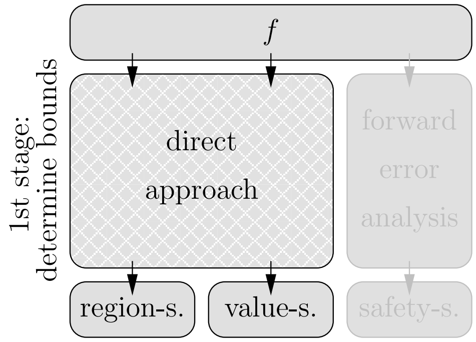


Figure 2.12: The direct approach and its interface.

2.6.1 Presentation

The steps of the direct approach are summarized in Table 2.2. To facilitate the presentation of the geometric interpretation, we assume that the function f is continuous everywhere and that we do not allow any exceptional points. Then the critical set of f equals the zero set of f. Hence, the region of uncertainty is an environment of the zero set in this case. We define the region of uncertainty R_γ as it is defined in Formula (2.8). In the first step, we must define the domain for γ, which we denote by Γ-line. Or in other words, we choose $\hat{\gamma}$. Certain choices of Γ-line may sometimes be more useful than others, e.g., cubic environments where $\hat{\gamma}_i = \hat{\gamma}_j$ for all $1 \leq i, j \leq k$.

Now, assume that we have chosen Γ-line. In the second step, we estimate (an upper bound on) the volume of the region of uncertainty R_γ by a function $\nu_f(\gamma)$ for $\gamma \in \Gamma$-line. In the direct approach, we hope that a geometric interpretation of the zero set supports the estimation. For that purpose it would be helpful if the region of uncertainty were embedded in a line, a circle, or any other geometric structure that we could easily describe mathematically.

Assume further that we have fixed the bound ν_f. In the third step, we need to determine a function $\varphi_f(\gamma)$ on the minimum absolute value of f outside of R_γ. This is the most difficult step in the direct approach: *Although geometric interpretation may be helpful in the second step, mathematical considerations are necessary to derive* φ_f. Therefore, we hope that φ_f is "obvious" enough to be guessed. If there is no chance to guess φ_f, we need to try one of the alternative approaches from the next sections, namely, the bottom-up approach or the top-down approach.

Step 1: choose the set Γ-line (define $\hat{\gamma}$)
Step 2: estimate $\nu_f(\gamma)$ in dependence on Γ-line (define ν_f)
Step 3: estimate $\varphi_f(\gamma)$ in dependence on $\nu_f(\gamma)$ (define φ_f)

Table 2.2: Instructions for performing the direct approach.

2.6.2 Examples

We present two examples that use the direct approach to derive the bounds for the region-value-suitability.

Example 2.7. We consider the in_box predicate in the plane. Let u and v be two opposite vertices of the box, and let q be the query point. Then in_box(u, v, q) is decided by the sign of the function

$$\begin{aligned} f(u,v,q) &= f(u_x, u_y, v_x, v_y, q_x, q_y) \\ &:= \max\left\{(q_x - u_x)(q_x - v_x)\,,\ (q_y - u_y)(q_y - v_y)\right\}. \end{aligned} \tag{2.29}$$

The function is negative if x lies inside of the box, it is zero if x lies in the boundary, and it is positive if x lies outside of the box.

Step 1: We choose an arbitrary $\hat{\gamma} = (\hat{\gamma}_{u_x}, \hat{\gamma}_{u_y}, \hat{\gamma}_{v_x}, \hat{\gamma}_{v_y}, \hat{\gamma}_{q_x}, \hat{\gamma}_{q_y}) \in \mathbb{R}^6_{>0}$.

Step 2: The box is defined by u and v. This fact is truly independent of the choices for γ_{u_x}, γ_{u_y}, γ_{v_x} and γ_{v_y}. We observe that the largest box inside of the perturbation area U_δ is the boundary of U_δ itself. This observation leads to the upper bound

$$\begin{aligned} \nu_f(\gamma) &= \nu_f(\gamma_{u_x}, \gamma_{u_y}, \gamma_{v_x}, \gamma_{v_y}, \gamma_{q_x}, \gamma_{q_y}) \\ &:= 4\left(\gamma_{q_x}\delta_y \ + \ \gamma_{q_y}\delta_x\right) \end{aligned}$$

on the volume of the region of uncertainty if we take into account the horizontal distance γ_{q_x} and the vertical distance γ_{q_y} from the boundary of the box. That means, ν_f depends on the distances γ_{q_x} and γ_{q_y} of the query point q from the zero set.

Step 3: The evaluation of Formula (2.29) at query points where q_x has distance γ_{q_x} from u_x or v_x, and q_y has distance γ_{q_y} from u_y or v_y, leads to

$$\varphi_f(\gamma) \ := \ \min\left\{\left|\gamma_{q_x}^2 - \gamma_{q_x}\cdot|v_x - u_x|\right|,\ \left|\gamma_{q_y}^2 - \gamma_{q_y}\cdot|v_y - u_y|\right|\right\}.$$

The derived bounds fulfill the desired properties. ○

Example 2.8. We consider the in_circle predicate in the plane. Let c be the center of the circle, let $r > 0$ be its radius, and let q be the query point. Then in_circle(c, r, q) is decided by the sign of the function

$$\begin{aligned} f(c, r, q) \quad &= \quad f(c_x, c_y, r, q_x, q_y) \\ &:= \quad (q_x - c_x)^2 \;+\; (q_y - c_y)^2 \;-\; r^2. \end{aligned} \tag{2.30}$$

The function is negative if x lies inside of the circle, it is zero if x lies on the circle, and it is positive if x lies outside of the circle.

Step 1: We choose $\hat{\gamma} = (\hat{\gamma}_{c_x}, \hat{\gamma}_{c_y}, \hat{\gamma}_r, \hat{\gamma}_{q_x}, \hat{\gamma}_{q_y}) \in \mathbb{R}^5_{>0}$ where $\hat{\gamma}_{q_x} = \hat{\gamma}_{q_y}$. In addition, we choose $\hat{\gamma}_r < r$ for simplicity.

Step 2: The largest circle that fits into the perturbation area U_δ has radius $\min\{\delta_x, \delta_y\}$. If we intersect any larger circle with U_δ, the total length of the circular arcs inside of U_δ cannot be larger than $2\pi \cdot \min\{\delta_x, \delta_y\}$. This bounds the total length of the zero set.

We now define the region of uncertainty by spherical environments: The region of uncertainty is the union of open discs of radius γ_{q_x}, which are located at the zeros. Then the width of the region of uncertainty is given by the diameter of the discs, i.e., by $2\gamma_{q_x}$. As a consequence,

$$\begin{aligned} \nu_f(\gamma) \quad &= \quad \nu_f(\gamma_{c_x}, \gamma_{c_y}, \gamma_r, \gamma_{q_x}, \gamma_{q_y}) \\ &:= \quad 4\pi\gamma_{q_x} \cdot \min\{\delta_x, \delta_y\} \end{aligned}$$

is an upper bound on the volume of R_δ. That means, ν_f depends on the distance γ_{q_x} of the query point q from the zero set.

Step 3: The absolute value of Formula (2.30) is minimal if the query point q lies inside of the circle and has distance γ_{q_x} from it. This leads to

$$\begin{aligned} \varphi_f(\gamma) \quad &:= \quad \left|(r \;-\; \gamma_{q_x})^2 \;-\; r^2\right| \\ &= \quad \gamma_{q_x}\left(\gamma_{q_x} - 2r\right). \end{aligned}$$

The derived bounds fulfill the desired properties. ○

2.7 The Bottom-up Approach Using Calculation Rules

In the first stage of the analysis, this approach derives the bounding functions associated with the region- and value-suitability (see Figure 2.13). We can apply this approach to certain composed functions. That means, if f is composed by g and h, we can derive the bounds for f from the bounds for g and h under certain conditions. We present some mathematical constructs that preserve the region- and value-suitability and introduce useful calculation rules for their bounds. Namely, we introduce the *lower-bounding rule* in Section 2.7.1, the *product rule* in Section 2.7.2, and the *min rule* and *max rule* in Section 2.7.3. We point to a general way to formulate rules in Section 2.7.4. The list of rules is by far not complete. Nevertheless, these rules are already sufficient to derive the bounding functions for multivariate polynomials, as we show in Section 2.7.5. *With the bottom-up approach, we present an entirely new approach to derive the bounding functions for the region-suitability and value-suitability. Furthermore, we present a new way to analyze multivariate polynomials.*

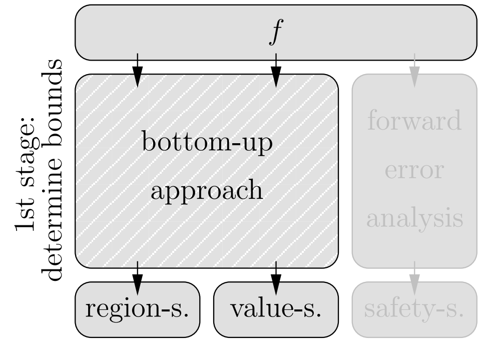


Figure 2.13: The bottom-up approach and its interface.

2.7.1 Lower-bounding Rule

Our first rule states that every function is region-value-suitable if there is a lower bounding function that is region-value-suitable. Note that there are no further restrictions on f.

Theorem 2.9 (lower bound)**.** *Let* $(f, k, A, \delta, e_{\max}, \Gamma\text{-line}, t)$ *be a predicate description. If there is a region-value-suitable function* $g : \bar{U}_\delta(A) \to \mathbb{R}$ *and* $c \in \mathbb{R}_{>0}$ *where*

$$|f(x)| \quad \geq \quad c\,|g(x)|, \tag{2.31}$$

then f is also region-value-suitable with the following bounding functions:

$$\begin{aligned} \nu_f(\gamma) &:= \nu_g(\gamma) \\ \varphi_f(\gamma) &:= c\varphi_g(\gamma). \end{aligned}$$

If f is in addition safety-suitable, f is analyzable.

Proof. Part 1 (region-suitable). Let $(a_i)_{i\in\mathbb{N}}$ be a sequence in the set $U_\delta(\bar{x})$ with $\lim_{i\to\infty} f(a_i) = 0$. Then Formula (2.31) implies that $\lim_{i\to\infty} g(a_i) = 0$. That means, critical points of f are critical points of g. Therefore, we set $C_f(\bar{x}) := C_g(\bar{x})$. As a consequence, the region bound $\nu_f(\gamma) := \nu_g(\gamma)$ is sufficient for the region-suitability of f.

Part 2 (value-suitable). Because we set $C_f(\bar{x}) = C_g(\bar{x})$, we have $R_f(\bar{x}) = R_g(\bar{x})$. Due to Formula (2.31), the minimum absolute value of f outside of the region of uncertainty $R_f(\bar{x})$ is bounded by the minimum absolute value of g outside of the (same) region of uncertainty $R_g(\bar{x})$. Hence, the bound $\varphi_f(\gamma) = c\varphi_g(\gamma)$ is sufficient for the value-suitability of f.

Part 3 (analyzable). Trivial. □

2.7.2 Product Rule

The next rule states that the product of region-value-suitable functions is also region-value-suitable. Furthermore, we show how to derive appropriate bounds.

Theorem 2.10 (product)**.** *Let $(f, k, A_g \times A_{gh} \times A_h, \delta, e_{\max}, \Gamma\text{-line}, t)$ be a predicate description where $A_g \subset \mathbb{R}^j$, $A_{gh} \subset \mathbb{R}^{\ell-j}$ and $A_h \subset \mathbb{R}^{k-\ell}$ for $j \in \mathbb{N}_0$ and $\ell, k \in \mathbb{N}$ with $j \le \ell \le k$. If there are two region-value-suitable functions*

$$g \quad : \quad \bar{U}_{(\delta_1,\ldots,\delta_\ell)}(A_g \times A_{gh}) \to \mathbb{R}$$

and

$$h \quad : \quad \bar{U}_{(\delta_{j+1},\ldots,\delta_k)}(A_{gh} \times A_h) \to \mathbb{R}$$

such that

$$f(x_1, \ldots, x_k) \quad = \quad g(x_1, \ldots, x_\ell) \cdot h(x_{j+1}, \ldots, x_k),$$

then f is also region-value-suitable with the following bounding functions:

$$\begin{aligned} \varphi_f(\gamma) \quad &:= \quad \varphi_g(\gamma_1, \ldots, \gamma_\ell) \cdot \varphi_h(\gamma_{j+1}, \ldots, \gamma_k) && (2.32) \\ \nu_f(\gamma) \quad &:= \quad \min \left\{ \prod_{i=1}^{k}(2\delta_i), \right. \\ & \qquad \left. \nu_g(\gamma_1, \ldots, \gamma_\ell) \prod_{i=\ell+1}^{k}(2\delta_i) \; + \; \nu_h(\gamma_{j+1}, \ldots, \gamma_k) \prod_{i=1}^{j}(2\delta_i) \right\}. && (2.33) \end{aligned}$$

Furthermore, if $j = \ell$, we can replace the last equation by the tighter bound

$$\chi_f(\gamma) \quad := \quad \chi_g(\gamma_1, \ldots, \gamma_j) \cdot \chi_h(\gamma_{j+1}, \ldots, \gamma_k). \qquad (2.34)$$

If f is in addition safety-suitable, f is analyzable (independent of $j = \ell$).

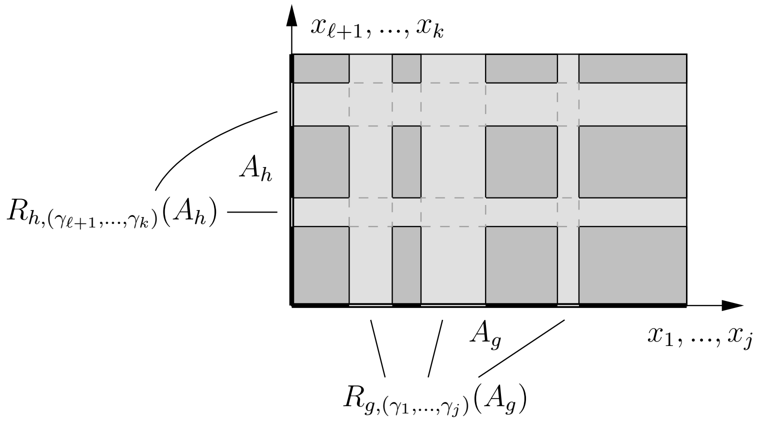

$x_{\ell+1}, ..., x_k$
A_h
$R_{h,(\gamma_{\ell+1},...,\gamma_k)}(A_h)$
A_g
$x_1, ..., x_j$
$R_{g,(\gamma_1,...,\gamma_j)}(A_g)$

Figure 2.14: Case $j = \ell$: The (dark-shaded) complement of R_f is the Cartesian product of the complement of R_g and the complement of R_h.

Proof. Part 1 (value-suitable). Let $x \in U_\delta(\bar{x})$ such that $(x_1, \ldots, x_\ell)$ does not lie in the region of uncertainty[18] of g, i.e.,

$$(x_1, \ldots, x_\ell) \notin R_{g,(\gamma_1,\ldots,\gamma_\ell)}(\bar{x}_1, \ldots, \bar{x}_\ell), \tag{2.35}$$

and that $(x_{j+1}, \ldots, x_k)$ does not lie in the region of uncertainty of h, i.e.,

$$(x_{j+1}, \ldots, x_k) \notin R_{h,(\gamma_{j+1},\ldots,\gamma_k)}(\bar{x}_{j+1}, \ldots, \bar{x}_k). \tag{2.36}$$

Because g and h are value-suitable, we obtain:

$$\begin{aligned} |f(x)| &= |g(x_1, \ldots, x_\ell)| \cdot |h(x_{j+1}, \ldots, x_k)| \\ &\geq \varphi_g(\gamma_1, \ldots, \gamma_\ell) \cdot \varphi_h(\gamma_{j+1}, \ldots, \gamma_k) \\ &= \varphi_f(\gamma) \end{aligned}$$

on the absolute value of f.

Part 2 (region-suitable). Because of the argumentation above, we must construct the region of uncertainty R_f such that $x \in \mathbb{R}^k$ lies outside of R_f only if the conditions in Formula (2.35) and (2.36) are fulfilled.

Case $j = \ell$. Then the arguments of g and h are disjoint. This case is illustrated in Figure 2.14. We observe that for each point $(x_1, \ldots, x_j)$ outside of R_g and for each point $(x_{\ell+1}, \ldots, x_k)$ outside of R_h, their concatenation x lies outside of R_f. Therefore, we determine the volume of the complement of R_f inside of the perturbation

[18] To avoid confusion, we occasionally add the function name to the index of the region of uncertainty or the perturbation area within the proof, e.g. $R_{f,\gamma}$ and $U_{f,\delta}$.

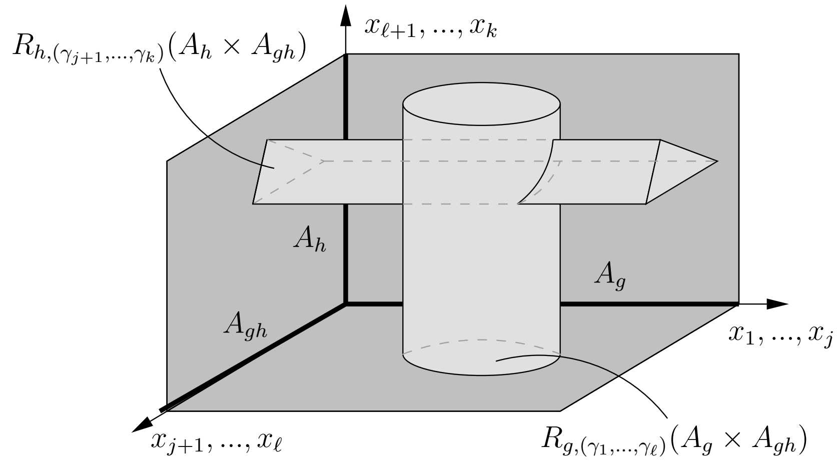

$R_{h,(\gamma_{j+1},\ldots,\gamma_k)}(A_h \times A_{gh})$
$x_{\ell+1}, \ldots, x_k$
A_h
A_g
A_{gh}
$x_1, \ldots, x_j$
$x_{j+1}, \ldots, x_\ell$
$R_{g,(\gamma_1,\ldots,\gamma_\ell)}(A_g \times A_{gh})$

Figure 2.15: Case $j < \ell$: The (light-shaded) region of uncertainty R_f is the union of two Cartesian products.

area as

$$\begin{aligned}\mu\left(U_{f,\delta}(\bar{x}) \setminus R_{f,\gamma}(\bar{x})\right) \quad = \quad & \mu\Big(U_{g,(\delta_1,\ldots,\delta_j)}(\bar{x}_1, \ldots, \bar{x}_j) \\ & \qquad \setminus R_{g,(\gamma_1,\ldots,\gamma_j)}(\bar{x}_1, \ldots, x_j)\Big) \\ & \cdot \mu\left(U_{h,(\delta_{\ell+1},\ldots,\delta_k)}(\bar{x}_{\ell+1}, \ldots, \bar{x}_k)\right. \\ & \qquad \left.\setminus R_{h,(\gamma_{\ell+1},\ldots,\gamma_k)}(\bar{x}_{\ell+1}, \ldots, x_k)\right).\end{aligned}$$

As a consequence, Formula (2.34) is true.

Case $j < \ell$. In contrast to the discussion above, g and h share the arguments $x_{j+1}, \ldots, x_\ell$. This case is illustrated in Figure 2.15. We denote the projection of the first j (respectively, the last $k - \ell$) coordinates by $\pi_{\leq j}$ (respectively, $\pi_{>\ell}$). In this case, Formula (2.34) does not have to be true. This is why we define R_f as

$$\begin{aligned}R_{f,\gamma}(\bar{x}) \quad := \quad & R_{g,(\gamma_1,\ldots,\gamma_\ell)}(\bar{x}_1, \ldots, \bar{x}_\ell) \times \pi_{>\ell}(\bar{U}_\delta(\bar{x})) \\ & \cup\, \pi_{\leq j}(\bar{U}_\delta(\bar{x})) \times R_{h,(\gamma_{j+1},\ldots,\gamma_k)}(\bar{x}_{j+1}, \ldots, \bar{x}_k).\end{aligned}$$

We now can upper-bound the volume of R_f by means of ν_g and ν_h, which leads immediately to the sum in the last line of Formula (2.33). Of course, the volume of the region of uncertainty is bounded by the volume of the perturbation area, which justifies the first line of Formula (2.33). This finishes the proof. □

2.7.3 Min Rule, Max Rule

The next two rules state that the minimum and maximum of finitely many region-value-suitable functions are also region-value-suitable. Furthermore, we show how to derive appropriate bounds.

Theorem 2.11 (min, max)**.** *Let g and h be two region-value-suitable functions as defined in Theorem 2.10. Then the functions*

$$\begin{aligned} f_{\min}, f_{\max} &: \bar{U}_\delta(A_g \times A_{gh} \times A_h) \to \mathbb{R}, \\ f_{\min}(x_1, \dots, x_k) &:= \min\{g(x_1, \dots, x_\ell), h(x_{j+1}, \dots, x_k)\} \\ f_{\max}(x_1, \dots, x_k) &:= \max\{g(x_1, \dots, x_\ell), h(x_{j+1}, \dots, x_k)\} \end{aligned}$$

are region-value-suitable with bounds $\varphi_{f_{\min}}$ and $\nu_{f_{\min}}$ for $f_{\min}$ and bounds $\varphi_{f_{\max}}$ and $\nu_{f_{\max}}$ for $f_{\max}$ where

$$\begin{aligned} \varphi_{f_{\min}}(\gamma) &:= \min\{\varphi_g(\gamma_1, \dots, \gamma_\ell), \varphi_h(\gamma_{j+1}, \dots, \gamma_k)\} \\ \varphi_{f_{\max}}(\gamma) &:= \max\{\varphi_g(\gamma_1, \dots, \gamma_\ell), \varphi_h(\gamma_{j+1}, \dots, \gamma_k)\} \qquad (2.37) \\ \nu_{f_{\min}}(\gamma) := \nu_{f_{\max}}(\gamma) &:= \nu_f(\gamma) \text{ (see Formula (2.33)).} \end{aligned}$$

Furthermore, if $j = \ell$, we can replace $\nu_{f_{\min}}(\gamma)$ and $\nu_{f_{\max}}(\gamma)$ by the tighter bounds

$$\chi_{f_{\min}}(\gamma) := \chi_{f_{\max}}(\gamma) := \chi_g(\gamma_1, \dots, \gamma_j) \cdot \chi_h(\gamma_{j+1}, \dots, \gamma_k).$$

If $f_{\min}$ (respectively $f_{\max}$) is in addition safety-suitable, it is also analyzable (independent of $j = \ell$).

Proof. The line of argumentation follows the proof of Theorem 2.10 exactly. □

2.7.4 General Rule

We do not claim that the list of rules is complete. On the contrary, we suggest that the approach may be extended by further rules. We emphasize that the bottom-up approach is constructive; we build new region-value-suitable functions from already proven region-value-suitable functions. The argumentation always follows the proof of the product rule, which means that the compound of g and h inherits the desired property from g and h: (a) *outside of the union* of the regions of uncertainty for *shared* arguments, and (b) *inside of the Cartesian product of the complement* of the regions of uncertainty for *disjoint* arguments (see Figure 2.14).

We remark that if we want to derive the bounds for a specific function f, we first need to determine the parse tree of f according to the known rules; this may be a non-obvious task in general. The instructions of the bottom-up approach are summed up in Table 2.3.

Step 1:	determine parse tree according to the rules
Step 2:	determine bounds bottom-up according to the parse tree

Table 2.3: Instructions for performing the bottom-up approach.

2.7.5 Example: Multivariate Polynomials

It is important to see that the rules lead to a generic approach to constructing entire classes of region-value-suitable functions. In the following, we use this approach to analyze multivariate polynomials. (A different way to analyze multivariate polynomials was presented before in [60].) So far, we know that univariate polynomials are region-value-suitable. We now show how we transfer the region-value-suitability property of $(k-1)$-variate polynomials to k-variate polynomials by means of the product rule and the lower bound rule. Moreover, we completely analyze k-variate polynomials afterwards.

Preparation

We prepare the analysis of multivariate polynomials with further definitions. Let $k \in \mathbb{N}$. For $\beta \in \mathbb{N}_0^k$ and $x \in \mathbb{R}^k$, we define x^β as the term $x^\beta := x_1^{\beta_1} \cdot \ldots \cdot x_k^{\beta_k}$.

We next define the reverse lexicographic order[19] on k-tuples. Let $\alpha, \beta \in \mathbb{N}_0^k$. Then we define $\alpha \prec \beta$ if and only if there is $\ell \in \{1, \ldots, k\}$ such that $\alpha_j = \beta_j$ for all $\ell < j \leq k$ and $\alpha_\ell < \beta_\ell$.

In addition, we denote by $\mathcal{P}(k)$ the set of bijective functions $\sigma : \{1, \ldots, k\} \to \{1, \ldots, k\}$. In other words, $\mathcal{P}(k)$ is the set of permutations[20] of $\{1, \ldots, k\}$.

Let $\alpha, \beta \in \mathbb{N}_0^k$, and let $\sigma \in \mathcal{P}(k)$. We define the *permutation* σ *of a tuple* $\alpha = (\alpha_1, \ldots, \alpha_k)$ by $\sigma(\alpha) := \big(\alpha_{\sigma^{-1}(1)}, \ldots, \alpha_{\sigma^{-1}(k)}\big)$. We further define the *reverse lexicographic order after the permutation* σ as

$$\alpha \prec_\sigma \beta \qquad :\Longleftrightarrow \qquad \sigma(\alpha) \prec \sigma(\beta).$$

Let $\mathcal{I} \subset \mathbb{N}_0^k$ be finite. We denote the set of largest elements in $\mathcal{I}$ by

$$\mathcal{I}_{\max} \quad := \quad \{\beta \in \mathcal{I} \,:\, \text{there is } \sigma \in \mathcal{P}(k) \text{ such that } \alpha {\prec_\sigma} \beta \text{ for all } \alpha \in \mathcal{I}, \alpha \neq \beta\}\,.$$

We observe that there may be $\beta \in \mathcal{I}$ which do not belong to $\mathcal{I}_{\max}$. We further observe that different permutations may lead to the same local maximum. For each $\beta \in \mathcal{I}_{\max}$, we collect these permutations in the set

$$\mathcal{P}_\beta(k) \quad := \quad \{\sigma \in \mathcal{P}(k) \,:\, \beta = \max\nolimits_{\prec_\sigma} \mathcal{I}\}\,.$$

The region- and value-suitability

We prove that all multivariate polynomials are region-value-suitable.

Lemma 2.12. *Let* $(f, k, A, \delta, e_{\max}, \Gamma\text{-line}, t)$ *be a predicate description for the* k*-variate polynomial* $(k \geq 2)$

$$f(x) := \sum_{\iota \in \mathcal{I}} a_\iota x^\iota$$

[19] For lexicographic order see Cormen et al. [18].

[20] Algebra: For permutation see Lamprecht [53].

where $\mathcal{I} \subset \mathbb{N}_0^k$ is finite and $a_\iota \in \mathbb{R}_{\neq 0}$ for all $\iota \in \mathcal{I}$. Then f is region-value-suitable. There are bounding functions for every $\beta \in \mathcal{I}_{\max}$:

$$\begin{aligned}\varphi_f(\gamma) &:= |a_\beta| \cdot \gamma^\beta \\ \chi_f(\gamma) &:= \prod_{i=1}^{k} 2\,(\delta_i - \beta_i \gamma_i)\,.\end{aligned}$$

Proof. Preparing consideration. Let $\beta \in \mathcal{I}_{\max}$ and let $\sigma \in \mathcal{P}_\beta(k)$. Once chosen, β and σ are fixed in this proof. Because of the reverse lexicographic order, the maximal exponent of $x_{\sigma(k)}$ in $f(x)$ is $\beta_{\sigma(k)}$. Therefore, we can write f as

$$f(x) = b_{\beta_{\sigma(k)}} \cdot x_{\sigma(k)}^{\beta_{\sigma(k)}} + b_{\beta_{\sigma(k)}-1} \cdot x_{\sigma(k)}^{\beta_{\sigma(k)}-1} + \ldots + b_1 \cdot x_{\sigma(k)} + b_0$$

where the $b_i(x_{\sigma(1)}, \ldots, x_{\sigma(k-1)})$ are $(k-1)$-variate polynomials for $0 \leq i \leq \beta_{\sigma(k)}$. We consider for a moment the complex continuation of the polynomial f, i.e., $f \in \mathbb{C}[z]$. Furthermore, we *assume*[21] that the value of $b_{\beta_{\sigma(k)}}$ is not zero. Then there are $\beta_{\sigma(k)}$ (not necessarily distinct) functions $\zeta_i : \mathbb{C}^{k-1} \to \mathbb{C}$ such that we can write f as

$$f(z) \quad = \quad b_{\beta_{\sigma(k)}}(z_{\sigma(1)}, \ldots, z_{\sigma(k-1)}) \cdot \prod_{i=1}^{\beta_{\sigma(k)}} (z_{\sigma(k)} - \zeta_i(z_{\sigma(1)}, \ldots, z_{\sigma(k-1)})).$$

We remark that if we consider f as a polynomial in $z_{\sigma(k)}$ with parameterized coefficients b_i, then the functions ζ_i define the parameterized roots. Even if the location of the roots is variable, the total number of the roots is definitely bounded by $\beta_{\sigma(k)}$. In case $z_{\sigma(k)}$ has a distance of at least $\gamma_{\sigma(k)}$ to the values ζ_i, we can lower bound the absolute value of f by

$$|f(z)| \quad \geq \quad \left| b_{\beta_{\sigma(k)}}\left(z_{\sigma(1)}, \ldots, z_{\sigma(k-1)}\right)\right| \cdot \gamma_{\sigma(k)}^{\beta_{\sigma(k)}}. \tag{2.38}$$

Therefore, this bound is especially true for real arguments. Before we end the consideration in the complex space, we add a remark. Sagraloff et al. [70, 60] suggested a way to improve this estimate: While preserving the *total* region-bound φ_f, it is possible to redistribute the region of uncertainty around the zeros of f in a way that the amount of the *individual* region-contribution per zero may differ; they have shown that a certain redistribution improves the estimate in Formula (2.38). Next, we use mathematical induction to prove that f is region-value-suitable.

Part 1 (basis). Let $j = 1$. Due to Lemma 2.5 univariate polynomials are region-value-suitable.

Part 2 (inductive step). Let $1 < j \leq k$. We define the function g_j as

$$g_j\left(z_{\sigma(1)}, \ldots, z_{\sigma(j-1)}\right) \quad := \quad b_{\beta_{\sigma(j)}}\left(z_{\sigma(1)}, \ldots, z_{\sigma(j-1)}\right).$$

[21] We discuss this assumption in Part 2 of the proof.

Since g_j is a polynomial in $j-1$ variables, g_j is region-value-suitable by induction. Because of Theorem 2.9, the function $|g_j|$ is region-value-suitable with the same bounds. Furthermore, we define the functions

$$\begin{aligned} h_j(z_{\sigma(j)}) &:= \gamma_{\sigma(j)}^{\beta_{\sigma(j)}} \\ \varphi_{h_j}(\gamma_{\sigma(j)}) &:= \gamma_{\sigma(j)}^{\beta_{\sigma(j)}} \\ \nu_{h_j}(\gamma_{\sigma(j)}) &:= 2\beta_{\sigma(j)}\gamma_{\sigma(j)}. \end{aligned}$$

Obviously, h_j is region-value-suitable. We have $|f_j| \geq |g_j| \cdot h_j$. Then the product $|g_j| \cdot h_j$ is also region-value-suitable because of Theorem 2.10. Be aware that the construction of the estimate in Formula (2.38) is based on the assumption that the coefficient $b_{\beta_{\sigma(j)}}$ of f_j is not zero. We observe that this is only guaranteed outside of the region of uncertainty of g_j. We further observe that the construction in the proof of Theorem 2.9 preserves the region of uncertainty, that means, $R_{g_j} \subset R_{f_j}$. Therefore, the assumption is justified and we can conclude that f_j is region-value-suitable. It remains to be shown that the claimed bounding functions φ_f and ν_f are true.

Part 3 (φ_f). The basis $j=1$ follows from Lemma 2.5:

$$\varphi_{f_1}\left(\gamma_{\sigma(1)}\right) := |a_\beta| \cdot \gamma_{\sigma(1)}^{\beta_{\sigma(1)}}.$$

(Be aware that the real coefficient a_β is contained in every g_j for $1 < j \leq k$.) For the induction step, let $1 < j \leq k$. We need the following observation: Because of the reverse lexicographic order, the maximal exponent of $x_{\sigma(j-1)}$ in the parameterized coefficient $b_{\beta_{\sigma(j)}}(x_{\sigma(1)}, \ldots, x_{\sigma(j-1)})$ is $\beta_{\sigma(j-1)}$. We have

$$\varphi_{f_j}\left(\gamma_{\sigma(1)}, \ldots, \gamma_{\sigma(j)}\right) := |a_\beta| \cdot \prod_{\ell=1}^{j} \gamma_{\sigma(\ell)}^{\beta_{\sigma(\ell)}}.$$

The case $j=k$ proves the claim.

Part 4 (χ_f). The basis $j=1$ follows from Lemma 2.5:

$$\chi_{f_1}\left(\gamma_{\sigma(1)}\right) := 2\left(\delta_{\sigma(1)} - \beta_{\sigma(1)}\gamma_{\sigma(1)}\right).$$

For the induction step, let $1 < j \leq k$. Because the argument list of g_j and h_j are disjoint, we apply Formula (2.34) and obtain

$$\chi_{f_j}\left(\gamma_{\sigma(1)}, \ldots, \gamma_{\sigma(j)}\right) := \prod_{\ell=1}^{j} 2\left(\delta_{\sigma(\ell)} - \beta_{\sigma(\ell)}\gamma_{\sigma(\ell)}\right).$$

The case $j=k$ proves the claim. □

The analysis

We prove the analyzability of multivariate polynomials and apply the approach of quantified relations to derive a precision function.

Theorem 2.13 (multivariate polynomial)**.** *Let f be a k-variate polynomial ($k \geq 2$) of total degree d as defined in Lemma 2.12, and let $(f, k, A, \delta, e_{\max}, \Gamma\text{-line}, t)$ be a predicate description for f with cubical neighborhoods $\delta_i = \delta_j$ and $\gamma_i = \gamma_j$ for all $1 \leq i, j \leq k$. Then f is analyzable. Furthermore, we obtain the bounding function*

$$L_{\text{safe}}(p) \;=\; \lceil -\beta^* \log_2 (1 - \sqrt[k]{p}) \;+\; c_{\text{m}}(\beta) \rceil \tag{2.39}$$

where

$$c_{\text{m}}(\beta) \;:=\; \log_2 \frac{(d + 1 + \lceil \log_2 |\mathcal{I}| \rceil) \cdot |\mathcal{I}| \cdot \max_{\iota \in \mathcal{I}} |a_\iota| \cdot 2^{d e_{\max} + \beta^* + 1} \cdot \hat{\beta}^{\beta^*}}{|a_\beta| \cdot (t\delta_1)^{\beta^*}}$$

and the bound

$$L_{\text{grid}}(p) \;:=\; \left\lceil e_{\max} - 1 - \log_2 \left((1 - t) \cdot \frac{\delta_1 \left(1 - \sqrt[k]{p}\right)}{\hat{\beta}} \right) \right\rceil .$$

for $\beta \in \mathcal{I}_{\max}$, $\beta^* := \sum_{1=i}^{k} \beta_i$, and $\hat{\beta} := \max_{1 \leq i \leq k} \beta_i$.

We observe that $\hat{\beta} \leq d$ and $\beta^* \leq d$. Note that the choice of $\beta \in \mathcal{I}_{\max}$ is an optimization problem: We suggest to choose β such that the constant β^* in the asymptotic bound $L_{\text{safe}}(p) = O\left(-\beta^* \log(1 - \sqrt[k]{p})\right)$ for $p \to 1$ is small.

Proof. Part 1 (analyzable). Let $\beta \in \mathcal{I}_{\max}$. Due to Lemma 2.12, f is region-value-suitable. In addition, Corollary 2.19 provides a fp-safety bound $S_{\inf f}(L)$ for k-variate polynomials in Formula (2.55). The function $S_{\inf f}(L)$ converges to zero and is invertible. It follows that f is safety-suitable and thus analyzable.

Part 2 (analysis). We apply the approach of quantified relations. Let $\delta_1, \gamma_1 \in \mathbb{R}_{>0}$ and $\delta_1 = \delta_i$ and $\gamma_1 = \gamma_i$ for all $1 \leq i \leq k$. In addition, let $\hat{\beta} := \max_{1 \leq i \leq k} \beta_i$. Step 1′: We first derive an upper bound ε_χ on the volume of the complement of the region of uncertainty according to the precision p. We obtain

$$\varepsilon_\chi(p) \;:=\; p \prod_{i=1}^{k} 2\delta_i \;=\; p\,(2\delta_1)^k .$$

Step 2′: Because of the cubical neighborhood, we redefine

$$\chi_f(\gamma) \;:=\; 2^k \left(\delta_1 - \hat{\beta}\gamma_1\right)^k .$$

Then we use ε_χ and χ_f to determine γ_1:

$$\begin{array}{rrcl} & \chi_f(\gamma) & = & \varepsilon_\chi(p) \\ \Leftrightarrow & 2^k\left(\delta_1 - \hat{\beta}\gamma_1\right)^k & = & p\,2^k\,\delta_1^k \\ \Leftrightarrow & \left(1 - \frac{\hat{\beta}\gamma_1}{\delta_1}\right)^k & = & p \\ \Rightarrow & 1 - \frac{\hat{\beta}\gamma_1}{\delta_1} & = & \sqrt[k]{p} \\ \Leftrightarrow & \gamma_1(p) & := & \dfrac{\delta_1\left(1 - \sqrt[k]{p}\right)}{\hat{\beta}}. \end{array} \tag{2.40}$$

Step 3: Since γ represents the augmented region of uncertainty, the normal sized region is induced by $t\gamma$.
Step 4: We fix the bound φ_f on the absolute value and set

$$\begin{aligned} \varphi(p) &= \varphi_f(t\gamma(p)) \\ &= |a_\beta| \cdot (t\gamma(p))^\beta \\ &= |a_\beta| \cdot \prod_{i=1}^{k} (t\gamma_i(p))^{\beta_i} \\ &= |a_\beta| \cdot (t\gamma_1(p))^{\beta^*} \\ &= |a_\beta| \cdot \left(\frac{t\delta_1\left(1 - \sqrt[k]{p}\right)}{\hat{\beta}}\right)^{\beta^*} \end{aligned}$$

where $\beta^* := \sum_{i=1}^{k} \beta_i$.
Step 5: To derive the bound on the precision, we consider the inverse of Formula (2.55), which is

$$\begin{aligned} S^{-1}_{\inf f}(\varphi(p)) &= \log_2 \frac{(d+1+\lceil \log_2 |\mathcal{I}| \rceil) \cdot |\mathcal{I}| \cdot \max |a_\iota| \cdot 2^{de_{\max}+1}}{\varphi(p)} \\ &= \log_2 \frac{(d+1+\lceil \log_2 |\mathcal{I}| \rceil) \cdot |\mathcal{I}| \cdot \max |a_\iota| \cdot 2^{de_{\max}+1} \cdot (2\hat{\beta})^{\beta^*}}{|a_\beta| \cdot (t\delta_1\left(1 - \sqrt[k]{p}\right))^{\beta^*}} \\ &= -\beta^* \log_2 (1 - \sqrt[k]{p}) \\ &\quad + \log_2 \frac{(d+1+\lceil \log_2 |\mathcal{I}| \rceil) \cdot |\mathcal{I}| \cdot \max |a_\iota| \cdot 2^{de_{\max}+1} \cdot (2\hat{\beta})^{\beta^*}}{|a_\beta| \cdot (t\delta_1)^{\beta^*}}. \end{aligned}$$

This way, we obtain the bound

$$L_{\text{safe}}(p) := \left\lceil S^{-1}_{\inf f}(\varphi(p)) \right\rceil.$$

We further obtain

$$L_{\text{grid}}(p) \;:=\; \left\lceil e_{\max} - 1 - \log_2\left((1-t) \cdot \frac{\delta_1\left(1 - \sqrt[k]{p}\right)}{\hat{\beta}}\right)\right\rceil.$$

if we replace the min expression in Formula (2.27) with the right side of Formula (2.40). □

Since the formula for $L_{\text{safe}}(p)$ in the lemma above seems rather complicated, we interpret it here. We exemplify the asymptotic behavior $L_{\text{safe}}(p) = O\left(-d \log(1 - \sqrt[k]{p})\right)$ for $p \to 1$ in the following corollary: We show that "slightly" more than d additional bits of the precision are sufficient to halve the failure probability.

Corollary 2.14. *Let f be a k-variate polynomial ($k \geq 2$) of total degree d and let $L_{\text{safe}} : (0,1) \to \mathbb{N}$ be the precision function in Formula (2.39). Then*

$$L_{\text{safe}}\left(\frac{1+p}{2}\right) \quad \leq \quad L_{\text{safe}}(p) + \lceil \lambda \beta^* \rceil$$

where $\beta^* = \sum_{i=1}^{k} \beta_i \leq d$ and

$$\lambda \quad := \quad \log_2 \left(\frac{1 - \sqrt[k]{p}}{1 - \sqrt[k]{\frac{1+p}{2}}} \right).$$

Proof. All quantities are as defined in Theorem 2.13. We obtain

$$\begin{aligned}
L_{\text{safe}}\left(\frac{1+p}{2}\right) \quad &= \quad \left\lceil -\beta^* \log_2 \left(1 - \sqrt[k]{\frac{1+p}{2}}\right) \; + \; c_{\text{m}}(\beta) \right\rceil \\
&= \quad \left\lceil -\beta^* \log_2 \left((1 - \sqrt[k]{p}) \cdot \frac{1 - \sqrt[k]{\frac{1+p}{2}}}{1 - \sqrt[k]{p}} \right) \; + \; c_{\text{m}}(\beta) \right\rceil \\
&= \quad \left\lceil -\beta^* \log_2 (1 - \sqrt[k]{p}) - \beta^* \log_2 \left(\frac{1 - \sqrt[k]{\frac{1+p}{2}}}{1 - \sqrt[k]{p}} \right) \; + \; c_{\text{m}}(\beta) \right\rceil \\
&\leq \quad L_{\text{safe}}(p) + \left\lceil \beta^* \log_2 \left(\frac{1 - \sqrt[k]{p}}{1 - \sqrt[k]{\frac{1+p}{2}}} \right) \right\rceil .
\end{aligned}$$

This proves the claim. □

2.8 The Top-down Approach Using Replacements

This approach derives the bounding functions associated with region- and value-suitability in the first stage of the analysis (see Figure 2.16). In the bottom-up approach, we consider a sequence of functions that is incrementally built-up from simple functions and *ends up* at the function f under consideration. In contrast to that, we now construct a sequence of functions top-down that *begins* with f and leads to a (different) sequence by dealing with the arguments of f coordinatewise. However, the top-down approach works in two phases: We only derive the auxiliary functions in the first phase, and we determine the bounds for the region- and value-suitability bottom-up in the second phase. This is why we also call this approach *pseudo-top-down*.

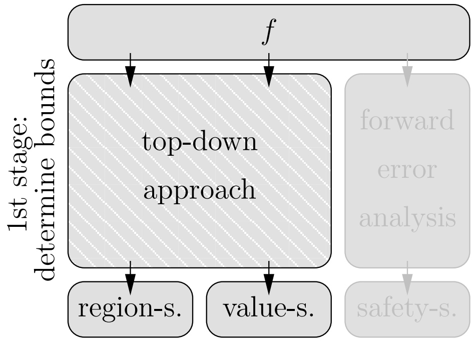


Figure 2.16: The top-down approach and its interface.

The idea of developing a top-down approach is not new and was first introduced by Mehlhorn et al. [59], followed by their journal article appeared in [60]. *As opposed to previous publications, our top-down approach is different for several reasons: It is designed to fit to the method of quantified relations, and it is based on our general conditions to analyze functions. (We do not need auxiliary constructions like exceptional points, continuity or a finite zero set.)*

New definitions are introduced in Section 2.8.1. We define the basic idea of a *replacement* in Section 2.8.2. Subsequently, we show how we can apply a *sequence of replacements* to the function under consideration in Section 2.8.3. We present the top-down approach to derive the bounding functions in Section 2.8.4 and consider an example in Section 2.8.5. Finally, for clarity, we answer selected questions in Section 2.8.6.

2.8.1 Definitions

We prepare the presentation with various definitions and begin with a projection. Let $\ell, k \in \mathbb{N}$ with $\ell \leq k$, let $I := \{1, \ldots, k\}$ and let

$$s : \{1, \ldots, \ell\} \to I$$

be an injective mapping. We then define the projection

$$\pi_s(x) \quad := \quad \big(x_{s(1)}, \ldots, x_{s(\ell)}\big)\,.$$

We extend the projection in a natural way to sets $X \subset \mathbb{R}^k$ by

$$\pi_s(X) \quad := \quad \{\pi_s(x) \,:\, x \in X\}\,.$$

Since we often make use of the projection π in the context of an index $i \in I$, we define the following abbreviations in their obvious meanings:

$$\begin{aligned}
\pi_i(x) &\;:=\; (x_i),\\
\pi_{<i}(x) &\;:=\; (x_1, \ldots, x_{i-1}),\\
\pi_{>i}(x) &\;:=\; (x_{i+1}, \ldots, x_k),\\
\pi_{\neq i}(x) &\;:=\; (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k).
\end{aligned}$$

We remark on these contextual definitions that the greatest index k is always given implicitly by the set I of indices. The usage of such orthogonal projections leads to the following condition on the set A of projected inputs: *It is a necessary condition in the top-down analysis that A as well as the perturbation area $\bar{U}_\delta(A)$ are closed axis-parallel boxes without holes.*

We briefly motivate the next notation: Assume that the function f has a k-ary argument. During the analysis of f, we often bind $k-1$ of these variables to values given in a $(k-1)$-tuple, say ξ. We do this to study the local behavior of f in dependence on a single free argument, say x_i.

Definition 2.17 (free-variable star)**.** Let $(f, k, A, \delta, e_{\max}, \Gamma\text{-box}, t)$ be a predicate description where A is an axis-parallel box without holes. In addition, let $I := \{1, \ldots, k\}$, and let $i \in I$. For each $(k-1)$-tuple $\xi := (\xi_1, \ldots, \xi_{i-1}, \xi_{i+1}, \ldots, \xi_k) \in \pi_{\neq i}(A)$, we define the function $f_\xi^{*i}(x_i)$ as

$$\begin{gathered}
f_\xi^{*i} : \pi_i(A) \to \mathbb{R},\\
x_i \;\mapsto\; f_\xi^{*i}(x_i) = f(x_1, \ldots, x_k)|_{x_j = \xi_j \;\forall j \in I,\, j \neq i} \;=\; f(\xi_1, \ldots, \xi_{i-1}, x_i, \xi_{i+1}, \ldots, \xi_k).
\end{gathered}$$

In other words, we consider f_ξ^{*i} as the function f where x_i is a free variable and all remaining variables are bound to the tuple ξ. We illustrate the definition with an example and consider the function $f(x_1, x_2, x_3) := 3x_1^2 + 2x_2^3 - 4x_3$. Then $f_{(4,7)}^{*2}$ is a function in x_2, and we have

$$\begin{aligned}
f_{(4,7)}^{*2}(x_2) &\;=\; f(x_1, x_2, x_3)|_{x_1=4 \,\wedge\, x_3=7}\\
&\;=\; 3 \cdot 4^2 + 2x_2^3 - 4 \cdot 7 = 2x_2^3 - 20.
\end{aligned}$$

We sometimes do not attach the tuple ξ to f^{*i} to relieve the reading if ξ is uniquely defined by the context.

Once we focus on the function f_ξ^{*i} in one variable, say x_i, we are interested in its induced critical set. This critical set surely depends on the choice of ξ. We have seen that the region-suitability is a necessary condition for the analyzability of the function. Therefore, the next definition is used to mark those ξ for which f_ξ^{*i} is or is not region-suitable.

Definition 2.18 (region-regularity)**.** Let $(f, k, A, \delta, e_{\max}, \Gamma\text{-box}, t)$ be a predicate description where A is an axis-parallel box without holes. We call $\xi \in \pi_{\neq i}(A)$ *region-regular* if f_ξ^{*i} is region-suitable on $\pi_i(A)$. Otherwise, we call ξ *non-region-regular.*

The region-suitability of f_ξ^{*i} implies that the functions $\nu_{f_\xi^{*i}}$ and $\chi_{f_\xi^{*i}}$ exist. If i is fixed, there are families of functions f_ξ^{*i} (and hence families of functions $\nu_{f_\xi^{*i}}$ and $\chi_{f_\xi^{*i}}$) that depend on the region-regular ξ. We examine these families in the next paragraph.

2.8.2 Single Replacement

We hereafter consider the following setting: *Let* $(f, k, A, \delta, e_{\max}, \Gamma\text{-box}, t)$ *be a predicate description where* A *is an axis-parallel box without holes, and let* $I := \{1, \ldots, k\}$*.* In addition, we denote the domain of f by $\mathrm{dom}(f)$.

We develop the top-down approach step-by-step. For a given index $i \in I$, our first aim is to lower-bound the absolute value of f by a function g, whose argument lists differ solely in the i-th position. While f depends on $x_i \in \pi_i(U_\delta(A))$, the function g depends on a new variable $\gamma_i \in \pi_i(\Gamma\text{-box})$. Hence, we say that the construction of g is motivated by the *replacement of* x_i *with* γ_i in the argument list of f.

We present the construction of the function g for a fixed index $i \in I$. We focus on the functions f_ξ^{*i} to study the local behavior of f in its i-th argument. We are interested in tuples $\xi \in \pi_{\neq i}(\mathrm{dom}(f))$ for which f_ξ^{*i} is region-suitable. We collect these points in the set

$$X_{f,i} \quad := \quad \{\xi \in \pi_{\neq i}(\mathrm{dom}(f)) : \xi \text{ is region-regular}\}\,.$$

To understand our interest in the set $X_{f,i}$, we remind ourselves of the following fact: For region-regular ξ, open neighborhoods of the critical set $C_{f_\xi^{*i}}$ are guaranteed to exist for any given (arbitrarily small) volume. This is not true for non-region-regular points which, therefore, must belong to the critical set of the objective function. We next define the objective function g. Let

$$g : \pi_{<i}(\mathrm{dom}(f)) \times \pi_i(\Gamma\text{-box}) \times \pi_{>i}(\mathrm{dom}(f)) \to \mathbb{R}_{\geq 0}$$

be the function with the pointwise definition

$$g(\xi_1, \ldots, \xi_{i-1}, \gamma_i, \xi_{i+1} \ldots, \xi_k) \quad := \quad \begin{cases} 0 & : \quad \xi \notin X_{f,i} \\ \inf\limits_{\text{(C1)}} \inf\limits_{\text{(C2)}} \left| f_\xi^{*i}(x_i) \right| & : \quad \xi \in X_{f,i} \end{cases} \tag{2.41}$$

$$\begin{array}{lcl} \text{(C1)} & : & \bar{x}_i \in \pi_i(A) \\ \text{(C2)} & : & x_i \in \bar{U}_{f^{*i},\delta_i}(\bar{x}_i) \setminus R_{f^{*i},\gamma_i}(\bar{x}_i) \end{array}$$

for all $\xi \in \pi_{\neq i}(\mathrm{dom}(f))$ and all $\gamma_i \in \pi_i(\Gamma\text{-box})$. The domains $\mathrm{dom}(f)$ and $\mathrm{dom}(g)$ only differ in the i-th coordinate. Whenever ξ is non-region-regular, we set g to zero. (This is essential for the sequence of replacements in Section 2.8.3 since this handling triggers the exclusion of an open neighborhood of ξ—and not just the exclusion of the point ξ itself.) In case ξ is region-regular, we set g to the infimum of the absolute value of f outside of the region of uncertainty for the various $\bar{x}_i$. Note that we must consider the infimum in the definition of g in Formula (2.41) because $|f_\xi^{*i}|$ does not need to have a minimum. We do not assume that f is continuous or semi-continuous.

Definition 2.19. We call the presented construction of the function g the *function resulting from the replacement of f 's argument x_i with γ_i*. We denote the replacement by $\mathrm{rep}(f, x_i \to \gamma_i)$.

We summarize the steps during the replacement of an argument of f and emphasize the relation between the quantities: Let f be given. We then begin with the consideration of the auxiliary function f_ξ^{*i}. We use it to determine the auxiliary set of region-regular points $X_{f,i}$. To determine the function g afterwards, we again examine f_ξ^{*i}, but now only for the points in $X_{f,i}$.

In the proof of the analysis in Section 2.8.4, we use the statement that the replacement $\mathrm{rep}(f, x_i \to \gamma_i)$ results in a positive function that lower bounds the absolute value of f in a certain sense. We formalize and prove this statement in the next lemma.

Lemma 2.15. *Let $(f, k, A, \delta, e_{\max}, \Gamma\text{-box}, t)$ be a predicate description where A is an axis-parallel box without holes, let $I := \{1, \dots, k\}$, and let $i \in I$. Moreover, let $g := \mathrm{rep}(f, x_i \to \gamma_i)$. We then have*

$$|f(\xi_1, \dots, \xi_{i-1}, x_i, \xi_{i+1} \dots, \xi_k)| \geq g(\xi_1, \dots, \xi_{i-1}, \gamma_i, \xi_{i+1} \dots, \xi_k) > 0 \qquad (2.42)$$

for all region-regular points $\xi \in X_{f,i}$, for all $\gamma_i \in \pi_i(\Gamma\text{-box})$, for all $\bar{x}_i \in \pi_i(A)$, and for all $x_i \in \bar{U}_{f^{*i},\delta_i}(\bar{x}_i) \setminus R_{f^{*i},\gamma_i}(\bar{x}_i)$.

Proof. The left unequation in Formula (2.42) follows immediately from the construction of the function $g = \mathrm{rep}(f, x_i \to \gamma_i)$ because we only consider points lying outside of the region of uncertainty $R_{f^{*i},\gamma_i}(\bar{x}_i)$.

To prove the right unequation in Formula (2.42), we assume that there is a region-regular $\xi \in X_{f,i}$ and $\gamma_i \in \pi_i(\Gamma\text{-box})$ such that $g(\xi_1, \dots, \xi_{i-1}, \gamma_i, \xi_{i+1} \dots, \xi_k) = 0$. This implies that, for $\bar{x}_i \in \pi_i(A)$, there must be a sequence $(a_j)_{j\in\mathbb{N}}$ in the area $\bar{U}_{f^{*i},\delta_i}(\bar{x}_i) \setminus R_{f^{*i},\gamma_i}(\bar{x}_i)$ for which $\lim_{j\to\infty} f_\xi^{*i}(a_j) = 0$. Consequently, $a := \lim_{j\to\infty} a_j$ must belong to the critical set. Since the region of uncertainty R_{f^{*i},γ_i} guarantees the exclusion of the open γ_i-neighborhood of the critical set (which includes the open γ_i-neighborhood of a), almost all points of the sequence $(a_j)_{j\in\mathbb{N}}$ must also lie in R_{f^{*i},γ_i}. This leads to a contradiction to the assumption and proves the claim. □

We add that the right unequation in Formula (2.42) presumes that ξ is region-regular as is stated in the lemma. We obtain $g \equiv 0$ if $X_{f,i}$ is the empty set. We continue with a simple example that illustrates the method to determine $\mathrm{rep}(f, x_i \to \gamma_i)$.

Example 2.9. Let $f(x_1, x_2) = x_1^2 + x_2^2$. Then $I = \{1, 2\}$. In addition, let $i = 2$, and let A be an axis-parallel rectangle that contains the origin $(0, 0)$. We consider $f_{\xi_1}^{*2}(x_2) = \xi_1^2 + x_2^2$. Since f^{*2} is region-suitable, this leads to $X_{f,2} = \pi_{\neq 2}(A) = \pi_1(A)$. We obtain

$$g(\xi_1, \gamma_2) \quad := \quad \begin{cases} \gamma_2^2 & : \quad \xi_1 = 0 \\ \xi_1^2 & : \quad \text{otherwise.} \end{cases}$$

The critical set of g contains a single point in the case $\xi_1 = 0$ and is empty in the other case. ○

We end this subsection with two observations. First, although $g(\xi_1, \gamma_2) > 0$ in the example above, the limit

$$\inf_{\xi_1 \in X_{f,2} \wedge \xi_1 \neq 0} g(\xi_1, \gamma_2) \quad = \quad 0.$$

Second, if the lower-bounding function g is region-value-suitable, the function f is also region-value-suitable because of Theorem 2.9. This observation is the driving force of the top-down approach.

2.8.3 Sequence of Replacements

We know so far how a variable x_i of the argument list of the function f under consideration can be replaced with a new variable γ_i. The advantage of the new variable γ_i is that it reflects the distance to the critical set, somehow. We announce that, opposed to x_i, the variable γ_i is appropriate for the analysis. A benefit of γ_i is that it is not necessary to study the precise location of the critical set; the knowledge about the "width" of the critical set is sufficient.

The idea behind the top-down approach is to apply the replacement procedure k times in a row to replace all original arguments $(x_1, \ldots, x_k)$ of f by the new substitutes $(\gamma_1, \ldots, \gamma_k) \in \Gamma$-box. To keep the presentation as general as possible, we maintain the order of the k replacements variable. Let $\sigma : I \to I$ be a bijective function that defines the order in which we replace the arguments of f. We interpret $\sigma(i) = j$ as the replacement of x_j with γ_j in the i-th step.

We now look for a recursive definition to derive the sequence $g_1, \ldots, g_k$ of functions that result from these replacements. We define the basis of the recursion as $g_0 := f$ with $g_0 : \bar{U}_\delta(A) \to \mathbb{R}$ and $\text{dom}(g_0) = \bar{U}_\delta(A)$. We set $g_i := \text{rep}(g_{i-1}, x_{\sigma(i)} \to \gamma_{\sigma(i)})$ for $i \in I$. In other words, we focus on the replacement of $x_{\sigma(i)}$ in step $i \in I$, which means that we assume that we have just derived the functions $g_1, \ldots, g_{i-1}$. We then determine the set of region-regular points

$$X_{g_{i-1},\sigma(i)} \quad := \quad \big\{\xi \in \pi_{\neq\sigma(i)}(\text{dom}(g_{i-1})) : \xi \text{ is region-regular}\big\},$$

so we check if the function

$$g_{i-1,\xi}^{*\sigma(i)} : \pi_{\sigma(i)}(\text{dom}(g_{i-1})) \to \mathbb{R}_{\geq 0},$$
$$g_{i-1,\xi}^{*\sigma(i)}\big(x_{\sigma(i)}\big) \mapsto g_{i-1}\big(\xi_1, \ldots, \xi_{\sigma(i)-1}, x_{\sigma(i)}, \xi_{\sigma(i)+1}, \ldots, \xi_k\big)$$

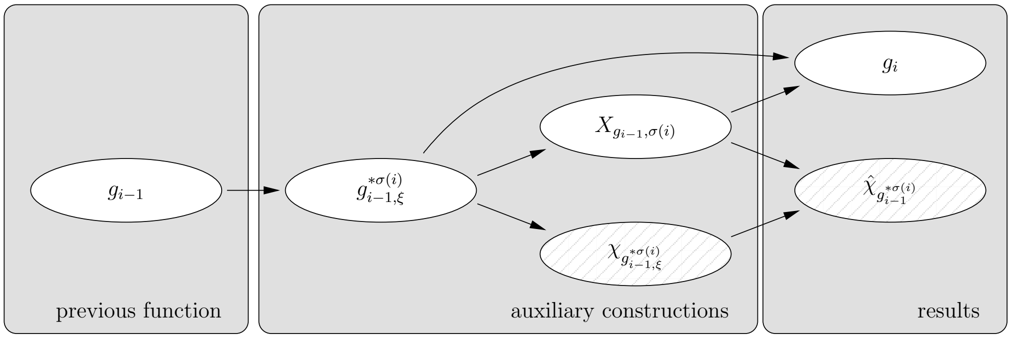

g_i
$X_{g_{i-1},\sigma(i)}$
g_{i-1}
$g^{*\sigma(i)}_{i-1,\xi}$
$\hat{\chi}_{g^{*\sigma(i)}_{i-1}}$
$\chi_{g^{*\sigma(i)}_{i-1,\xi}}$
previous function
auxiliary constructions
results

Figure 2.17: Illustration of the dependencies during the i-th replacement. The white-colored quantities are defined in Section 2.8.3 and the striped quantities in Section 2.8.4. Here, "$A \to B$" means that B is derived from A.

is region-suitable for a given ξ. Thereafter, we define the domain of the succeeding function g_i as

$$g_i : \pi_{<\sigma(i)}(\mathrm{dom}(g_{i-1})) \times \pi_{\sigma(i)}(\Gamma\text{-box}) \times \pi_{>\sigma(i)}(\mathrm{dom}(g_{i-1})) \to \mathbb{R}_{\geq 0}$$

and use $X_{g_{i-1},\sigma(i)}$ to define $g_i(\xi_1, \ldots, \xi_{\sigma(i)-1}, \gamma_{\sigma(i)}, \xi_{\sigma(i)+1} \ldots, \xi_k)$

$$:= \begin{cases} 0 & : \ \xi \notin X_{g_{i-1},\sigma(i)} \\ \inf\limits_{\text{(C1)}} \inf\limits_{\text{(C2)}} \left| g^{*\sigma(i)}_{i-1}(x_{\sigma(i)}) \right| & : \ \xi \in X_{g_{i-1},\sigma(i)} \end{cases} \tag{2.43}$$

$$\begin{array}{lcl} \text{(C1)} & : & \bar{x}_{\sigma(i)} \in \pi_{\sigma(i)}(\mathrm{dom}(g_{i-1})) \\ \text{(C2)} & : & x_{\sigma(i)} \in \bar{U}_{g^{*\sigma(i)}_{i-1},\delta_{\sigma(i)}}(\bar{x}_{\sigma(i)}) \setminus R_{g^{*\sigma(i)}_{i-1},\gamma_{\sigma(i)}}(\bar{x}_{\sigma(i)}) \end{array}$$

for all $\xi \in \pi_{\neq\sigma(i)}(\mathrm{dom}(g_{i-1}))$ and all $\gamma_{\sigma(i)} \in \pi_{\sigma(i)}(\Gamma\text{-box})$. We summarize the relation between the quantities during the i-th replacement in Figure 2.17. (The striped quantities are introduced later.)

The definitions above are chosen such that the function g_i exists. After the k-th step, the recursion ends with $g_k : \Gamma\text{-box} \to \mathbb{R}_{\geq 0}$. We remark that if we apply this mechanism to functions that are not admissible for controlled perturbation, the sequence of replacements will end up with a function g_k that fails the analysis from the next section.

Example 2.10. We return to the 2-dimensional in_box-predicate. It is sufficient for this example to assume that the box is fixed somehow and that the only argument of the predicate is the query point $q = (x_1, x_2)$. We this time consider the various domains and critical sets of the functions g_i that result from the sequence of replacements. (The order of the replacements is not important for this example.) The situation is illustrated in Figure 2.18. Picture (a) shows the domain (shaded region) of the function $f = g_0$. We know that the critical set is the boundary of the query box.

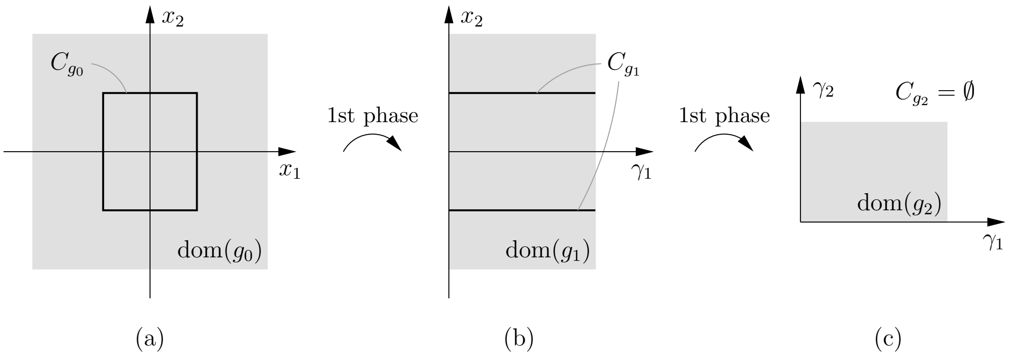


Figure 2.18: Illustration of the various domains and critical sets that result from the sequence of replacements for the 2-dimensional in_box-predicate.

After the replacement $\mathrm{rep}(g_0, x_1 \to \gamma_1)$, the first argument belongs to the set $\pi_1(\Gamma\text{-box})$ resulting in an altered domain (see Picture (b)). We make two observations. First, the critical set of g_1 is formed by two horizontal lines that are caused by the top and bottom part of the box C_{g_0}. What is the reason for this? If we consider the absolute value of g_0 while moving its argument along a horizontal line that passes through the top or bottom line segment of the box (x_2 is then fixed), it leads to a mapping that is zero on an open interval; in this case, the mapping cannot be region-suitable. Second, there are no further contributions to the critical set of g_1. What is the reason? If we consider the absolute value of g_0 along a horizontal line that passes through the interior of the box, it leads to a mapping that is region-suitable.

Picture (c) shows the situation after the second replacement $\mathrm{rep}(g_1, x_2 \to \gamma_2)$. The function g_2 is positive on its entire domain Γ-box. The reason for this is that if we consider the absolute value of g_1 along a vertical line (γ_1 is then fixed), it leads to a mapping that is region-suitable. ○

2.8.4 Derivation and Correctness of the Bounds

Although we have replaced each x_i with γ_i in the argument list of f in a top-down manner, we are not able to determine the bounds ν_f and φ_f in the same way. To achieve this goal, we need to go through the collected information bottom-up again. The reason is that at the time we arrive at a function, say g_{i-1}, we cannot check directly if g_{i-1} is region- and value-suitable. Instead, we want the predecessor to inherit these properties from the successor g_i. We will see that, once we arrive at g_k, we can easily check if g_k has the desired properties. This way, we can possibly show that g_0 (i.e., f) is also region-value-suitable.

Therefore, we divide the analysis in two phases. The first phase consists of the deduction of g_k via the sequence of replacements and was already presented in the last section. The second phase consists of the deduction of the bounding functions

φ_f and χ_f and is the subject of this section. We begin with an auxiliary statement which claims that g_k is non-decreasing in each argument under certain circumstances.

Lemma 2.16. *Let $(f, k, A, \delta, e_{\max}, \Gamma\text{-box}, t)$ be a predicate description where A is an axis-parallel box without holes, and let $I := \{1, \ldots, k\}$. Let $\sigma : I \to I$ be bijective, i.e., an order on the elements of I. Finally, let $g_0 := f$ and $g_j := \operatorname{rep}(g_{j-1}, x_{\sigma(j)} \to \gamma_{\sigma(j)})$ for all $1 \leq j \leq k$, i.e., g_k is the resulting function after the k replacements. If the function g_k is positive*[22]*, it is non-decreasing in γ_i on $\pi_i(\Gamma\text{-box})$ for all $i \in I$.*

Proof. We refer to the explicit definition of g_i in Formula (2.43) that reflects the replacement of the i-th argument: For growing $\gamma_{\sigma(i)}$, we shrink the domain for $x_{\sigma(i)}$ due to condition (C2). Formally, for $\gamma', \gamma'' \in \pi_{\sigma(i)}(\Gamma\text{-box})$ with $\gamma' < \gamma''$, the corresponding regions of uncertainty are related in the way

$$R_{g_{i-1}^{*\sigma(i)}, \gamma'}(\bar{x}_{\sigma(i)}) \;\subset\; R_{g_{i-1}^{*\sigma(i)}, \gamma''}(\bar{x}_{\sigma(i)}).$$

Because the function value of g_i is defined by the infimum absolute value, the function g_i must be non-decreasing in its i-th argument $\gamma_{\sigma(i)}$ for region-regular ξ by construction.

The same argumentation is true for each of the k replacements and is independent of the actual sequence of replacements. This finishes the proof. □

The domain of the function g_k is naturally Γ-box. Even if Γ-box has the same cardinality than $\mathbb{R}$ for $k \geq 2$, it is non-obvious how to define an invertible function χ_{g_k} on Γ-box. Please note that such a bound is required to use the method of quantified relations. For this purpose, we restrict the domain in the analysis to Γ-line: It is true that the elements of $\gamma \in \Gamma$-line are now interlinked, but the important fact is that we can still choose them arbitrarily close to zero.

To further prepare the analysis, we have to focus on a peculiarity of the auxiliary function $g_{i-1,\xi}^{*\sigma(i)}$ for a given $i \in I$. Remember that $\nu_{g_{i-1,\xi}^{*\sigma(i)}}$ and $\chi_{g_{i-1,\xi}^{*\sigma(i)}}$ are families of functions with parameter $\xi \in X_{g_{i-1},\sigma(i)}$. Therefore, we are facing the following issue: For a given $i \in I$, how can we deal with these two families of functions? The first solution that occurs to us is to replace each family with just one bounding function—so this is what we do. We define the pointwise limits of these families as

$$\hat{\nu}_{g_{i-1}^{*\sigma(i)}}\left(\gamma_{\sigma(i)}\right) \quad := \quad \sup_{\xi \in X_{f,i}} \; \nu_{g_{i-1,\xi}^{*\sigma(i)}}\left(\gamma_{\sigma(i)}\right)$$

and

$$\hat{\chi}_{g_{i-1}^{*\sigma(i)}}\left(\gamma_{\sigma(i)}\right) \quad := \quad \inf_{\xi \in X_{f,i}} \; \chi_{g_{i-1,\xi}^{*\sigma(i)}}\left(\gamma_{\sigma(i)}\right) \tag{2.44}$$

for $\gamma \in \Gamma$-box and make use of these new bounds in the analysis. To illustrate this part in the analysis, we have added the two striped quantities in Figure 2.17.

We are now ready to present the top-down approach to analyze real-valued functions. We claim and prove the results in the following theorem.

[22] This is why we have defined Γ-box as an *open* set.

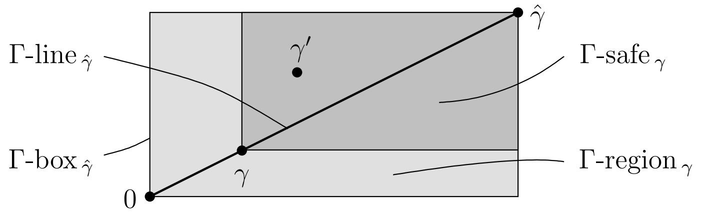

$\hat{\gamma}$
$\Gamma\text{-line}_{\hat{\gamma}}$
γ'
$\Gamma\text{-safe}_{\gamma}$
$\Gamma\text{-box}_{\hat{\gamma}}$
$\Gamma\text{-region}_{\gamma}$
γ
0

Figure 2.19: This is an exemplified 2-dimensional illustration of the decomposition of the Γ-box into the sets $\Gamma\text{-safe}_{\gamma}$ and $\Gamma\text{-region}_{\gamma}$ for $\gamma \in \Gamma\text{-line}$.

Theorem 2.17 (top-down approach). *Let $(f, k, A, \delta, e_{\max}, \Gamma\text{-box}, t)$ be a predicate description where A is an axis-parallel box without holes, and let $I := \{1, \dots, k\}$. Let $\sigma : I \to I$ be bijective, i.e., an order on the elements of I. Finally, let $g_0 := f$ and $g_j := \operatorname{rep}(g_{j-1}, x_{\sigma(j)} \to \gamma_{\sigma(j)})$ for all $1 \leq j \leq k$. We define φ_f and χ_f as*

$$\begin{aligned} \varphi_f(\gamma) &:= g_k(\gamma) \\ \chi_f(\gamma) &:= \prod_{j=1}^{k} \hat{\chi}_{g_{j-1}^{*\sigma(j)}}\big(\gamma_{\sigma(j)}\big) . \end{aligned}$$

If g_k is positive on Γ-box *and χ_f is invertible on*[23] Γ-line*, then f is region-value-suitable with the bounding functions*[24] *φ_f and χ_f.*

Proof. We prove the claim in three parts. First, we show that there are certain bounding functions φ_{g_k} and χ_{g_k} for which g_k is region-value-suitable. Second, we prove that if the function g_i has such bounding functions, then g_{i-1} also has appropriate bounding functions. Finally, we deduce the claim of the theorem.

Part 1 (basis). We assume that g_k is positive on the open set Γ-box, that means, we consider the function $g_k : \Gamma\text{-box} \to \mathbb{R}_{>0}$. To begin with, we decompose the domain in two parts (see Figure 2.19). Let $\gamma \in \Gamma\text{-line}$. We define the unique open axis-parallel box with opposite vertices γ and[25] $\hat{\gamma}$ as

$$\Gamma\text{-safe}_{\gamma} \quad := \quad \big\{\gamma' \in \Gamma\text{-box} : \gamma_i \leq \gamma'_i \text{ for all } i \in I\big\}.$$

We denote its complement within the Γ-box by

$$\Gamma\text{-region}_{\gamma} \quad := \quad \Gamma\text{-box} \setminus \Gamma\text{-safe}_{\gamma}.$$

We think of $\Gamma\text{-region}_{\gamma}$ as the region of uncertainty and $\Gamma\text{-safe}_{\gamma}$ as the region whose floating-point numbers are guaranteed to evaluate fp-safe. We claim that g_k is region-

[23] We know that Γ-line ⊂ Γ-box.

[24] We can use ν_f instead of χ_f because of Formula (2.17).

[25] We have introduced $\hat{\gamma}$ to define $\Gamma\text{-box}_{\hat{\gamma}}$ and $\Gamma\text{-line}_{\hat{\gamma}}$. More information and the formal bound is given in Remark 2.2.2 on Page 38.

value-suitable on Γ-box in the following sense: We set the bounding functions to

$$\begin{aligned}\varphi_{g_k}(\gamma) &:= g_k(\gamma) \\ \chi_{g_k}(\gamma) &:= \prod_{j=1}^{k} (\hat{\gamma}_j - \gamma_j)\end{aligned}$$

and claim that two statements are fulfilled for every $\gamma \in \Gamma$-line. They are:

1. The absolute value of $g_k(\gamma')$ is at least $\varphi_{g_k}(\gamma)$ for all points $\gamma' \in \Gamma\text{-safe}_{\gamma}$.

2. The volume of $\Gamma\text{-safe}_{\gamma}$ is $\chi_{g_k}(\gamma)$.

To prove the first statement, we consider the function value of g_k along a path of k axis-parallel line segments from γ to γ'. The path starts at $\gamma = (\gamma_1, \ldots, \gamma_k)$, connects the $(k-1)$ points $(\gamma'_1, \ldots, \gamma'_j, \gamma_{j+1} \ldots, \gamma_k)$ with $1 \leq j < k$ in ascending order of j and ends at $\gamma' = (\gamma'_1, \ldots, \gamma'_k)$. Along this path, the function value of g_k is non-decreasing because of Lemma 2.16: For all $i \in I$, the function g_k is non-decreasing in its i-th argument $\gamma_i \in \pi_i(\Gamma\text{-box})$ for fixed $\xi \in \pi_{\neq i}(\Gamma\text{-box})$.

The proof of the second statement is straight forward: Because the box is axis-parallel, its volume is the product of its edge-lengths. We make the observation that the function $\chi_{g_k}(\gamma)$ is strictly monotonically increasing on Γ-line and hence must be invertible on this domain.

We conclude the first part of the proof: *For a given* $\gamma \in \Gamma$-line*, we have shown that the function value of* g_k *is at least* $\varphi_{g_k}(\gamma)$ *on an area of volume* $\chi_{g_k}(\gamma)$*.* This way we have found evidence that g_k is region-value-suitable in the meaning above.

Part 2 (induction). We claim: *For* $i \in I$ *and* $\gamma \in \Gamma$-line*, the function value of* g_{i-1} *is at least* $\varphi_{g_{i-1}}(\gamma)$ *on an area of volume* $\chi_{g_{i-1}}(\gamma)$ *with*

$$\begin{aligned}\varphi_{g_{i-1}}(\gamma) &:= \varphi_{g_i}(\gamma) && (2.45) \\ \chi_{g_{i-1}}(\gamma) &:= \chi_{g_i}(\gamma) \cdot \frac{\hat{\chi}_{g_{i-1}^{*\sigma(i)}}\left(\gamma_{\sigma(i)}\right)}{\hat{\gamma}_{\sigma(i)} - \gamma_{\sigma(i)}}. && (2.46)\end{aligned}$$

We prove the claim by mathematical induction for descending $i \in I$. Basis $(i = k)$. Due to the first part, we can base the proof on the bounding functions φ_{g_k} and χ_{g_k}. Induction step $(i \in I)$. We assume that the bounding functions are true for all $j \in I$ with $i \leq j \leq k$ and prove the claim for $i - 1$. We do this next.

Remember the definition $g_i := \text{rep}\big(g_{i-1}, x_{\sigma(i)} \to \gamma_{\sigma(i)}\big)$. In the step backwards from g_i to g_{i-1}, we observe the following difference in their two axis-parallel domains due to condition (C2) of Formula (2.43): The counterpart to the situation in which the $\sigma(i)$-th argument of g_i lies in $\pi_{\sigma(i)}\left(\Gamma\text{-safe}_{\gamma}\right)$ is the situation in which the $\sigma(i)$-th argument of g_{i-1} lies in

$$\bar{U}_{g_{i-1,\delta_{\sigma(i)}}^{*\sigma(i)}}\left(\bar{x}_{\sigma(i)}\right) \setminus R_{g_{i-1,\gamma_{\sigma(i)}}^{*\sigma(i)}}\left(\bar{x}_{\sigma(i)}\right) \tag{2.47}$$

and belongs to the region-regular case. Furthermore, the volume of this area is guaranteed to be at least $\hat{\chi}_{g_{i-1}^{*\sigma(i)}}\left(\gamma_{\sigma(i)}\right)$ due to Formula (2.44). Because the axis-parallel domains of g_i and g_{i-1} do not differ in directions different to the $\sigma(i)$-th main axis, their volumes (which are the product of edge lengths) solely differ in a factor. Therefore, we can estimate the volume $\chi_{g_{i-1}}(\gamma)$ at the product $\chi_{g_i}(\gamma)$ where we replace the factor $\left(\hat{\gamma}_{\sigma(i)} - \gamma_{\sigma(i)}\right)$ by $\hat{\chi}_{g_{i-1}^{*\sigma(i)}}(\gamma_{\sigma(i)})$; this validates Formula (2.46).

Because of Lemma 2.15, the lower-bounding function φ_{g_i} is also a lower-bounding function on the volume of the area, which is defined in Formula (2.47). This validates Formula (2.45).

Part 3 (conclusion). We have shown so far that *for a given* $\gamma \in \Gamma$-line*, the function value of* $f = g_0$ *is at least* $\varphi_f(\gamma)$ *on an area of volume* $\chi_f(\gamma)$ *because*

$$\varphi_f(\gamma) \;=\; \varphi_{g_0}(\gamma) \;=\; \varphi_{g_1}(\gamma) \;=\; \cdots \;=\; \varphi_{g_k}(\gamma) \;=\; g_k(\gamma)$$

and because

$$\begin{aligned}
\chi_f(\gamma) &= \chi_{g_0}(\gamma) \\
&= \chi_{g_1}(\gamma) \cdot \frac{\hat{\chi}_{g_0^{*\sigma(1)}}\left(\gamma_{\sigma(1)}\right)}{\hat{\gamma}_{\sigma(1)} - \gamma_{\sigma(1)}} \\
&= \chi_{g_2}(\gamma) \cdot \frac{\hat{\chi}_{g_1^{*\sigma(2)}}\left(\gamma_{\sigma(2)}\right)}{\hat{\gamma}_{\sigma(2)} - \gamma_{\sigma(2)}} \cdot \frac{\hat{\chi}_{g_0^{*\sigma(1)}}\left(\gamma_{\sigma(1)}\right)}{\hat{\gamma}_{\sigma(1)} - \gamma_{\sigma(1)}} \\
&\;\vdots \\
&= \chi_{g_k}(\gamma) \cdot \prod_{i=1}^{k} \frac{\hat{\chi}_{g_{i-1}^{*\sigma(i)}}\left(\gamma_{\sigma(i)}\right)}{\hat{\gamma}_{\sigma(i)} - \gamma_{\sigma(i)}} \\
&= \prod_{j=1}^{k} \left(\hat{\gamma}_j - \gamma_j\right) \cdot \prod_{i=1}^{k} \frac{\hat{\chi}_{g_{i-1}^{*\sigma(i)}}\left(\gamma_{\sigma(i)}\right)}{\hat{\gamma}_{\sigma(i)} - \gamma_{\sigma(i)}} \\
&= \prod_{i=1}^{k} \left(\hat{\gamma}_{\sigma(i)} - \gamma_{\sigma(i)}\right) \cdot \prod_{i=1}^{k} \frac{\hat{\chi}_{g_{i-1}^{*\sigma(i)}}\left(\gamma_{\sigma(i)}\right)}{\hat{\gamma}_{\sigma(i)} - \gamma_{\sigma(i)}} \\
&= \prod_{i=1}^{k} \hat{\chi}_{g_{i-1}^{*\sigma(i)}}\left(\gamma_{\sigma(i)}\right).
\end{aligned}$$

If χ_f is in addition invertible on the domain Γ-line, f is region-value-suitable. This finishes the proof. □

One prerequisite in the last theorem is that g_k is positive on the open Γ-box. We make the observation that we cannot validate this property unless we have determined the entire sequence of replacements from $f = g_0$ down to g_k. So, it is possible that the analysis fails at the end of the first phase.

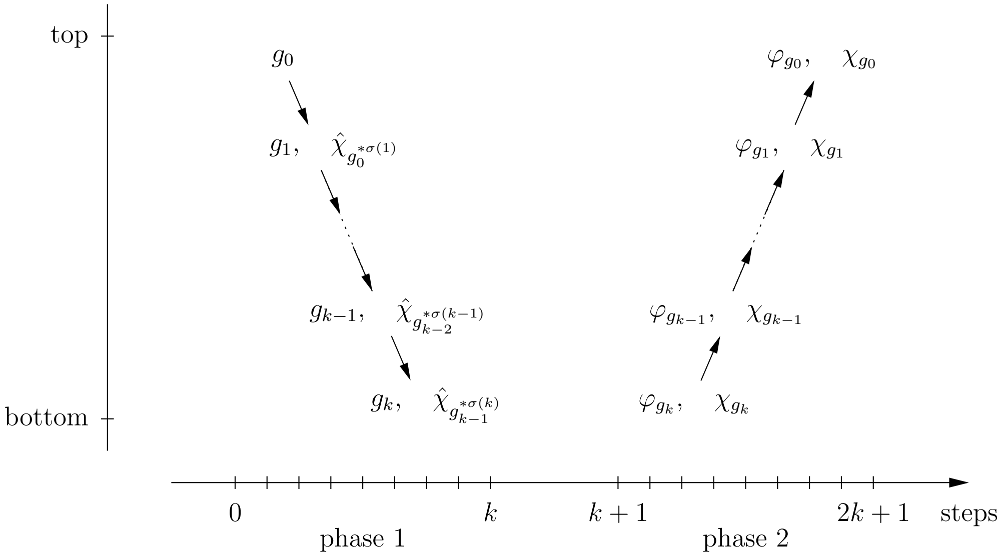

top
bottom
g_0
$g_1, \quad \hat{\chi}_{g_0^{*\sigma(1)}}$
$g_{k-1}, \quad \hat{\chi}_{g_{k-2}^{*\sigma(k-1)}}$
$g_k, \quad \hat{\chi}_{g_{k-1}^{*\sigma(k)}}$
$\varphi_{g_0}, \quad \chi_{g_0}$
$\varphi_{g_1}, \quad \chi_{g_1}$
$\varphi_{g_{k-1}}, \quad \chi_{g_{k-1}}$
$\varphi_{g_k}, \quad \chi_{g_k}$
0
k
$k+1$
$2k+1$
steps
phase 1
phase 2

Figure 2.20: Instructions for performing the top-down approach. This illustration reflects the steps in which the quantities are determined according to Theorem 2.17.

Furthermore, we make the observation that the bounding functions φ_f and χ_f are actually derived bottom-up in the the second phase of their derivation. That means, although we technically determine the sequence of functions g_i in a top-down manner on the surface, the validity of the formulas is derived bottom-up afterwards. We summarize the steps of the top-down approach in Figure 2.20.

2.8.5 Examples

Example 2.11. We use the top-down approach to determine the bounding functions $\varphi_{\text{in\_box}}$ and $\chi_{\text{in\_box}}$ for the predicate in_box. We assume that the box is fixed somehow and that the only argument for the predicate is the query point. (There is not much influence on the analysis by the remaining parameters.) The predicate can be realized, for example, by the function

$$f(x) \quad := \quad \min_{1 \leq i \leq k} \left\{ \ell_i^2 - (x_i - c_i)^2 \right\}$$

where $c \in \mathbb{R}^k$ is the center of the axis-parallel box, and its edge lengths are given by 2ℓ. We eliminate the variables in ascending order from x_1 to x_k, that means, we set $\sigma(i) := i$ for all $1 \leq i \leq k$.

Part 1 ($\varphi_{\text{in\_box}}$). To determine $\varphi_{\text{in\_box}}$, we need g_k; to determine g_k, we need the entire sequence of replacements; and, to determine g_i, we need to determine the value of the "inf inf" expression in dependence on γ_i in Formula (2.43). We do this next. Because of the symmetry of f, the following discussion is valid for all coordinates x_i.

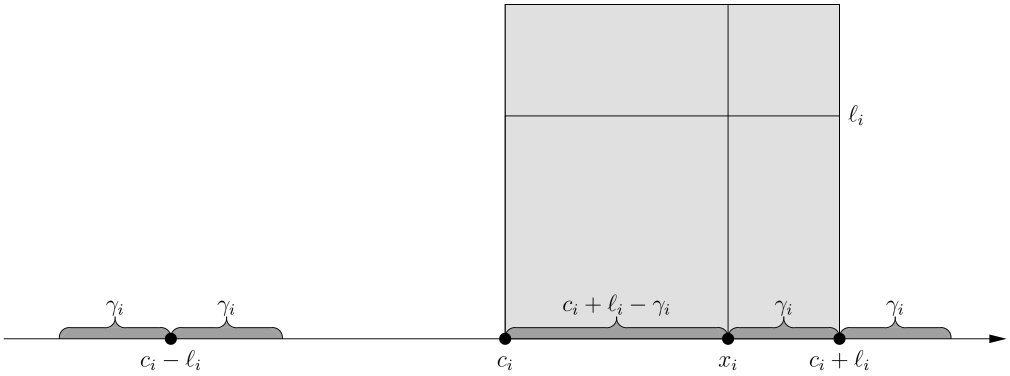


Figure 2.21: An illustration that supports the relation between the quantities of f^{*i} for the region-regular case of the predicate in_box.

To prepare the replacement of variables, we examine the function f^{*i} for the region-regular case (see Figure 2.21). The critical set $C_{f^{*i}}$ contains two points, namely, $c_i - \ell_i$ and $c_i + \ell_i$. We denote by γ_i the minimal distance of x_i to a point in $C_{f^{*i}}$. (We assume that $\hat{\gamma}_i$ must be less than ℓ_i; otherwise, the interior of the box would be covered entirely by the region of uncertainty and the predicate would lose its meaning.) The absolute value of f grows in the distance to $C_{f^{*i}}$. To determine a guaranteed lower bound on the absolute value of f, we assume that the distance of x_i to $C_{f^{*i}}$ is exactly γ_i. In addition, we make the observation that $|f|$ grows slower towards the interior of the box than away from the box; therefore, we must also assume that x_i lies between $c_i - \ell_i$ and $c_i + \ell_i$ to obtain a convincing bound. This leads to the worst-case consideration $|x_i - c_i| = |c_i + \ell_i - \gamma_i|$. We make use of the binomial theorem to derive the unequation

$$\begin{aligned} \left|\ell_i^2 - (x_i - c_i)^2\right| &\geq \left|\ell_i^2 - (c_i + \ell_i - \gamma_i)^2\right| \\ &= \left|2\ell_i\gamma_i - \gamma_i^2\right| \\ &= \left|(2\ell_i - \gamma_i)\,\gamma_i\right|. \end{aligned}$$

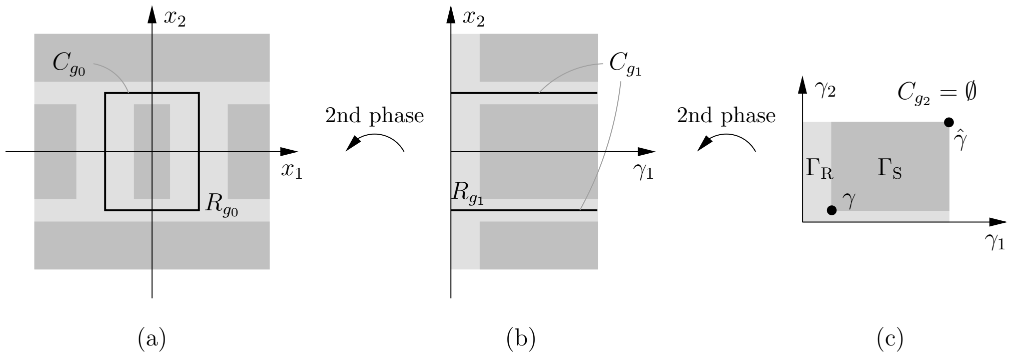
We next define the functions g_i as

$$\begin{aligned} g_i(\gamma_1, \ldots, \gamma_i, x_{i+1}, \ldots, x_k) \;:=\; \min\Big(&\left\{(2\ell_j - \gamma_j)\gamma_j : 1 \leq j \leq i\right\} \\ &\cup \left\{\ell_j^2 - (x_j - c_j)^2 : i < j \leq k\right\}\Big), \end{aligned}$$

and in the end, the sequence of replacements leads to

$$\begin{aligned} \varphi_{\text{in\_box}}(\gamma) &:= g_k(\gamma) \\ &= \min_{1 \leq j \leq k} (2\ell_j - \gamma_j)\,\gamma_j. \end{aligned}$$

Part 2 ($\chi_{\text{in\_box}}$). We now determine a bound on the volume of the complement of



(a) (b) (c)

Figure 2.22: Illustration of the regions of uncertainty for the various domains in the analysis of the 2-dimensional in_box-predicate.

the region of uncertainty. For every $i \in I$, a valid bounding function is given by

$$\hat{\chi}_{g^{*i}_{i-1}}(\gamma_i) \quad = \quad 2\delta_i - 4\gamma_i.$$

This results in the following bound on the total volume:

$$\begin{aligned}\chi_{\mathrm{in\_box}}(\gamma) \quad &= \quad \prod_{i=1}^{k} \hat{\chi}_{g^{*i}_{i-1}}(\gamma_i) \\ &= \quad \prod_{i=1}^{k} (2\delta_i - 4\gamma_i)\,.\end{aligned}$$

Once we have determined the bounding functions $\varphi_{\mathrm{in\_box}}$ and $\chi_{\mathrm{in\_box}}$, it would be possible to finish the analysis with the method of quantified relations—but, this is not the goal of this section. ○

Example 2.12. This is the continuation of Examples 2.10 and 2.11. Below, we want to investigate the regions of uncertainty for the various functions g_i. More precisely, we are interested in the correlation between the regions that are defined bottom-up in the second phase of the approach.

Figure 2.22 visualizes the regions of uncertainty for the functions g_i. The regions of uncertainty are light-shaded, whereas their complements are dark-shaded. The decomposition is initiated by the choice of $\gamma \in \Gamma$-box. Since each component γ_i is positive, neighborhoods of the critical set are added to the region of uncertainty on the way back up to g_0.

The upper line segment of C_{g_0} causes the upper line of C_{g_1} as we have seen in Example 2.10. Conversely, we can now see that the upper line of C_{g_1} causes a region of uncertainty around the *line that passes through* the upper line segment of C_{g_0}. Be aware that our top-down approach is designed such that this behavior is forced for all non-region-regular situations. This implies that our method does not need any

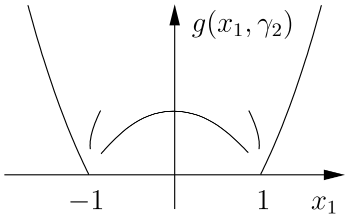


Figure 2.23: Exemplified drawing of the "in unit circle" predicate after the first replacement. The function values on the interval $[-1, 1]$ vary with γ_2.

kind of exceptional sets. On the contrary, there are no restrictions on the measure of the critical sets at all; the only thing that matters is the criterion whether f is region-suitable or not. ○

2.8.6 Further Remarks

To avoid misunderstandings in the presentation of our approach and to gain a deeper insight into it, we end this section with selected questions.

Does f have to be (upper- or lower-) continuous to be top-down analyzable? No, we do not assume any kind of continuity in our approach. Points of discontinuity may be critical, but they do not have to be.

May we assume that f is continuous? No, the top-down approach is defined recursively and the auxiliary functions g_i are not continuous in general. Consider, for example, the continuous polynomial $f(x_1, x_2) := x_1^2 + x_2^2 - 1$, which is the planar "in unit circle" predicate. Then $g_1(x_1, \gamma_2)$ is not continuous in four points for fixed γ_2. The function is illustrated in Figure 2.23. This is why the top-down approach *must* work for discontinuous functions.

Does a critical set of measure zero imply that f is region-suitable? No, not in general. A notorious example is the density of $\mathbb{Q}$ in $\mathbb{R}$. Let $A \subset \mathbb{R}$ be an interval. Although $A \cap \mathbb{Q}$ is a set of measure zero, there is no $\varepsilon > 0$ such that the neighborhood $U_\varepsilon(A \cap \mathbb{Q})$ has a volume smaller than $\mu(A)$. This property contradicts the region-suitability.

Does region-suitability imply a finite critical set? No. A counter-example is the function $x \cdot \sin\left(\frac{1}{x}\right)$, which is region-suitable although it has infinitely many zeros in any neighborhood of zero. (By the way, this function is also value-suitable.) We summarize: *Critical sets of region-suitable functions are countable, but not every countable critical set implies region-suitability.*

Is it possible to neglect isolated points in the analysis? We may never exclude critical points from the analysis because the definition of the region of uncertainty is based on them. We may exclude less-critical points, provided that we adjust the success-probability "by hand". We may neglect non-critical points, provided that we still determine the correct inf-value-suitable bound $\varphi_{\inf f}$. (See also Section 2.2.2.)

May we add additional points to the critical set? Yes, we may add points to the critical set provided that f is still guaranteed to be region-suitable. (See also Section 2.2.2.)

Can we decide if f is top-down analyzable without developing the sequence of replacements? It is a necessary condition for the top-down analyzability of f that g_k is positive everywhere. It is not clear how we can guarantee this property in general without deriving g_k.

2.9 Determining the Lower Fp-safety Bound

We next introduce the design of guards and fp-safety bounds. Guards are necessary to implement guarded evaluations in $\mathcal{A}_\mathrm{G}$. In Section 2.9.1, we explain how guards can be implemented for a wide class of functions, including polynomials. To analyze the behavior of guards, we introduce fp-safety bounds in Section 2.9.2. We explain how we determine the fp-safety bound in the analysis (see Figure 2.24). Furthermore, we prove the fp-safety bounds used in previous sections.

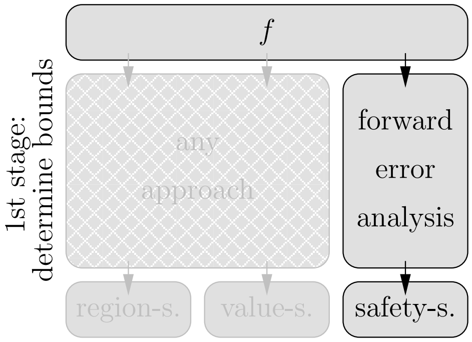


Figure 2.24: An error analysis is used to derive the bounding function for the safety-suitability in the first stage of the analysis.

2.9.1 Implementing Guarded Evaluations

Our presentation of guarded evaluations is based on rounding error analyses following the approach in [34, 11, 60]. A further development is presented in the appendix of [60].

Rounding Error Analysis

The implementation of guards is based on maximum error bounds. To determine the error bounds, we use rounding error analyses. Note that the error bound of a function f depends on the formula E that realizes f and, especially, on the chosen *sequence of evaluation*. We cite some rules to determine error bounds in Table 2.4. Expressions E that are composed of addition, subtraction, multiplication and absolute value can be bounded by the value B_E in the last row of the table. This includes the evaluation of polynomials; for further operators see [34, 11, 60]. The quantities ind_E and $\sup_E$ are derived according to the sequence of evaluation of E. The value ind_x is 0 if $x \in \mathbb{F}_L$, and it is 1 if x is rounded.

Example 2.13. We determine the error bound for the expression

$$E(x_1, \dots x_k)|_\mathbb{F} \quad = \quad (((a \cdot x_1) \cdot x_2) \cdots x_k)|_\mathbb{F}$$

E	$\sup_E$	ind_E
x	$\|\, x\|_{\mathbb{F}}\,\|$	0 or 1
$E_1 \pm E_2$	$(\sup_{E_1} + \sup_{E_2})\|_{\mathbb{F}}$	$1 + \max\{\text{ind}_{E_1}, \text{ind}_{E_2}\}$
$E_1 \cdot E_2$	$(\sup_{E_1} \cdot \sup_{E_2})\|_{\mathbb{F}}$	$1 + \text{ind}_{E_1} + \text{ind}_{E_2}$
$\|E\|$	$\sup_E$	ind_E
$B_E := \text{ind}_E \cdot \sup_E \cdot 2^{-L}$		

Table 2.4: This table reprints parts of Table 2.1 in Funke [34, p. 11]. The row for $|E|$ is added by us.

where $k \in \mathbb{N}$, $a \in \mathbb{R}$ is a coefficient, and $x \in U_\delta(\bar{x})|_{\mathbb{G}} \subseteq [-2^{e_{\max}}, 2^{e_{\max}}]^k$. A worst-case consideration leads to $\text{ind}_a = 1$, and $\sup_a = |\, a|_{\mathbb{F}}\,|$ for the coefficient and $\text{ind}_{x_i} = 0$, and $\sup_{x_i} = |\, x_i|_{\mathbb{F}}\,|$ for $1 \leq i \leq k$. Then we obtain $\text{ind}_{ax_1} = 2$ and $\sup_{ax_1} = |\, ax_1|_{\mathbb{F}}\,|$ after the first multiplication. Taking all multiplications into account, we get $\text{ind}_E = k+1$ and $\sup_E = |\, ax_1 \cdots x_k|_{\mathbb{F}}\,|$. According to Table 2.4, we obtain the *dynamic error bound*

$$B_E(L, x) \quad = \quad (k+1) \cdot |\, ax_1 \cdots x_k|_{\mathbb{F}}\,| \cdot 2^{-L}$$

and the *static error bound*

$$B_E(L) \quad = \quad (k+1) \cdot |\, a|_{\mathbb{F}}\,| \cdot 2^{ke_{\max} - L}$$

where $2^{e_{\max}}$ is an upper bound on the absolute value of a perturbed input. ○

Remark 2.3. We make the observation that the bound $B_E(L)$ approaches zero when L approaches infinity, that means,

$$\lim_{L \to \infty} B_E(L) \quad = \quad 0.$$

Furthermore, we observe that *all error bounds derived from Table 2.4 have this property.* ○

Guarded Evaluation

In guarded algorithms $\mathcal{A}_{\text{G}}$ every predicate evaluation $f(x)|_{\mathbb{F}}$ must be protected by a guard $\mathcal{G}_f(x)$ that verifies the sign of the result. Guards can be implemented using the dynamic (or the weaker static) error bounds. Let $B_f(L,x)$ be an upper bound on the rounding error of $f(x)|_{\mathbb{F}}$ for floating point arithmetic $\mathbb{F}_L$, that means,

$$B_f(L,x) \quad \geq \quad |\, f(x)|_{\mathbb{F}_L} - f(x)\,|. \tag{2.48}$$

We then can immediately derive the implication

$$|\, f(x)|_{\mathbb{F}}\,| > B_f(L,x) \qquad \Rightarrow \qquad \operatorname{sign}(f(x)|_{\mathbb{F}_L}) = \operatorname{sign}(f(x)). \tag{2.49}$$

We use the unequation on the left hand side to construct a *guard* $\mathcal{G}_f$ *for* f where

$$\mathcal{G}_f(x) \quad := \quad \big(\,|\, f(x)|_{\mathbb{F}}\,| > B_f(L,x)\,\big).$$

If $\mathcal{G}_f(x)$ is true, $f(x)$ has the correct sign. Note that this definition is in accordance with Definition 2.2 on Page 17.

Example 2.14. Let u and v be two points in the plane and let ℓ be the unique line that passes through u and v. The side-of-line predicate sol determines for a query point q if it lies to the left of ℓ, to the right of ℓ, or on ℓ. We decide this predicate by the sign of the function

$$\begin{aligned} \operatorname{sol}(u,v,q) &= \operatorname{sol}(u_x,u_y,v_x,v_y,q_x,q_y) \\ &= \det\begin{pmatrix} u_x & u_y & 1 \\ v_x & v_y & 1 \\ q_x & q_y & 1 \end{pmatrix} \\ &= u_xv_y - u_xq_y - q_xv_y + v_xu_y - v_xq_y + q_xu_y. \end{aligned}$$

If the result is positive, q lies to the left of ℓ; if the result is negative, q lies to the right of ℓ; and if the result is zero, q lies on the line. Determining the evaluation order to

$$((u_xv_y - u_xq_y) - (q_xv_y - v_xu_y)) - (v_xq_y - q_xu_y), \tag{2.50}$$

implies the error bound

$$B_{\text{sol}} \quad = \quad 24 \cdot 2^{2e_{\max}-L}$$

according to the presented rounding error analysis. ○

2.9.2 Analyzing Guards With Fp-safety Bounds

We now explain how to analyze the behavior of guards according to [34, 11, 60]. Remember that we perform the analysis in real space. The implication

$$|\, f(x)\,| > 2B_f(L,x) \qquad \Rightarrow \qquad |\, f(x)|_{\mathbb{F}}\,| > B_f(L,x) \tag{2.51}$$

is true because of Formula (2.48). The inequality on the left hand side is a relation that we can safely verify in real space. We can always use the static error bound to construct a *fp-safety bound* $S_{\inf f}$ *for* f as

$$S_{\inf f}(L) \quad := \quad 2B_f(L) \tag{2.52}$$

where $B_f(L)$ is the static error bound. Note that this definition is in accordance with Definition 2.6 on Page 27 because the implications in Formulas (2.49) and (2.51) guarantee the desired implication in Formula (2.6). *Because of Remark 2.3, the fp-safety bound* $S_{\inf f}(L)$ *fulfills the safety-condition on page 33 by construction.* We next derive a fp-safety bound for univariate polynomials.

Corollary 2.18. *Let* f *be a univariate polynomial*

$$f(x) \quad = \quad a_d \cdot x^d + a_{d-1} \cdot x^{d-1} + \ldots + a_1 \cdot x + a_0 \tag{2.53}$$

of degree d*. Then*

$$S_{\inf f}(L) \quad := \quad (d+2) \cdot \max_{1 \leq i \leq d} |a_i| \cdot 2^{(d+1)e_{\max}+1-L} \tag{2.54}$$

is a fp-safety bound for f *on* $[-2^{e_{\max}}, 2^{e_{\max}}]$ *where* $e_{\max} \in \mathbb{N}$.

Proof. We apply the error analysis of this section. We evaluate Formula (2.53) from the right to the left. We obtain the static error bound

$$B_f(L) \quad := \quad \mathrm{ind}_f \cdot \sup_f \cdot 2^{-L} \quad = \quad (d+2) \cdot \left(\max_{1 \leq i \leq d} |a_i| \cdot 2^{(d+1)e_{\max}} \right) \cdot 2^{-L}.$$

Finally, we set the fp-safety bound to $S_{\inf f}(L) := 2B_f(L)$. □

Multiplications usually cause larger rounding errors than additions. Surprisingly, the evaluation of univariate polynomials with the Horner scheme[26] (which minimize the number of multiplications) does not lead to a smaller error bound than the one we have derived in the proof. We next derive an error bound for k-variate polynomials. We define $x^\iota := x_1^{\iota_1} \cdot \ldots \cdot x_k^{\iota_k}$ for $\iota \in \mathbb{N}_0^k$ and $x \in \mathbb{R}^k$.

Corollary 2.19. *Let* f *be the* k*-variate polynomial* ($k \geq 2$)

$$f(x) \quad := \quad \sum_{\iota \in \mathcal{I}} a_\iota x^\iota$$

where $\mathcal{I} \subset \mathbb{N}_0^k$ *is finite, and* $a_\iota \in \mathbb{R}_{\neq 0}$ *for all* $\iota \in \mathcal{I}$*. Let* d *be the total degree of* f*, and let* N_T *be the number of terms in* f*. Then*

$$S_{\inf f}(L) \quad := \quad (d + 1 + \lceil \log N_\mathrm{T} \rceil) \cdot N_\mathrm{T} \cdot \max_{\iota \in \mathcal{I}} |a_\iota| \cdot 2^{de_{\max}+1-L} \tag{2.55}$$

is a fp-safety bound for f *on* $[-2^{e_{\max}}, 2^{e_{\max}}]^k$ *where* $e_{\max} \in \mathbb{N}$.

[26] For Horner scheme see Hotz [43].

Proof. We begin with the determination of the error bound B_f. The maximum absolute value of the term $a_\iota x^\iota$ is obviously upper-bounded by the product of a bound on a_ι and a bound on x^ι. Because $|x_i| \leq 2^{e_{\max}}$ for all $1 \leq i \leq k$, we have

$$\sup_{a_\iota x^\iota} \quad \leq \quad \max_{\iota \in \mathcal{I}} |a_\iota| \cdot 2^{de_{\max}}.$$

Since we know the number N_{T} of terms in f, we can then upper-bound $\sup_f$ by

$$\sup_f \quad \leq \quad N_{\mathrm{T}} \cdot \max_{\iota \in \mathcal{I}} |a_\iota| \cdot 2^{de_{\max}}.$$

In addition, we have $\mathrm{ind}_{a_\iota x^\iota} = d + 1$ since we evaluate d multiplications, and only a_ι may not be in the set $\mathbb{F}$. (Remember that, because of the perturbation, the values x_i belong to the grid $\mathbb{G}$ which is a subset of $\mathbb{F}$.)

To keep ind_f as small as possible, we sum up the N_{T} terms pairwise such that the tree of evaluation has depth $\lceil \log N_{\mathrm{T}} \rceil$. This leads to $\mathrm{ind}_f = d + 1 + \lceil \log N_{\mathrm{T}} \rceil$. Therefore, we conclude that

$$B_f(L) \quad = \quad (d + 1 + \lceil \log N_{\mathrm{T}} \rceil) \cdot \left(N_{\mathrm{T}} \cdot \max_{\iota \in \mathcal{I}} |a_\iota| \cdot 2^{de_{\max}} \right) \cdot 2^{-L}.$$

As usual, we set $S_{\inf f}(L) := 2B_f(L)$. □

Example 2.15. (Continuation of Example 2.14). According to Formula (2.52), the lower floating-point safety bound of the side-of-line predicate is

$$S_{\mathrm{inf,sol}} \quad = \quad 48 \cdot 2^{2e_{\max} - L}.$$

Unfortunately, it is necessary to perform the error analysis step-by-step to achieve this result. We now see that we can also use Corollary 2.19 to determine the safety bound. We easily derive $d = 2$, $N_{\mathrm{T}} = 6$, and $\max_{\iota \in \mathcal{I}} |a_\iota| = 1$ from Formula (2.50). Due to Formula (2.55), this leads to the safety bound

$$\begin{aligned} S'_{\mathrm{inf,sol}} \quad &= \quad (2 + 1 + 3) \cdot 6 \cdot 1 \cdot 2^{2e_{\max} + 1 - L} \\ &= \quad 72 \cdot 2^{2e_{\max} - L}. \end{aligned}$$

We observe that the step-by-step analysis may lead to better a result, but the derived corollaries can help us to accelerate the overall analysis for polynomials. ○

2.10 The Treatment of Range Errors (All Components)

In this section, we address a floating-point issue that is caused by poles of rational functions. So far, the implementation and analysis of functions has been based on the fact that signs of floating-point evaluations are only non-reliable on certain environments of zero. We argue that signs of evaluations may also be non-reliable on environments of poles. We do this for the purpose of embedding rational functions into our theory. In Section 2.10.1, we extend the previous implementation considerations such that they can deal with range errors. In Section 2.10.2, we expand the analysis to range errors of the floating-point arithmetic $\mathbb{F}$. *This is the first time that we present the practical and theoretical treatment of range errors in order to include rational functions in the analysis.*

2.10.1 Extending the Implementation

We examine the simple rational function $f(x) = \frac{1}{x}$. It is well-known that the function value of f at the pole $x = 0$ does not exist in $\mathbb{R}$ (unless we introduce the unsigned symbolic value $\pm\infty$, see Forster [30]). We observe that we cannot determine the function value of f in a neighborhood of a pole with floating-point arithmetic $\mathbb{F}_{L,K}$ because the absolute value of f may be *too large.* Moreover, we observe that the *sign of f may change* on a neighborhood of a pole. Both observations suggest that *poles play a role similar to that of zeros in the context of controlled perturbation.* We now extend the implementation such that it becomes able to deal with range errors.

We extend the implementation of guarded evaluations in the following way: If the absolute value of f cannot be represented with the floating-point arithmetic $\mathbb{F}_{L,K}$ because it is too large, we abort $\mathcal{A}_{\mathrm{G}}$ with the notification of a *range error.* We do not concern ourselves with the source of the range error: It may be "division by zero" or "overflow." The implementation of the second guard per evaluation is straight forward. Some programming languages provide an exception handling that can be used for this objective.

In addition, we must change the implementation of the controlled perturbation algorithm $\mathcal{A}_{\mathrm{CP}}$. If $\mathcal{A}_{\mathrm{G}}$ fails because of a *range error,* we increase the bit length K of the exponent (instead of the precision L). Be aware that we are talking about the exponent, so an additive augmentation of the bit length implies a multiplicative augmentation of the range. These simple changes guarantee that the floating-point arithmetic $\mathbb{F}_{L,K}$ gets adjusted to the necessary dimensions in neighborhoods of poles or in regions where the function value is extremely large.

2.10.2 Extending the Analysis of Functions

For the purpose of dealing with range errors in the analysis, we need to adapt several parts of the analysis tool box. Below we present the necessary changes and extensions in the same order in which we have developed the theory.

Criticality and the region-suitability

The changes to deal with range errors affect the interface between the two stages of the analysis of functions. First, we extend the definition of criticality. We demand that certain points (e.g., poles of rational functions) are critical, too, and refine Definition 2.7 in the following way.

Definition 2.20 (critical)**.** Let $(f, k, A, \delta, e_{\max})$ be a predicate description. We call a point $c \in \bar{U}_\delta(\bar{x})$ *critical* if

$$\inf_{x \in U_\varepsilon(c) \setminus \{c\}} |f(x)| = 0 \qquad \text{or} \qquad \sup_{x \in U_\varepsilon(c) \setminus \{c\}} |f(x)| = \infty$$

on a neighborhood $U_\varepsilon(c)$ for infinitesimal small $\varepsilon > 0$. Furthermore, we call c *less-critical* if c is not critical, but $f(c) = 0$ or c is a pole. Points that are neither critical nor less-critical are called *non-critical*.

For simplicity and as before, we define the *critical set* $C_{f,\delta}$ to be the union of critical and less-critical points within $\bar{U}_\delta(\bar{x})$. Be aware that the new definition of criticality may expand the region of uncertainty. As a consequence, it affects the *region-suitability* and the bound ν_f, respectively χ_f. Note that Definition 2.20 guarantees that we exclude neighborhoods of poles from now on. Because we have integrated poles into the definition of criticality, we have implicitly adapted the region-suitability.

The sup-value-suitability

We have only considered $\inf |f|$ outside of the region of uncertainty so far. To get a quantified description of range issues in the analysis, we need to consider $\sup |f|$ as well. What we have called value-suitability so far is now called, more precisely, *inf-value-suitability*. Its bounding function, that we have called $\varphi_f(\gamma)$ so far, is now called $\varphi_{\inf f}(\gamma)$.

In addition to Definition 2.14, we introduce *sup-value-suitability*, that means, there is an upper-bounding function $\varphi_{\sup f}(\gamma)$ on the absolute value of f outside of the region of uncertainty R_f. We show how the new bound is determined with the bottom-up approach later on. Based on the new terminology, we call f *(totally) value-suitable* if f is both inf-value-suitable and sup-value-suitable.

The sup-safety-suitability and analyzability

We also extend Definition 2.15. What we have called safety-suitability so far is now called, more precisely, *inf-safety-suitability*. Its bounding function $S_{\inf f}(L)$ is now called the *lower fp-safety bound*.

In addition we introduce *sup-safety-suitability*, that means, there is an invertible upper-bounding function $S_{\sup f}(K)$ on the absolute value of f with the following meaning: If we know that

$$|f(x)| \quad \leq \quad S_{\sup f}(K),$$

then $f(x)|_{\mathbb{F}}$ is definitely a finite number in $\mathbb{F}_{L,K}$. We call $S_{\sup f}(K)$ the *upper fp-safety bound*. Such a bound is trivially given by[27]

$$S_{\sup f}(K) \quad := \quad 2^{2^{K-1}} - S_{\inf f}(L).$$

Based on the new terminology, we call f *(totally) safety-suitable* if f is both inf-safety-suitable and sup-safety-suitable. As a consequence, we call f *analyzable* if f is region-suitable, value-suitable (both subtypes) and safety-suitable (both subtypes).

The method of quantified relations

We next extend the method of quantified relations such that the new bounds on the range of floating-point arithmetic are included into the analysis. In addition to the precision function $L_f(p)$, we determine the bounding function

$$K_f(p) \quad := \quad \left\lceil S_{\sup f}^{-1}\left(\varphi_{\sup f}\left(t \cdot \nu_f^{-1}\left(\varepsilon_\nu\left(p\right)\right)\right)\right)\right\rceil .$$

We deduce the maximum absolute value of f outside of the region of uncertainty from the probability; then, we use the upper fp-safety bound to deduce the necessary bit length of the exponent. The derivation of $K_f(p)$ is absolutely analog to the derivation of $L_{\text{safe}}(p)$ in Steps 1–5 of the method of quantified relations.

We summarize our results so far: If we have the bounding functions of the interface of the function analysis, we know that the floating-point arithmetic $\mathbb{F}_{L_f(p),K_f(p)}$ is sufficient to safely evaluate f at a random grid point in the perturbation area with probability p.

Furthermore, we can derive a probability function p_f if f is analyzable and $\varphi_{\inf f}$ and $\varphi_{\sup f}$ are both invertible. Analog to the definition of $p_{\inf}(L)$ in Section 2.5.2, we derive the additional bound on the probability

$$p_{\sup}(K) \quad := \quad \varepsilon_\nu^{-1}\left(\nu_f\left(\frac{1}{t} \cdot \varphi_{\sup f}^{-1}\left(S_{\sup f}(K)\right)\right)\right)$$

from $K_f(p)$. This leads to the final *probability function* $p_f : \mathbb{N} \times \mathbb{N} \to (0,1)$ where

$$p_f(L,K) \quad := \quad \min\left\{p_{\inf}(L),\, p_{\sup}(K),\, p_{\text{grid}}(L)\right\}$$

for parameter $t \in (0,1)$.

The bottom-up approach

We now extend the calculation rules of the bottom-up approach to also derive the bounding function $\varphi_{\sup f}(\gamma)$ from simpler sup-value-suitable functions. First, we replace the lower-bounding rule in Theorem 2.9 with the following sandwich-rule.

[27] First, the largest floating-point number that is representable with $\mathbb{F}_{L,K}$ is $(2-2^{-L})2^{2^{K-1}}$. Second, we must take the maximal floating-point rounding error into account.

Theorem 2.20 (sandwich). *Let $(f, k, A, \delta, e_{\max}, \Gamma\text{-line}, t)$ be a predicate description. If there is a region-value-suitable function $g : \bar{U}_\delta(A) \to \mathbb{R}$ and $c_1, c_2 \in \mathbb{R}_{>0}$ where*

$$c_1 \, |g(x)| \quad \leq \quad |f(x)| \quad \leq \quad c_2 \, |g(x)|,$$

then f is also region-value-suitable with the following bounding functions:

$$\begin{aligned} \nu_f(\gamma) &:= \nu_g(\gamma) \\ \varphi_{\inf f}(\gamma) &:= c_1 \varphi_{\inf g}(\gamma) \\ \varphi_{\sup f}(\gamma) &:= c_2 \varphi_{\sup g}(\gamma). \end{aligned}$$

If f is in addition safety-suitable, f is analyzable.

Proof. The region-suitability and inf-value-suitability follows from the proof of Theorem 2.9. The sup-value-suitability is proven in a similar way to Part 2 of the same proof. □

We next extend the product rule in Theorem 2.10. We merely add the assignment

$$\varphi_{\sup f}(\gamma) \quad := \quad \varphi_{\sup g}(\gamma_1, \ldots, \gamma_\ell) \cdot \varphi_{\sup h}(\gamma_{j+1}, \ldots, \gamma_k)$$

after Formula (2.32). Its proof follows Part 1 of the proof of Theorem 2.10.

Finally, we extend the min-rule and the max-rule in Theorem 2.11. We add the two assignments

$$\begin{aligned} \varphi_{\sup f_{\min}}(\gamma) &:= \min\{\varphi_{\sup g}(\gamma_1, \ldots, \gamma_\ell), \varphi_{\sup h}(\gamma_{j+1}, \ldots, \gamma_k)\} \quad \text{and} \\ \varphi_{\sup f_{\max}}(\gamma) &:= \max\{\varphi_{\sup g}(\gamma_1, \ldots, \gamma_\ell), \varphi_{\sup h}(\gamma_{j+1}, \ldots, \gamma_k)\} \end{aligned}$$

after Formula (2.37). Their proofs follow Part 1 of the proof of Theorem 2.10.

The top-down approach

Similar to the functions $\varphi_{\inf g_i}$, which are simply called φ_{g_i} in the overview in Figure 2.20, we determine the functions $\varphi_{\sup g_i}$ in the second phase of the pseudo-top-down approach in a bottom-up fashion.

This completes the integration of the range considerations into the analysis tool box. Please note that none of the changes presented in this section restrict the applicability of the analysis tool box in any way. On the contrary, they are necessary for the correctness and generality of the tool box.

2.11 The Analysis of Rational Functions

We have solved the arithmetical issues that occur in the implementation and analysis of rational functions. In addition, we must solve technical issues in the implementation of guards and, moreover, provide a general technique to derive a quantitative analysis for rational functions. *This is the first time that we include rational functions in the analysis.*

Let $f := \frac{g}{h}$ be a rational function, that means, let g and h be multivariate polynomials. Let k be the number of arguments of f, i.e., we consider $f(x)$ where $x = (x_1, \ldots, x_k)$. The arguments of g and h may be any subsequence of x, but each x_i is at least an argument of g or an argument of h. We know that g and h are analyzable (see Section 2.7.5).

First, we discuss the implementation of guards for rational functions. We make the observation that—independent of the evaluation sequences of g and h—the *division* of the value of g by the value of h is the *very last operation* in the evaluation of f. Because of the standardization of floating-point arithmetic (e.g., see [44]), the sign of f is computed correctly if the signs of g and h are computed correctly. Therefore, it is sufficient for an implementation of a predicate that branches on the sign of a rational function f to use the guard $\mathcal{G}_f := (\mathcal{G}_g \wedge \mathcal{G}_h)$.

How do we analyze this predicate, that means, how can we relate the known quantities? Let x be given. In the case that the (dependent) arguments of g and h lie outside of their region of uncertainty, we can deduce the relation

$$\frac{S_{\inf g}}{S_{\sup h}} \quad \leq \quad f(x) \quad \leq \quad \frac{S_{\sup g}}{S_{\inf h}}. \tag{2.56}$$

Unfortunately, this is not what we need. In this way, we can only deduce the value of f from the values of g and h, but not vice versa: If $f(x)$ fulfills Formula (2.56), we cannot deduce that the guards $\mathcal{G}_g$ and $\mathcal{G}_h$ are true. For example, assume that $f(x) = 1$; then we know that the values of g and h are equal, but we do not know if their values are fp-safe or close to zero.

Therefore, we choose a different way to analyze the behavior of guard $\mathcal{G}_f$. Since g and h are multivariate polynomials, we can analyze the behavior of $\mathcal{G}_g$ and $\mathcal{G}_h$ and derive the precision functions $L_g(p)$ and $L_h(p)$, as we have seen in earlier sections. If we demand that g and h evaluate successfully with probability $\frac{1+p}{2}$ each, f evaluates successfully with probability p since the sum of the failure probability of g and h is at most $(1-p)$. This leads to the precision function

$$L_f(p) \quad := \quad \max\left\{ L_g\left(\frac{1+p}{2}\right), L_h\left(\frac{1+p}{2}\right) \right\},$$

which reflects the behavior of $\mathcal{G}_f$ and, therefore, analyzes the behavior of an implementation of the rational function evaluation of f.

2.12 General Analysis of Algorithms (Composition)

We have only presented components of the tool box so far that are used to analyze functions. We now introduce the components that are used to analyze an entire controlled perturbation algorithm $\mathcal{A}_{\mathrm{CP}}$. Figure 2.25 illustrates the analysis of algorithms. Similar to the analysis of functions, the algorithm analysis has two stages. The *interface* between the stages is introduced in Section 2.12.1. It consists of necessary algorithm properties (to the left of the dashed line) and the analyzability of the used predicates (to the right of the dashed line). We also explain how to determine the bounds associated with the algorithm properties. In Section 2.12.2, we give an overview of the algorithm properties. The *method of distributed probability* represents the actual analysis of algorithms and is presented in Section 2.12.3. It is followed by an example in Section 2.12.4.

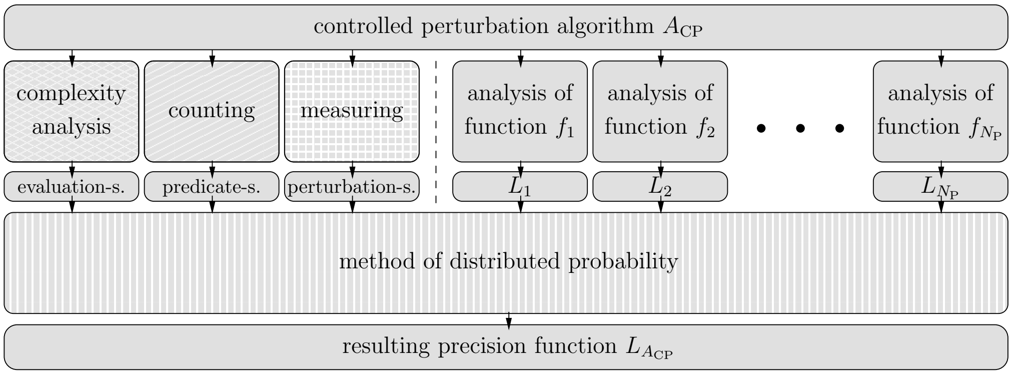

controlled perturbation algorithm A_{CP}
complexity analysis
counting
measuring
analysis of function f_1
analysis of function f_2
analysis of function $f_{N_{\mathrm{P}}}$
evaluation-s.
predicate-s.
perturbation-s.
L_1
L_2
$L_{N_{\mathrm{P}}}$
method of distributed probability
resulting precision function $L_{A_{\mathrm{CP}}}$

Figure 2.25: Illustration of the analysis of controlled perturbation algorithms.

2.12.1 Necessary Conditions for the Analysis of Algorithms

We introduce some properties of controlled perturbation algorithms. Sometimes we use the same names for algorithm and function properties to emphasize the analog. We describe to which algorithms we can *apply* controlled perturbation, for which ones we can *verify* that they terminate, and which ones can be *analyzed* in a quantitative way because they are *suitable* for the analysis.

Definition 2.21 (applicable.)**.** Let $\mathcal{A}_{\mathrm{CP}}$ be a controlled perturbation algorithm. We call $\mathcal{A}_{\mathrm{CP}}$ *applicable* if there is a precision function $L_{\mathcal{A}_{\mathrm{CP}}} : (0,1) \times \mathbb{N} \to \mathbb{N}$ and $\eta \in \mathbb{N}$ with the property: At least one from η runs of the embedded guarded algorithm $\mathcal{A}_{\mathrm{G}}$ is expected to terminate successfully for a randomly perturbed input of size $n \in \mathbb{N}$ with probability at least $p \in (0,1)$ for every precision $L \in \mathbb{N}$ with $L \geq L_{\mathcal{A}_{\mathrm{CP}}}(p,n)$.

The applicability of an algorithm has a strong meaning: For every success probability $p \in (0,1)$ and for every input size $n \in \mathbb{N}$ there is still a *finite* precision that fulfills the requirements. As a matter of fact, a controlled perturbation algorithm reaches this precision in *finite* many steps. This means if the algorithm $\mathcal{A}_{\mathrm{CP}}$ is applicable, its execution is guaranteed to terminate.

In the definition of applicability, we define the precision function $L_{\mathcal{A}_{\mathrm{CP}}}(p,n)$ as a function in the desired success probability p and the input size n. The bound also depends on other quantities like the perturbation parameter δ, an upper bound on the absolute input values, or the maximum rounding-error. However, the latter quantities have some influence on the determination of the bounding functions in the analysis of functions; they occur as parameters in formula $L_{\mathcal{A}_{\mathrm{CP}}}$ and are not mentioned as arguments.

Definition 2.22 (verifiable.)**.** Let $\mathcal{A}_{\mathrm{CP}}$ be a controlled perturbation algorithm. We call $\mathcal{A}_{\mathrm{CP}}$ *verifiable* if the following conditions are fulfilled:
1. All used predicates are verifiable.
2. The perturbation area $\mathcal{U}_{\mathcal{A}_{\mathrm{CP}},\delta}(\bar{y})$ contains an open neighborhood of $\bar{y}$.
3. The total number of predicate evaluations is bounded.
4. The number of predicates is bounded.

Three properties are necessary for the analyzability of algorithms: evaluation-, predicate- and perturbation-suitability. However, the three conditions are not sufficient for the analysis of algorithms since there are also prerequisites on the used predicates. In the following, we define the various properties, explain how we obtain the bounding functions that are associated with the necessary conditions, and show how the algorithm properties are related to each other.

Definition 2.23. Let $\mathcal{A}_{\mathrm{CP}}$ be a controlled perturbation algorithm.

- (predicate-suitable). We call $\mathcal{A}_{\mathrm{CP}}$ *predicate-suitable* if the number of different predicates is upper-bounded by a function $N_{\mathrm{P}} : \mathbb{N} \to \mathbb{N}$ in dependence on the input size n.
- (evaluation-suitable). We call $\mathcal{A}_{\mathrm{CP}}$ *evaluation-suitable* if the total number of predicate evaluations is upper-bounded by a function $N_{\mathrm{E}} : \mathbb{N} \to \mathbb{N}$ in dependence on the input size n.
- (perturbation-suitable). Let $\mathcal{U}_{\mathcal{A}_{\mathrm{CP}},\delta}(\bar{y})$ be the perturbation area of $\mathcal{A}_{\mathrm{CP}}$ around $\bar{y}$; we assume that $\mathcal{U}_{\mathcal{A}_{\mathrm{CP}},\delta}(\bar{y})$ is scalable with parameter δ and that it has a fixed shape, e.g., cube, box, sphere, ellipsoid, etc. We call $\mathcal{A}_{\mathrm{CP}}$ *perturbation-suitable* if there is a bounding function $V : \mathbb{R}^k_{>0} \to \mathbb{R}_{>0}$ such that there is an open axis-parallel box $U_{\mathcal{A}_{\mathrm{CP}},\delta}(\bar{y})$ with volume at least $V(\delta)$ and $U_{\mathcal{A}_{\mathrm{CP}},\delta}(\bar{y}) \subset \mathcal{U}_{\mathcal{A}_{\mathrm{CP}},\delta}(\bar{y})$.
- (analyzable). We call $\mathcal{A}_{\mathrm{CP}}$ *analyzable* if the following conditions are fulfilled:
  1. All used predicates are analyzable.
  2. $\mathcal{A}_{\mathrm{CP}}$ is predicate-suitable, evaluation-suitable, and perturbation-suitable.

In the definition of the predicate-suitability, the number $N_{\mathrm{P}} \in \mathbb{N}$ of different predicates is usually fixed for a geometric algorithm. Our motivation was to keep the presentation as general as possible.

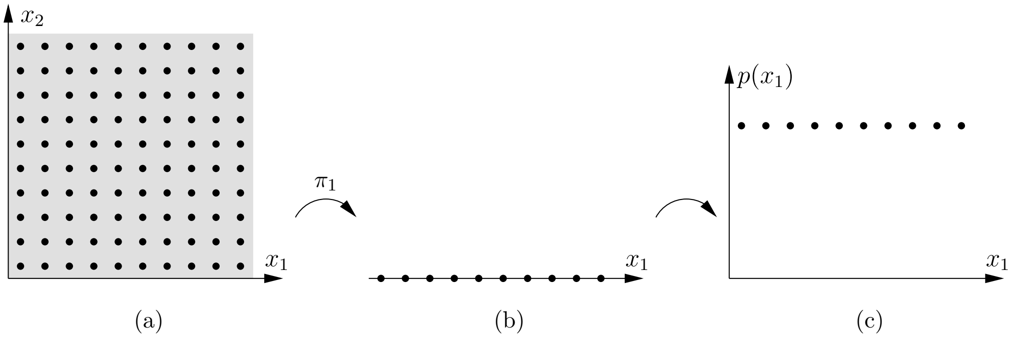


Figure 2.26: (a) The original perturbation area $\mathcal{U}_{\mathcal{A}_{\mathrm{CP}},\delta}$ is an axis-parallel box. (b) Its projection is uniformly distributed. (c) The points in the projection are chosen with the same probability.

We allow any shape of the perturbation area $\mathcal{U}_{\mathcal{A}_{\mathrm{CP}},\delta}$ that fulfills the condition in the definition. As opposed to that, we have assumed that the perturbation area $U_{f,\delta}$ in the analysis of functions is an axis-parallel box. This seems contradictory and needs further explanation. After each perturbation $y \in \mathcal{U}_{\mathcal{A}_{\mathrm{CP}},\delta}(\bar{y})|_{\mathbb{G}_{L,K}}$ of the input, we try to evaluate the whole sequence of predicates. We assume that the random perturbation is chosen from a discrete uniform distribution in a subset of the n-dimensional space. As opposed to that, function f_i has just $k_i \ll n$ arguments $x = (x_1, \ldots, x_{k_i})$. Mathematically speaking, we determine the input x of f_i by an orthogonal projection of y onto a k_i-dimensional plane. We now make the following observation: *If we examine the orthogonal projection onto a k_i-dimensional plane, the projected points do not occur with the same probability in general if the perturbation area is not an axis-parallel box.* We illustrate an example in Figures 2.26 and 2.27. However, we prove in Section 2.12.3 that there is an implementation of $\mathcal{A}_{\mathrm{CP}}$ that we can analyze—given that we know the bounding function V mentioned in the definition of perturbation-suitability.

We next explain how we can determine the three bounding functions associated with the three necessary algorithm properties (see Figure 2.25). We count the number N_{P} of used predicates to determine the bounding function $N_{\mathrm{P}}(n)$. We usually obtain the bounding function $N_{\mathrm{E}}(n)$ on the number of predicate evaluations with a complexity analysis. The bound η results from a geometric consideration: We only need to determine the volume of the perturbation area. If the perturbation area has an ordinary shape, its computation is straight forward. We consider an example.

Example 2.16. Let the input of $\mathcal{A}_{\mathrm{CP}}$ be m points in the plain, that means, $n = 2m$. In addition, let the perturbation area around each point be a disc of radius δ. Then the axis-parallel box of maximum volume inside of such a disc is a square and has

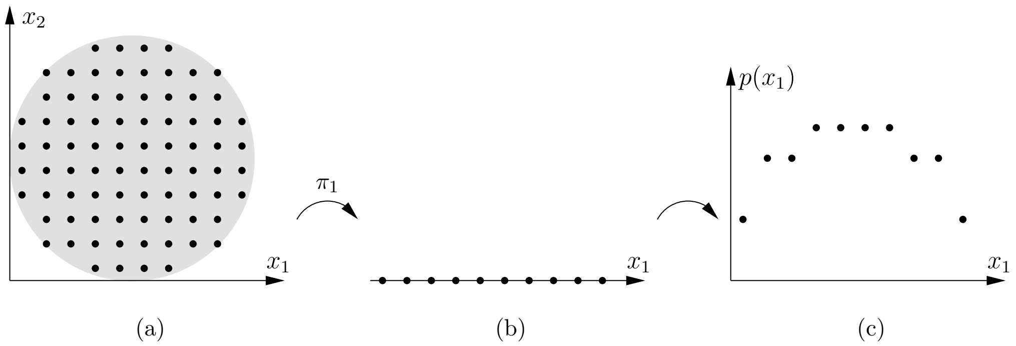


Figure 2.27: (a) The original perturbation area $\mathcal{U}_{\mathcal{A}_{\mathrm{CP}},\delta}$ is a sphere. (b) Its projection is uniformly distributed. (c) The points in the projection are not chosen with the same probability.

edge length $\delta\sqrt{2}$. We obtain:

$$\begin{aligned}\eta &:= \left\lceil \frac{\mu(\mathcal{U}_\delta)}{\mu(U_\delta)} \right\rceil \\ &= \left\lceil \frac{\mu(m \text{ discs of radius } \delta)}{\mu(m \text{ cubes of edge length } \delta\sqrt{2})} \right\rceil \\ &= \left\lceil \frac{m \cdot \pi\delta^2}{m \cdot 2\delta^2} \right\rceil \\ &= 2.\end{aligned}$$

We observe that the bound η does not depend on m (or n). ○

We prove two implications.

Lemma 2.21. *Let algorithm $\mathcal{A}_{\mathrm{CP}}$ be analyzable. Then $\mathcal{A}_{\mathrm{CP}}$ is verifiable.*

Proof. This is trivially true. □

Lemma 2.22. *Let algorithm $\mathcal{A}_{\mathrm{CP}}$ be verifiable. Then $\mathcal{A}_{\mathrm{CP}}$ is applicable.*

Proof. To show that $\mathcal{A}_{\mathrm{CP}}$ is applicable, we prove the following existence. *There is $\eta \in \mathbb{N}$ such that for every $p \in (0,1)$ and every $n \in \mathbb{N}$, there is a precision $\mathcal{L}_{p,n}$ with the property: For a randomly perturbed input of size n, at least one from η runs of $\mathcal{A}_{\mathrm{G}}$ is expected to terminate successfully with probability at least p for every precision $L \in \mathbb{N}$ with $L \geq \mathcal{L}_{p,n}$.* Then the function $L_{\mathcal{A}_{\mathrm{CP}}}(p,n) := \mathcal{L}_{p,n}$ has the desired property, which proves the claim.

First, we show that there is an appropriate $\eta \in \mathbb{N}$. Because $\mathcal{A}_{\mathrm{CP}}$ is verifiable, the perturbation area $\mathcal{U}_{\mathcal{A}_{\mathrm{CP}},\delta}(\bar{y})$ contains an *open* set around $\bar{y}$. Therefore, there is an open axis-parallel box around $\bar{y}$ with $U_\delta(\bar{y}) \subset \mathcal{U}_{\mathcal{A}_{\mathrm{CP}},\delta}(\bar{y})$. Then there is also a natural number

$$\eta := \left\lceil \frac{\mu\left(\mathcal{U}_{\mathcal{A}_{\mathrm{CP}},\delta}(\bar{y})\right)}{\mu\left(U_\delta(\bar{y})\right)} \right\rceil.$$

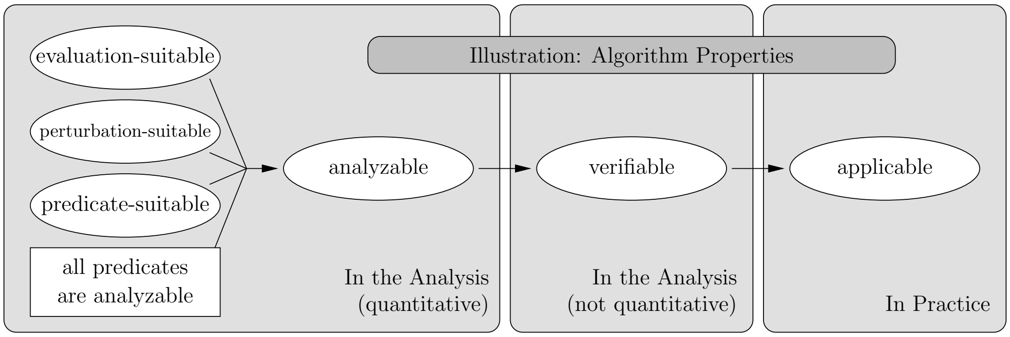


Illustration: Algorithm Properties
evaluation-suitable
perturbation-suitable
predicate-suitable
all predicates are analyzable
analyzable
verifiable
applicable
In the Analysis (quantitative)
In the Analysis (not quantitative)
In Practice

Figure 2.28: The illustration summarizes the implications of the various algorithm properties we have defined in this section.

That means, if we randomly choose η points from a uniformly distributed grid in $\mathcal{U}_{\mathcal{A}_{\mathrm{CP}},\delta}(\bar{y})|_{\mathbb{G}_{L,K}}$, we may expect that at least one point lies also inside of $U_\delta(\bar{y})$.

Let $p \in (0,1)$, and let $n \in \mathbb{N}$. In addition, let $y \in U_\delta(\bar{y})|_{\mathbb{G}_{L,K}}$ be chosen randomly. Since $\mathcal{A}_{\mathrm{CP}}$ is verifiable, there is an upper-bound $N_{\mathrm{E}} \in \mathbb{N}$ on the total number of predicate evaluations. Therefore, we can distribute the total failure probability $(1-p)$ among the N_{E} predicate evaluations. We call this failure probability

$$\varrho \quad := \quad \frac{1-p}{N_{\mathrm{E}}}.$$

Obviously, $\mathcal{A}_{\mathrm{G}}(y)$ is successful with probability p if every predicate evaluation fails with probability at most ϱ.

Let $N_{\mathrm{P}} \in \mathbb{N}$ be the number of different predicates in $\mathcal{A}_{\mathrm{G}}$ that are decided by the functions $f_1, \ldots, f_{N_{\mathrm{P}}}$. Because $\mathcal{A}_{\mathrm{CP}}$ is verifiable, all used predicates are verifiable and thus applicable. Then Definition 2.11 implies the existence of precision functions $L_{f_1}, \ldots, L_{f_{N_{\mathrm{P}}}}$. Therefore, there is a precision

$$\mathcal{L}_{p,n} \quad := \quad \max_{1 \le i \le N_{\mathrm{P}}} \; L_{f_i}(1-\varrho)\,,$$

which has the desired property because of Definition 2.11. This finishes the proof. □

As a consequence of Lemma 2.21 and Lemma 2.22, the controlled perturbation implementation $\mathcal{A}_{\mathrm{CP}}$ terminates with certainty and yields the correct result for the perturbed input if $\mathcal{A}_{\mathrm{CP}}$ is analyzable.

2.12.2 Overview: Algorithm Properties

An overview of the defined algorithm properties is shown in Figure 2.28. They can be summarized as follows: A controlled perturbation algorithm $\mathcal{A}_{\mathrm{CP}}$ is guaranteed to terminate if $\mathcal{A}_{\mathrm{CP}}$ is *applicable*. If $\mathcal{A}_{\mathrm{CP}}$ is *verifiable*, we can prove that $\mathcal{A}_{\mathrm{CP}}$ terminates—even if we are not able to analyze its performance. And finally, we can give a quantitative analysis of the performance of $\mathcal{A}_{\mathrm{CP}}$ if $\mathcal{A}_{\mathrm{CP}}$ is *analyzable*.

The implications are: An evaluation-, perturbation- and predicate suitable algorithm that uses solely analyzable predicates is analyzable (see Definition 2.21). An analyzable algorithm is also verifiable (see Lemma 2.21). And finally, a verifiable algorithm is also applicable (see Lemma 2.22).

2.12.3 The Method of Distributed Probability

In this section, we introduce the method of distributed probability, which is used to analyze complete algorithms. Figure 2.25 shows the component and its interface.

Theorem 2.23 (distributed probability)**.** *Let $\mathcal{A}_{\mathrm{CP}}$ be analyzable. Then there is a general method to determine a precision function $L_{\mathcal{A}_{\mathrm{CP}}} : (0,1) \times \mathbb{N} \to \mathbb{N}$ and $K_{\mathcal{A}_{\mathrm{CP}}} : (0,1) \times \mathbb{N} \to \mathbb{N}$ and $\eta \in \mathbb{N}$ with the property: At least one from η runs of the embedded guarded algorithm $\mathcal{A}_{\mathrm{G}}$ is expected to terminate successfully for a randomly perturbed input of size n with probability at least $p \in (0,1)$ for every arithmetic $\mathbb{F}_{L,K}$ where $L \geq L_{\mathcal{A}_{\mathrm{CP}}}(p,n)$ and $K \geq K_{\mathcal{A}_{\mathrm{CP}}}(p,n)$.*

Proof. We prove the claim in three steps: First, we derive $\eta \in \mathbb{N}$ from the shape of the region of uncertainty. Then we determine a bound on the failure probability of each predicate evaluation. And finally, we analyze each predicate type to determine the worst-case precision. An overview of these steps is given in Table 2.5.

Step 1 (define η). We define η as the ratio

$$\eta \quad = \quad \left\lceil \frac{V(\delta)}{\mu(U_\delta(\bar{y}))} \right\rceil .$$

That means, if we randomly choose η points from a uniformly distributed grid in $\mathcal{U}_{\mathcal{A}_{\mathrm{CP}},\delta}(\bar{y})|_{\mathbb{G}_{L,K}}$, we may expect that at least one point lies also inside of $U_{\mathcal{A}_{\mathrm{CP}},\delta}(\bar{y})$.

Step 2 (define ρ). Let $p \in (0,1)$ be the desired success probability of the guarded algorithm $\mathcal{A}_{\mathrm{G}}$. Then $(1-p)$ is the failure probability of $\mathcal{A}_{\mathrm{G}}$. There are at most $N_{\mathrm{E}}(n)$ predicate evaluations for an input of size n. The guarded algorithm succeeds if and only if we evaluate all predicates successfully in a row for *the same* perturbed input. We observe that the evaluations do not have to be independent. Therefore, we define the failure probability of each predicate evaluation as the function

$$\varrho(p,n) \quad := \quad \frac{1-p}{N_{\mathrm{E}}(n)}$$

in dependence on p and n.

Step 1:	determine number of runs (define η)
Step 2:	determine "per evaluation" probability (define ρ)
Step 3:	compose precision function (define $L_{\mathcal{A}_{\mathrm{CP}}}$ and $K_{\mathcal{A}_{\mathrm{CP}}}$)

Table 2.5: Instructions for performing the method of distributed probability.

Step 3 (define $L_{\mathcal{A}_{\mathrm{CP}}}$ and $K_{\mathcal{A}_{\mathrm{CP}}}$). There are at most $N_{\mathrm{P}}(n)$ different predicates. Let $f_1, \dots, f_{N_{\mathrm{P}}(n)}$ be the functions that realize these predicates. Since all functions are analyzable, we determine their precision function L_{f_i} with the presented methods of our analysis tool box. Then we define the precision function for the algorithm as

$$\begin{aligned} L_{\mathcal{A}_{\mathrm{CP}}}(p,n) &:= \max_{1\le i\le N_{\mathrm{P}}(n)} L_{f_i}(1-\varrho(p,n)) \\ &= \max_{1\le i\le N_{\mathrm{P}}(n)} L_{f_i}\left(1-\frac{1-p}{N_{\mathrm{E}}(n)}\right). \end{aligned}$$

Analogically, we define

$$K_{\mathcal{A}_{\mathrm{CP}}}(p,n) = \max_{1\le i\le N_{\mathrm{P}}(n)} K_{f_i}\left(1-\frac{1-p}{N_{\mathrm{E}}(n)}\right).$$

Then every arithmetic $\mathbb{F}_{L,K}$ with $L \ge L_{\mathcal{A}_{\mathrm{CP}}}(p,n)$ and $K \ge K_{\mathcal{A}_{\mathrm{CP}}}(p,n)$ has the desired property by construction. □

2.12.4 Example

We conclude this section with an example for the analysis. For comprehensibility, we consider a simple computation of the well-known convex hull in the plane[28]. First, the algorithm sorts the input points by their x-coordinate. Second, it constructs the upper hull incrementally. Third, it computes the lower hull with a similar strategy. Finally, it clues the upper and lower hull together.

The first and last step are trivial. We add further explanation to the computation of the upper hull. We insert the points from the left to the right one at a time. As soon as the construction contains at least three points, we examine the three rightmost points in their order of insertion. If they form a right turn, we proceed with the insertion of the next input point. Otherwise, we remove the middle point and repeat the test for the three rightmost points (in case there are at least three points left). The resulting sequence of points defines the upper hull. The lower hull is computed similarly.

The algorithm can be realized with two predicates: The *less-than* predicate and the *side-of-line* predicate. The sign of the less-than predicate is reliable for floating-point arithmetic, so we neglect this predicate in the analysis. The side-of-line predicate sol is introduced on Page 81.

Our aim is to analyze the algorithm in a general way: *What precision L is sufficient to guarantee a successful termination of $\mathcal{A}_{\mathrm{G}}$ with probability p in case we insert n random points from the area $[-2^{e_{\max}}, 2^{e_{\max}}]^2$ while we allow circular perturbations with radius r.*

A scalable spherical perturbation area with radius r contains an axis-parallel box with edge lengths $2\delta_i$ where $\delta_i = \frac{r}{\sqrt{2}}$ for $i \in \{1,2\}$. The ratio between the volume of

[28] We refer to Algorithm ConvexHull on Page 6 in [8].

the spherical and the squared perturbation area is bounded by

$$\eta \;=\; \left\lceil \frac{|\mathcal{U}_{\mathcal{A}_{\mathrm{CP}},r}(\bar{y})|}{|U_{\mathcal{A}_{\mathrm{CP}},\delta_1}(\bar{y})|} \right\rceil \;=\; \left\lceil \frac{\pi r^2}{(2\delta_1)^2} \right\rceil \;=\; \left\lceil \frac{\pi r^2}{2r^2} \right\rceil \;=\; 2.$$

Therefore, $\mathcal{A}_{\mathrm{CP}}$ is perturbation-suitable. This implies Step 1 of the method of distributed probability.

$\mathcal{A}_{\mathrm{CP}}$ is also predicate-suitable since we only have to consider the side-of-line predicate sol in the analysis, i.e., $N_{\mathrm{P}}(n) = 1$.

During the construction of the upper hull, we insert n points and remove at most $n-2$ points. This upper-bounds the number of sol evaluations to $2(n-2)$ since at least three points are necessary to apply sol. The same argumentation is true for the lower hull. Therefore, the total number of sol evaluations is bounded by $N_{\mathrm{E}}(n) = 4(n-2)$. Consequently, $\mathcal{A}_{\mathrm{CP}}$ is evaluation-suitable. The maximum failure probability per evaluation is then

$$\varrho(p,n) \;=\; \frac{1-p}{N_{\mathrm{E}}(n)} \;=\; \frac{1-p}{4(n-2)}$$

according to Step 2 of the method of distributed probability.

The predicate $\mathrm{sol}(u_x, u_y, v_x, v_y, q_x, q_y)$ is analyzable due to Theorem 2.13 on Page 60. We easily derive the necessary quantities from the evaluation order

$$((u_x v_y - u_x q_y) - (q_x v_y - v_x u_y)) - (v_x q_y - q_x u_y)$$

and obtain: $k = 6$, $d = 2$, $\beta^* = 2$, $\hat{\beta} = 1$, $|\mathcal{I}| = 6$, $\max_{\iota \in \mathcal{I}} |a_\iota| = 1$, and $|a_\beta| = 1$. (The choice of β does not matter here.) Therefore, the theorem provides the bound

$$\begin{aligned} L_{\mathrm{safe,sol}}(p) \;&=\; \Bigg\lceil -\beta^* \log_2 (1 - \sqrt[k]{p}) \;+ \\ &\qquad \log_2 \frac{(d + 1 + \lceil \log_2 |\mathcal{I}| \rceil) \cdot |\mathcal{I}| \cdot \max_{\iota \in \mathcal{I}} |a_\iota| \cdot 2^{d e_{\max} + \beta^* + 1} \cdot \hat{\beta}^{\beta^*}}{|a_\beta| \cdot (t\delta_1)^{\beta^*}} \Bigg\rceil \\ &=\; \left\lceil -2 \log_2 (1 - \sqrt[6]{p}) \;+\; \log_2 \frac{(2+1+3) \cdot 6 \cdot 1 \cdot 2^{2 \cdot e_{\max} + 2 + 1} \cdot 1^2}{1 \cdot \left(t \frac{r}{\sqrt{2}} \right)^2} \right\rceil \\ &=\; \left\lceil -2 \log_2 (1 - \sqrt[6]{p}) \;+\; 2 \log_2 6 + 2 e_{\max} + 4 - 2 \log tr \right\rceil . \end{aligned}$$

In addition, we obtain the bound

$$\begin{aligned} L_{\mathrm{grid,sol}}(p) \;&:=\; \left\lceil e_{\max} - 1 - \log_2 \left((1-t) \frac{\delta_1 \, (1 - \sqrt[k]{p})}{\hat{\beta}} \right) \right\rceil \\ &:=\; \left\lceil e_{\max} - 1 - \log_2 \left((1-t) \frac{r}{\sqrt{2}} (1 - \sqrt[6]{p}) \right) \right\rceil . \end{aligned}$$

Finally, we conclude from the above that $\mathcal{A}_{\mathrm{CP}}$ is analyzable and perform the remaining Step 3 of the method of distributed relations. This leads to

$$\begin{aligned}
L_{\mathcal{A}_{\mathrm{CP}}}(p,n) &= \max_{1 \le i \le N_{\mathrm{P}}(n)} L_{f_i}(1-\varrho(p,n)) \\
&= L_{\mathrm{sol}}(1-\varrho(p,n)) \\
&= \max\left\{L_{\mathrm{safe,sol}}(1-\varrho(p,n)), L_{\mathrm{grid,sol}}(1-\varrho(p,n))\right\} \\
&= \max\left\{L_{\mathrm{safe,sol}}\left(1-\frac{1-p}{4(n-2)}\right), L_{\mathrm{grid,sol}}\left(1-\frac{1-p}{4(n-2)}\right)\right\}
\end{aligned}$$

We now use the result to analyze an exemplified situation: What precision L is sufficient to guarantee a successful termination of $\mathcal{A}_{\mathrm{G}}$ with probability $p = \frac{1}{2}$ in case we insert up to $n = 10000$ random points from the square $[-1000, 1000]^2$ while we allow circular perturbations with radius $r = 8$. The range of the input values is obviously $e_{\max} = 10$. If we choose $t = \frac{1}{2}$, we calculate

$$\begin{aligned}
L_{\mathcal{A}_{\mathrm{CP}}}\left(\frac{1}{2}, 10000\right) &= \max\left\{L_{\mathrm{safe,sol}}\left(1-\frac{1-\frac{1}{2}}{4(10000-2)}\right),\right. \\
&\qquad \left. L_{\mathrm{grid,sol}}\left(1-\frac{1-\frac{1}{2}}{4(10000-2)}\right)\right\} \\
&= \max\left\{L_{\mathrm{safe,sol}}\left(\frac{79983}{79984}\right), L_{\mathrm{grid,sol}}\left(\frac{79983}{79984}\right)\right\} \\
&= \max\left\{\lceil 62.914... \rceil, \lceil 26.372... \rceil\right\} \\
&= 63
\end{aligned}$$

2.13 General Controlled Perturbation Implementations

We present a general way to implement controlled perturbation algorithms $\mathcal{A}_{\mathrm{CP}}$ to which we can apply our analysis tool box. The algorithm template is illustrated as Algorithm 2. It is important to see that all statements necessary for the controlled perturbation management are simply wrapped around the function call of $\mathcal{A}_{\mathrm{G}}$.

Algorithm 2 : $\mathcal{A}_{\mathrm{CP}}(\mathcal{A}_{\mathrm{G}}, \bar{y}, \mathcal{U}_\delta, \psi, \eta)$

```
/* initialization */
$L \leftarrow$ precision of built-in floating-point arithmetic
$K \leftarrow$ exponent bit length of built-in floating-point arithmetic
$e_{\max} \leftarrow$ determine upper bound $2^{e_{\max}}$ on $|\bar{y}_i| + \delta$

repeat
  /* run guarded algorithm */
  for $i = 1$ to $\eta$ do
    $y \leftarrow$ random point in $\overline{\mathcal{U}}_\delta(\bar{y})|_{\mathbb{G}_{L,K,e_{\max}}}$
    $\omega \leftarrow \mathcal{A}_{\mathrm{G}}(y, \mathbb{F}_{L,K})$
    if $\mathcal{A}_{\mathrm{G}}$ succeeded then
      leave the for-loop
    end if
  end for

  /* adjust parameters */
  if $\mathcal{A}_{\mathrm{G}}$ failed then
    if floating point overflow error occurred then
      /* guard failed because of range error */
      $K \leftarrow K + \psi_K$
    else
      /* guard failed because of insufficient precision */
      $L \leftarrow \lceil \psi_L \cdot L \rceil$
    end if
  end if
until $\mathcal{A}_{\mathrm{G}}$ succeeded

/* return perturbed input $y$ and result $\omega$ */
return $(y, \omega)$
```

Remember that the original perturbation area is $\overline{\mathcal{U}}_\delta(\bar{y})|_{\mathbb{G}}$. The implementation of a uniform perturbation seems to be a non-obvious task for most shapes. Therefore, we propose axis-parallel perturbation areas in applications. (For example, we can replace spherical perturbation areas with cubes that are contained in them.) Axis-parallel areas have the added advantage that the perturbation is composed of random integral numbers as we have explained in Remark 2.1.

An argument of the controlled perturbation algorithm is the tuple $\psi = (\psi_L, \psi_K) \in$

$\mathbb{R} \times \mathbb{N}$ of constants, which are used for the augmentation of L and K. The real constant $\psi_L > 1$ is used for a multiplicative augmentation of L, and the natural number ψ_K is used for an additive augmentation of K.

Using the multiplicative constant ψ_L has also a positive effect on the running time of $\mathcal{A}_{\mathrm{CP}}$. The running time of $\mathcal{A}_{\mathrm{CP}}$ is basically the sum of the running times of the series of unsuccessful runs of $\mathcal{A}_{\mathrm{G}}$ plus the running time of the first successful run of $\mathcal{A}_{\mathrm{G}}$. Summing up the times for the unsuccessful runs of $\mathcal{A}_{\mathrm{G}}$ leads to a geometric series. The total time of the unsuccessful runs is therefore a constant multiple of the running time of the successful run of $\mathcal{A}_{\mathrm{G}}$. This constant vanishes in the Landau notation.

There is a variant of Algorithm 2 that also allows the increase of perturbation parameter δ. Beginning with $\delta = \delta_{\min} \in \mathbb{R}^k_{>0}$, we augment the perturbation parameter δ by a real factor $\psi_\delta > 1$ each time we repeat the for-loop. When we leave the for-loop, we reset δ to $\delta_{\min}$. We observe that this strategy implies an upper-bound on the perturbation parameter by $\delta_{\max} := \delta_{\min} \cdot \psi_\delta^{\eta-1}$. This is the bound we use in the analysis. To keep the presentation clear, we do not express variable perturbation parameters explicitly in the code.

A variable precision floating-point arithmetic is necessary for an implementation of $\mathcal{A}_{\mathrm{CP}}$. When we increase the precision in order to evaluate complex expressions successfully, the evaluation of subsequent simple expressions may last longer than necessary. Therefore, we suggest floating-point filters as they are used in interval arithmetic. That means, we use a multi-precision arithmetic that refines the precision on demand up to the given L. If it is necessary to exceed L, $\mathcal{A}_{\mathrm{G}}$ fails. In the analysis, we use this threshold on the precision.

2.14 Perturbation Policy

The meaning of perturbation is introduced in Section 2.2.1, and its implementation is explained in Remark 2.1 on Page 23. So far, we have considered the original input to be the point $\bar{y} \in \mathbb{R}^n$, which is the concatenation of *all* coordinates of *all* input points for the geometric algorithm $\mathcal{A}_{\mathrm{CP}}$. In contrast to that, we now concern ourselves with the geometric interpretation of the input and consider it as a sequence of geometric objects $\mathcal{O}_1, \ldots, \mathcal{O}_m$. Then a perturbation of the input is the sequence of perturbed objects. In this section, we define two different perturbation policies: The *pointwise* perturbation in Section 2.14.1 and the *object-preserving* perturbation in Section 2.14.2. The latter has the property that the topology of the input object is preserved.

2.14.1 Pointwise Perturbation

For pointwise perturbations, we assume that the geometric object is given by a sequence of points. A circle in the plane, for example, is given by three points. Another example is the polygon in Figure 2.29(a), which is represented by the sequence of four vertices *abcd*. The *pointwise perturbation* of a geometric object is the sequence of individually perturbed points of its description, i.e., randomly chosen points of their neighborhoods. Figure 2.29(b) shows a pointwise perturbed polygon $a'b'c'd'$ for our example. Because the perturbations of points are independent of each other, this policy is quite easy to implement. We observe that pointwise perturbations do not preserve the structure of the input object in general: The original polygon *abcd* is simple, whereas the perturbed polygon $a'b'c'd'$ in our example is not. And, the orientation of a circle that is defined by three perturbed points may differ from the orientation of the circle that is defined by the original points. Be aware that our analysis is particularly designed for pointwise perturbations. We suggest to apply this perturbation policy to inputs that are inherently disturbed, e.g., scanned data.

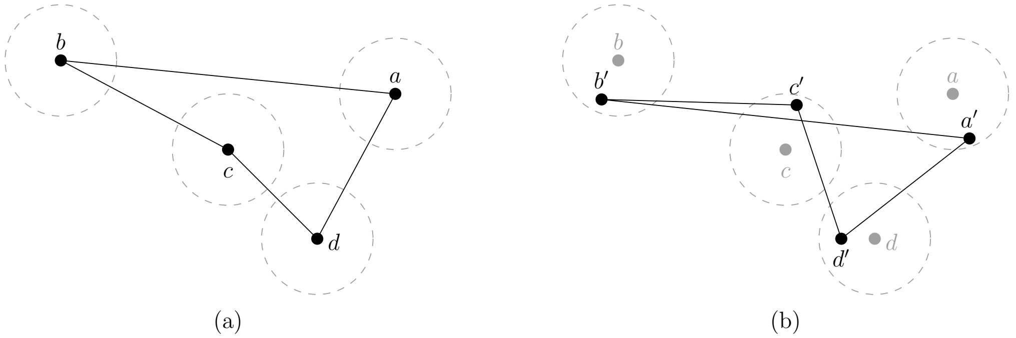
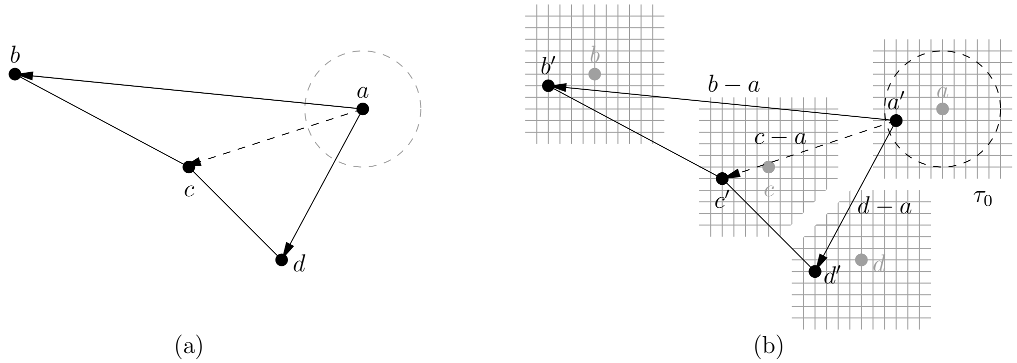


Figure 2.29: Example of a *pointwise perturbation* in the plane: (a) original input and (b) perturbed input.

2.14.2 Object-preserving Perturbation

For object-preserving perturbations, we assume that the geometric object is given by an anchor point and a sequence of fixed measurements.[29] A circle in the plane, for example, is given by a center (anchor point) and a radius (fixed measurement). Another example is the polygon $abcd$ in Figure 2.30(a), which is given by an anchor point, say a, and implicitly by the sequence of vectors (the measurements) pointing from a to b, from a to c, and from a to d. The *object-preserving perturbation* of a geometric object is a pointwise perturbation of its anchor point while maintaining all given measurements. Figure 2.30(b) shows polygon $a'b'c'd'$ that results from an object-preserving perturbation. We have $b' := a' + b - a$, etc. We observe that this perturbation is actually a translation of the object and hence preserves the structure of the input object in any respect: Its orientation, measurements and angles. The object-preserving perturbation of a circle, for example, changes its location but not its radius or orientation.



(a) (b)

Figure 2.30: Example of an *object-preserving perturbation* in the plane: (a) original input and (b) perturbed input.

The input must provide further information to support object-preserving perturbations. For the explicit representation, this policy requires a *labeling* of input values as *anchor points* (perturbable) or *measurements* (constant). For the implicit representation, the policy requires the subdivision of the input into single objects; then, we make one of these points the anchor point and derive the measurements for the remaining points. To allow the object-preserving perturbation, the implementation must offer the labeling of values or the distinction of input objects.

In this context, we are pleased to observe that our perturbation area $\overline{\mathcal{U}}_\delta(A)|_{\mathbb{G}}$ supports object-preservation because it is composed of a regular grid. *If the original object is represented without rounding errors, the perturbed object is represented exactly as well.* Of course, we can always apply object-preserving perturbations to finite-precision input objects.[30] As an example, assume that the input objects were

[29] The measurements may be given explicitly or implicitly. Both are acceptable.

[30] This is true because we can derive a sufficient grid unit from the given fixed-precision input.

created with a computer-aided design (CAD) system such that their coordinates are multiples of a certain unit. Then object-preserving perturbations are possible.

How can we analyze object-preserving perturbations? We consider the analysis of function f that realizes a predicate. For pointwise perturbations, we demand in Section 2.2.1 that f only depends on input values. For object-preserving perturbations, we only allow dependencies on anchor points: Every other point in the description of the object must be replaced in the formula by an expression that depends on the anchor point of the affected object. The dependency of the function on the variables is analyzed as before. Finally, we remark that we do not recommend perturbation policies that are based on scaling, stretching, sheering, or rotation since the perturbed input cannot be represented error free in general.

2.15 Appendix: List of Identifiers

The page numbers refer to the definitions of the identifiers. References to preliminary definitions are parenthesized.

Algorithms		**Page**
$\mathcal{A}$	the given geometric algorithm $\mathcal{A}(\bar{y})$.	-
$\mathcal{A}_{\mathrm{G}}$	the guarded version $\mathcal{A}_{\mathrm{G}}(y, \mathbb{F}_{L,K})$ of algorithm $\mathcal{A}$, i.e., all predicate evaluations are guarded.	18
$\mathcal{A}_{\mathrm{CP}}$	the controlled perturbation version $\mathcal{A}_{\mathrm{CP}}(\mathcal{A}_{\mathrm{G}}, \bar{y}, \delta, \psi)$ of algorithm $\mathcal{A}$. The implementation of $\mathcal{A}_{\mathrm{CP}}$ makes usage of $\mathcal{A}_{\mathrm{G}}$.	98

Sets and Number Systems		**Page**
$\mathbb{C}$	the set of complex numbers.	-
$\mathbb{F}_{L,K}$	1. the set of floating point numbers with radix 2 whose precision has up to L digits and whose exponent has up to K digits. 2. the floating point arithmetic that is induced this way.	17
$\mathbb{G}_{L,K,e_{\max}}$	the set of grid points. They are a certain subset of the floating point numbers $\mathbb{F}_{L,K}$ within the interval $[-2^{e_{\max}}, 2^{e_{\max}}]$.	22
$\mathbb{N}$; $\mathbb{N}_0$	the set of natural numbers; set of natural numbers including zero.	-
$\mathbb{Q}$	the set of rational numbers.	-
$\mathbb{R}$; $\mathbb{R}_{>0}$; $\mathbb{R}_{\neq 0}$	the set of real numbers; set of positive real numbers; set of real numbers excluding zero.	-
$\mathbb{Z}$	the set of integer numbers.	-
$X\|_{\mathbb{F}_{L,K}}$	the restriction of a set X to points in $\mathbb{F}_{L,K}$.	17
$X\|_{\mathbb{G}_{L,K,e_{\max}}}$	the restriction of a set X to points in $\mathbb{G}_{L,K,e_{\max}}$.	22

Identifiers of the Analysis		**Page**
A	the set of valid projected arguments $\bar{x}$ for f.	20
$B_E(L)$	a floating point error bound on the arithmetic expression E.	80
$C_f(\cdot)$	the critical set of f.	(27), 85
$\mathcal{G}_f$	a guard for f on the domain X.	17
K	the bit length of the exponent (see $\mathbb{F}_{L,K}$).	16

$K_f(p)$	a lower bound on the bit length of the exponent.	86
L	the bit length of the precision (see $\mathbb{F}_{L,K}$).	16
$L_{\mathcal{A}_{\text{CP}}}(p,n)$	the precision function of $\mathcal{A}_{\text{CP}}$.	89
$L_f(p)$	the precision function of f.	45
L_{grid}	a bound on the precision; caused by the grid unit condition.	(30), 45
L_{safe}	a bound on the precision; caused by the region- and safety-condition.	45
$N_{\text{E}}(n)$	an upper-bound on the number of predicate evaluations.	90
$N_{\text{P}}(n)$	an upper-bound on the number of different predicates.	90
$R_{f,\gamma}(\cdot)$	the region of uncertainty of f.	28
$R_{f,\text{aug}(\gamma)}(\cdot)$	the augmented region of uncertainty of f.	29
$S_{\inf f}(L)$	the lower fp-safety bound.	27, (85)
$S_{\sup f}(K)$	the upper fp-safety bound.	86
$U_{f,\delta}(\cdot)$	the perturbation area of f; its shape is an axis-parallel box.	20
$U_{\mathcal{A}_{\text{CP}},\delta}(\cdot)$	the perturbation area of $\mathcal{A}_{\text{CP}}$; its shape is an axis-parallel box.	90
$\mathcal{U}_{\mathcal{A}_{\text{CP}},\delta}(\cdot)$	the perturbation area of $\mathcal{A}_{\text{CP}}$; it may have any shape.	90
$e_{\max}$	the input value parameter (see Formula (2.1)).	20
f	the real-valued function $f : \bar{U}_\delta(A) \rightarrow \mathbb{R}$ under consideration. We assume that the sign of f decides a geometric predicate.	20
k	the arity of f.	20
n	the size of input $\bar{y}$.	19
$p_f(L,K)$	the probability function of f.	(47), 86
$p_{\text{grid}}(L)$	a bound on the probability; caused by the grid unit condition.	47
$p_{\inf}(L)$	a bound on the probability; caused by the region- and inf-safety-condition.	47
$p_{\sup}(K)$	a bound on the probability; caused by the sup-safety-condition.	86
$\text{pr}(f\vert_{\mathbb{G}})$	the least probability that a guarded evaluation of f is successful for inputs in $\mathbb{G}$ under the arithmetic $\mathbb{F}$.	23
$\frac{1}{t}$	the augmentation factor for the region of uncertainty.	30
$\bar{x}$	the arguments of f; projection of $\bar{y}$.	20

x	the perturbed arguments of f; projection of y.	20
$\bar{y}$	the original input to the algorithm.	19
y	the perturbed input $y \in U_\delta(\bar{y})$.	20
δ	the perturbation parameter which bounds the maximum amount of perturbation componentwise.	19
γ	the tuple of componentwise distances to the critical set.	29
Γ	the set of valid augmented γ.	29
Γ-box	like Γ; the set is an axis parallel box.	29
Γ-line	like Γ; the set is a line.	29
$\nu_f(\gamma)$	an upper-bound on the volume of $R_{f,\gamma}$.	37
τ	the grid unit.	22
$\varphi_{\inf f}(\gamma)$	a lower-bound on the absolute value of f outside of $R_{f,\gamma}$.	39, (85)
$\varphi_{\sup f}(\gamma)$	an upper-bound on the absolute value of f outside of $R_{f,\gamma}$.	85
$\chi_f(\gamma)$	a lower-bound on the complement of ν_f within the perturbation area.	38
ψ	the tuple $\psi = (\psi_L, \psi_K) \in \mathbb{R} \times \mathbb{N}$ is used for the augmentation of L and K.	98

Miscellaneous		**Page**
$\mu(\cdot)$	the Lebesgue measure.	21
$\pi(\cdot)$	the projection of points and sets, e.g., π_i, $\pi_{<i}$, $\pi_{>i}$, $\pi_{\neq i}$.	64
$\prec$	the reverse lexicographic order.	57
$\prec_\sigma$	the reverse lexicographic order after the permutation of the operands.	57

3 Complexity and Computation of Θ-Guarded Regions

In this chapter, we define and consider a planar guarding problem. We define the Θ-guarded region in Section 3.1. We realize that the difficulty in the study of Θ-guarded regions is that its shape and complexity varies with the angle Θ. We discuss the case $\Theta \geq \pi$ in Section 3.2. We analyze the shape of the Θ-guarded region, show its general relation to the convex hull, prove that its complexity equals the complexity of the convex hull, show its specific relation to positive α-hulls for certain guard sets and angle ranges, and develop an easy and efficient $O(n \log n)$ time algorithm to compute its boundary.

In the main part, in Section 3.3, we consider the case $\Theta < \pi$. The problem for these angles becomes much more involved, and the boundary of the Θ-region becomes more difficult to understand. We show in Section 3.3.1 that the boundary of the Θ-region is contained in an arrangement of circular arcs. In Section 3.3.2, we bound this set of circular arcs by $O(\frac{n}{\Theta})$. In case $\frac{\pi}{2} \leq \Theta < \pi$, we prove that the complexity of the Θ-region is $O(n)$. Even more, for smaller angles $\delta \leq \Theta < \frac{\pi}{2}$ where δ is a positive constant, we show that the complexity is $O(n^{1+\varepsilon})$ for any $\varepsilon > 0$. In case we consider the asymptotic complexity bound in n and $\frac{1}{\Theta}$, we prove $O(\frac{n^2}{\Theta^2})$. Furthermore, in Section 3.3.3, we construct a generic example for the latter case, which has complexity $\Omega(n^2)$. In Section 3.3.4, we present an algorithm that computes the Θ-region for $\Theta < \pi$. For the case $\frac{\pi}{2} \leq \Theta < \pi$, we further present an improved algorithm that takes $O(n \log n)$ time.

We summarize all complexity bounds in Section 3.4.

3.1 The Θ-Guarded Region

We are given a finite point set G in the plane, which we call *guards*, and an angle $\Theta \in [0, 2\pi]$. A Θ-cone is a cone with apex angle Θ. We say that the Θ-cone is *empty (with respect to G)* if it does not contain any point of G in its interior. A point $p \in \mathbb{R}^2$ is *Θ-guarded (with respect to G)* if every Θ-cone whose apex is located at p is non-empty. Furthermore, the set of all Θ-guarded points is called the *Θ-guarded region*, or the *Θ-region* for short. The Θ-region is defined uniquely by the pair (G, Θ). The motivation for this kind of guarding is that a point is well-guarded only if it is guarded from "all" directions. We consider Θ-cones as open sets; hence, the Θ-region is an open set, too. We always assume that G is non-empty. (If there are no guards, there is trivially no Θ-region.)

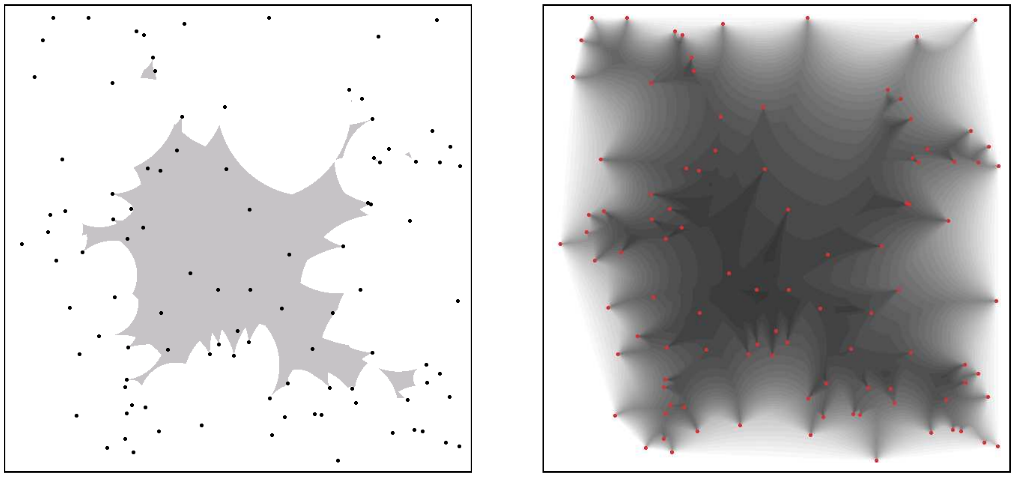


Figure 3.1: An example with $|G| = 50$. The Θ-region is not necessarily connected for $0 < \Theta < \pi$ (left). The isolines of function f show how components of the Θ-region disconnect for decreasing Θ in this example (right).

An example of a Θ-region is shown in the left picture of Figure 3.1. We remark that the computer generated pictures are based on the value of the continuous function $f : \mathbb{R}^2 \setminus G \to (0, 2\pi]$ where $p \mapsto \max\{\Theta : \exists \text{ empty } \Theta\text{-cone with apex } p\}$. The left picture of Figure 3.1 and the two rightmost pictures in Figure 3.7 are generated by plotting a grid point shaded if and only if f has a value below the threshold Θ. In the right picture of Figure 3.1, we have mapped different intervals of function values in $[0, \pi]$ to different gray scale values to visualize isolines (along the boundary of the gray scale value) of f in this example. Although these pictures visualize only function values at grid points, we may rely on the pictures since we deal with cones of a certain angle and not with arbitrarily thin stripes that could somehow pass between grid points.

We make several observations. A point $p \in \mathbb{R}^2$ does not belong to the Θ-region if there is an empty Θ-cone with apex p. Hence, no point inside an empty cone can

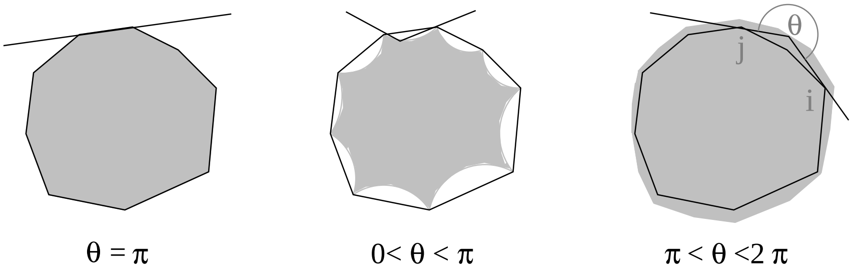


Figure 3.2: For $\Theta < \pi$ (resp. $\Theta > \pi$) the region lies inside (outside) the convex hull $CH(G)$ and the bounding arcs are bend inside (outside) the region.

belong to the region, and hence, the region cannot contain holes. A point p lies on the boundary of a Θ-region if there is at least one empty (open) Θ-cone with apex p and if the closure[1] of each Θ-cone with apex p is non-empty. The example in Figure 3.1 shows that the Θ-region is not necessarily connected for $0 < \Theta < \pi$. The shape of the Θ-region is invariant under translation, rotation, and scaling of G.

The shapes of all Θ-regions can be classified according to Θ. The boundary of the π-region is simply the *convex hull* $CH(G)$ because the intersection of all half-planes containing G (convex hull) is the same as removing every half-plane from $\mathbb{R}^2$ that does not contain any point of G (π-region). However, for $0 < \Theta < \pi$, empty (convex) Θ-cones can enter the convex hull through the edges, while for $\pi < \Theta < 2\pi$ the apexes of empty (concave) Θ-cones do not even have to touch the convex hull (see Figure 3.2). Therefore, the Θ-region is connected if $\Theta \geq \pi$. The 2π-region is obviously the plane $\mathbb{R}^2$, and the 0-region is the empty set.

Before we discuss the Θ-region for $0 < \Theta < \pi$, we discuss the simpler case $\pi < \Theta < 2\pi$ in the next section. Throughout the chapter, we use the following property of the *Inscribed Angle Theorem* (see Euklid [27, Book III, Proposition 20]): *Given a circular arc $C_{\ell,r}$ from ℓ to r, then $\angle lpr = \angle lqr$ holds for all $p, q \in C_{\ell,r}$.* We write $C^{\alpha}_{\ell,r}$ if the inscribed angle is α. The arc end points are always given in counter-clockwise order. Without loss of generality, the vertices of the convex hull are given in clockwise order.

[1] By way of exception, we consider closed cones here.

3.2 Boundary, Complexity and Computation for $\Theta \geq \pi$

We know that the π-region is the standard convex hull $CH(G)$. We remark that there are plenty of algorithms which compute $CH(G)$, for example, using the plane-sweep technique [8], the divide-and-conquer technique [69], or Graham's scan [18]. All these algorithms take $O(n \log n)$ time.

Further, we know that every point in the interior of the convex hull of G is Θ-guarded for $\Theta > \pi$ (see Figure 3.2). Intuitively, the boundary of the Θ-region is drawn by the apex of an empty Θ-cone, which is rotated around the convex hull $CH(G)$ such that its rays are always tangent to $CH(G)$. The following algorithm computes the boundary of the Θ-region in this case.

First of all, we compute the clockwise sequence of guards $G' = \{g_1, \ldots, g_k\}$ defining the convex hull (see for example [69]). Next, we construct an algorithm that outputs the circular arcs that define the boundary of the Θ-region. Therefore, we want to identify all pairs $(g_i, g_j) \in G' \times G'$ with $g_i \neq g_j$ for which there exists an empty Θ-cone that is tangent to g_i and g_j and whose apex lies outside of the convex hull. We say that the apex of the Θ-cone "sees" the polygonal chain of $CH(G)$ from g_i to g_j. The pairs (g_i, g_j) have the following property: The lines supporting the convex hull edges $(g_j, g_{\text{succ}(j)})$ and $(g_{\text{pred}(i)}, g_i)$ form an angle of intersection not less than Θ, and the lines supporting $(g_i, g_{\text{succ}(i)})$ and $(g_{\text{pred}(j)}, g_j)$ form an angle of intersection less than Θ. The sequence of all these pairs (g_i, g_j) and the corresponding circular arcs $C^{2\pi-\Theta}_{g_j,g_i}$ are computed by a cyclic scan over the sequence G'. The arc end points of the Θ-region boundary are computed as the intersection points of the circular arc $C^{2\pi-\Theta}_{g_j,g_i}$ with the supporting lines through $(g_j, g_{\text{succ}(j)})$ and $(g_{\text{pred}(i)}, g_i)$. Consequently, the Θ-region has at most the complexity of the convex hull. (The left picture of Figure 3.3 shows an example with smaller complexity.) The running time of the algorithm is dominated by the convex hull construction in $O(n \log n)$ time. We summarize the discussion.

Lemma 3.1. *For $\Theta > \pi$, the boundary of the Θ-guarded region can be computed in $O(n \log n)$ time, and its complexity is $O(|CH(G)|)$.*

Obviously, the Θ-region is a generalization of the convex hull. The constructive algorithm does not only lead to a precise description of the Θ-region for $\Theta > \pi$. It also proves that the complexity of the Θ-region is bounded by the complexity of the convex hull $CH(G)$ in this case.

We next show situations in which the Θ-region is equivalent to the positive α-hull. For angles Θ, which are slightly larger than π, the region resembles the hull with outward bent edges. The following example illustrates this point of view.

Lemma 3.2. *Let G be the vertices of a regular n-sided polygon for $n \geq 2$. Then the $(\frac{n+1}{n}\pi)$-region of G is the disc that contains G in its boundary.*

Proof. Let e be the center of the regular n-sided polygon, and let b and d be two adjacent vertices of the polygon. The situation is illustrated in the left picture of

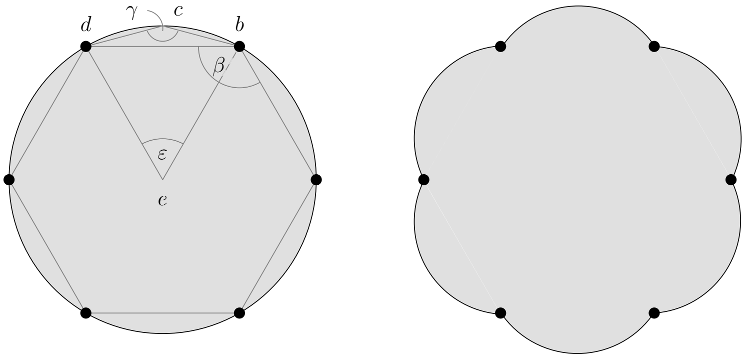


Figure 3.3: The $(\frac{n+1}{n}\pi)$-region of a regular n-sided polygon is a disc (left). The $(\frac{n+2}{n}\pi)$-region of a regular n-sided polygon (right).

Figure 3.3. By ε we denote the angle $\angle deb$. Since $|G| = n$, the angle ε is $\frac{2\pi}{n}$. We set $\varepsilon' := 2\pi - \varepsilon = \frac{n-1}{n}2\pi$. By β we denote the inner angle of the regular polygon at b. We have $\beta := \pi - \varepsilon = \frac{n-2}{n}\pi$. We set $\beta' := 2\pi - \beta = \frac{n+2}{n}\pi$. We now consider the point c on the circle that passes through G and lies immediately between b and d. By γ we denote the angle $\angle bcd$. Due to the Inscribed Angle Theorem (see Euklid [27]), we have $\gamma := \frac{1}{2}\varepsilon' = \frac{n-1}{n}\pi$. We set $\gamma' := 2\pi - \gamma = \frac{n+1}{n}\pi$.

We next consider the $\frac{n+1}{n}\pi$-region of G, that means, we set $\Theta := \gamma'$. Because the outer angle β' of the regular n-sided polygon at b is larger than Θ, b is unguarded. Consequently, no guard in G is Θ-guarded. Therefore, the points in G contribute to the boundary of the $\frac{n+1}{n}\pi$-region.

Furthermore, we have determined Θ such that the arc from b to d lies on the circle through G. Because the polygon is regular, this argument is true for all arcs. It follows that the $\frac{n+1}{n}\pi$-region is the disc that contains G in its boundary. □

Positive α-hulls are defined by Edelsbrunner et al. [22] as follows: Let G be a point set in the plane, and let α be a positive real. Then the positive α-hull is the intersection of all discs of radius $\frac{1}{\alpha}$ that contain the point set G.

Lemma 3.3. *Let G be the vertices of a regular n-sided polygon for $n \geq 2$, and let $\pi < \Theta \leq \frac{n+1}{n}\pi$. Then there is a positive α such that the Θ-region of G is equivalent to the positive α-hull.*

Proof. The claim follows from Lemma 3.2 and the definition of positive α-hulls. □

Be aware that there is no way to derive the radius $\frac{1}{\alpha}$ from Θ directly, without inspecting the coordinates of the guards.

To improve our understanding of Θ-regions for $\Theta \geq \pi$, we slightly extend the example and derive further statements from Lemma 3.2, which are valid for the vertex set G of a regular n-sided polygon. First, the arcs in the boundary of the Θ-region of G are incident to the same guards as the respective hull edges for $\pi \leq \Theta \leq \frac{n+2}{n}\pi$ (see the right picture of Figure 3.3). Second, the set G of guards is detached from the Θ-region for angles $\Theta > \frac{n+2}{n}\pi$. Third, the region is convex for $\pi \leq \Theta \leq \frac{n+1}{n}\pi$ and non-convex for $\Theta > \frac{n+1}{n}\pi$.

3.3 Boundary, Complexity and Computation for $\Theta < \pi$

We assume in this section that the angle is $0 < \Theta < \pi$. We describe the boundary of the Θ-region in Section 3.3.1, consider an upper-bound on its worst-case complexity in Section 3.3.2, prove a lower-bound on the worst-case complexity in Section 3.3.3, and present an algorithm to compute the Θ-region in Section 3.3.4.

3.3.1 Boundary

We give a mathematical description of the Θ-region below. First, we come back to the inscribed angles and explain its meaning for our setting. Let $e = (\ell, r) \in G \times G$ be any pair of guards. Then the set of points where we can place the apex of an empty Θ-cone passing through the line segment (ℓ, r) in the same direction is bounded by the circular arc, which is incident to ℓ and r with inscribed angles Θ, and the line segment lr. We denote this closed circular segment by $D^{\Theta}_{\ell,r}$ (or D_e for short) and its bounding circular arc by $C^{\Theta}_{\ell,r}$ (or C_e for short). Because of the orientation, the circular segment is described uniquely.

The construction of the Θ-region is motivated by the idea of locally removing sets T_i of unguarded points from the convex hull $CH(G)$ such that the remaining part matches the Θ-region (see Figure 3.2, middle), that means, we set

$$\Theta\text{-region} \quad := \quad CH(G) \setminus \left(\bigcup_{i \in I} T_i \right) \qquad (3.1)$$

for specific sets T_i. We now give the construction for these sets T_i. Consider any empty Θ-cone c that has at least one guard on each ray (see the left picture of Figure 3.4). First, we turn the cone clockwise while pushing the cone towards the point set, such that it always stays empty but touches a guard on each boundary (see the middle picture of Figure 3.4). We end this motion when the apex of the cone reaches the position of a guard, say ℓ_0. Afterwards, we relocate the cone in the original position and rotate the cone in a similar way counter-clockwise until the apex reaches the position of another guard, say r_0.

To describe the construction formally, we extend our notion. By L_i (resp. R_i) we denote the set of guards that are incident to the left (resp. right) ray of a cone during the construction (the white points in the right picture of Figure 3.4). We call the closure of the union of all cones, which are used during the construction, the *tunnel T_i with respect to L_i and R_i*, or tunnel for short (the shaded region in the right picture of Figure 3.4). Note that the index set I in Formula (3.1) summarizes the tunnels.

We observe that Formula (3.1) describes the Θ-region of G because each empty Θ-cone that intersects $CH(G)$ lies in at least one tunnel T_i: Let c be such a cone. We identify a tunnel by moving c in the direction of its medial axis until one of its rays is tangent to a guard. Then we let the empty cone slide along that point without rotation until the second ray is also tangent to a guard. According to our

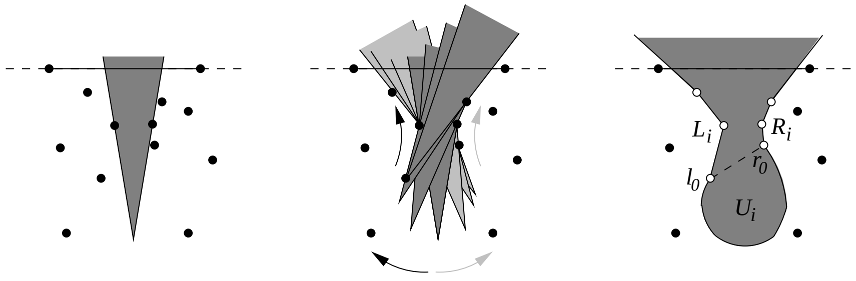


Figure 3.4: The construction of a tunnel.

construction, there is a tunnel that contains this cone and hence the cone c in its original position.

No point in T_i is Θ-guarded. Therefore, only its boundary possibly contributes to the boundary of the Θ-region. First, we consider its straight-line boundaries. Since $\Theta < \pi$, each point of a straight-line boundary can be crossed (at least) infinitesimally by an empty Θ-cone. That means, there are open neighborhoods of unguarded points around each point of a straight-line boundary, and hence, they cannot contribute to the Θ-region boundary. Due to our definition of a tunnel, points beyond the straight-line boundaries belong to a different tunnel; such points are processed independently of T_i.

Therefore, we only have to consider the curved boundary of T_i. Observe that during the construction, the apex of the rotating cone draws a *sequence of circular arcs* between ℓ_0 and r_0, which we will formalize next. We define the set

$$U_i \quad := \quad \bigcap_{(\ell,r)\in L_i\times R_i} D^{\Theta}_{\ell,r} \tag{3.2}$$

as the intersection of all circular segments for guard pairs in $L_i \times R_i$. In the following Lemma, we state how to derive the curved boundary of T_i from these circular segments. Let h_i be the closed half plane that is bounded by the line through ℓ_0 and r_0 and contains the sequence of arcs.

Lemma 3.4. *Let T_i, U_i and h_i be as defined above. Then $T_i \cap h_i = U_i$.*

Proof. (Superset.) Let $p \in U_i$. Assume there is no empty cone with apex p through tunnel T_i. This means, there is at least one pair $(\ell, r) \in L_i \times R_i$ with the property that $\angle \ell p r < \Theta$. Hence, $p \notin D^{\Theta}_{\ell,r}$, which is a contradiction to the construction of U_i.

(Subset.) Let p be the apex of an empty Θ-cone through tunnel T_i. This means that $\angle \ell p r \geq \Theta$ for all $(\ell, r) \in L_i \times R_i$, and hence, p lies in all corresponding circular segments $D^{\Theta}_{\ell,r}$. □

It follows that the Θ-region boundary is contained in the curved boundary of the union of the sets U_i, that means,

$$\partial\Theta\text{-region} \quad \subseteq \quad \partial \bigcup_{i\in I} U_i \quad = \quad \partial \bigcup_{i\in I} \left(\bigcap_{(\ell,r)\in L_i\times R_i} D^{\Theta}_{\ell,r} \right), \tag{3.3}$$

where I enumerates the tunnels. We observe that the intersections of $|L_i| \cdot |R_i|$ circular segments $D_{\ell,r}$ in Formulas (3.2) and (3.3) are too pessimistic. During the construction of a tunnel, we collect all guard pairs $(\ell, r) \subset L_i \times R_i$ that are simultaneously incident to the rotating cone in E_i. Since the touching point of L_i (resp. R_i) can only change in one direction to its neighbor in the sequence of L_i (resp. R_i), the size of E_i is given by $|L_i| + |R_i| - 1$. Therefore, we may reduce the intersection of the circular segments in Formula (3.2) to

$$U_i \quad := \quad \bigcap_{(\ell,r)\in E_i} D_{\ell,r} \cap h_i, \tag{3.4}$$

for which Lemma 3.4 is still valid. By $\mathcal{C}$ we denote the *union of all circular arcs* that appear in the boundary of U_i for $i \in I$.

3.3.2 Upper Bounds on the Worst-Case Complexity

We now discuss the worst-case complexity of the Θ-region in dependence on the number n of guards and the angle Θ. During the analysis of the complexity, we distinguish different cases for Θ. The case $\Theta \geq \pi$ is discussed in Section 3.2, and the 0-region is trivially the empty set. So, in this section, we consider the case $0 < \Theta < \pi$. Because the set of guards G is a discrete set, there is a positive Θ for every given G such that the Θ-region is also empty.

Lemma 3.5. *The Θ-region is the empty set for $\Theta \leq \frac{2\pi}{n}$.*

Proof. Consider the n rays emanating from a point $p \in \mathbb{R}^2 \setminus G$ through the guards in G. Then the rays form at least one empty cone with angle of at least $\frac{2\pi}{n}$, which contains an empty Θ-cone. Hence, p is unguarded. A similar argumentation for a guard $p \in G$ proves the existence of a cone with angle of at least $\frac{2\pi}{n-1}$. □

According to the right term in Formula (3.3), the complexity of the Θ-region is hidden in an arrangement of circular arcs. Since there are at most $O(n^2)$ different circular arcs, two for each guard pair, the complexity of the Θ-region is trivially $O(n^4)$. We show that the set $\mathcal{C}$ of circular arcs is of $O(\frac{n}{\Theta})$ size. Therefore, the complexity of the Θ-region is $O(\frac{n^2}{\Theta^2})$.

Theorem 3.6. *The set $\mathcal{C}$ of circular arcs, which defines the boundary of the Θ-region, has size $O(\frac{n}{\Theta})$.*

Proof. Instead of counting the arcs directly, we count their end points. Let p be an arc end point of a tunnel as shown in the left picture of Figure 3.5. In this position, a ray of the rotating cone is incident to two guards at once while the other ray is incident to at least one guard. (Note that *three* guards are necessary to define an arc end point.) We assume without loss of generality that two guards lie on the left ray. We focus on the guard ℓ_k that is closer to the apex and count how often *this guard* can be in a situation like this. On the one hand, there are at most $n - 1$

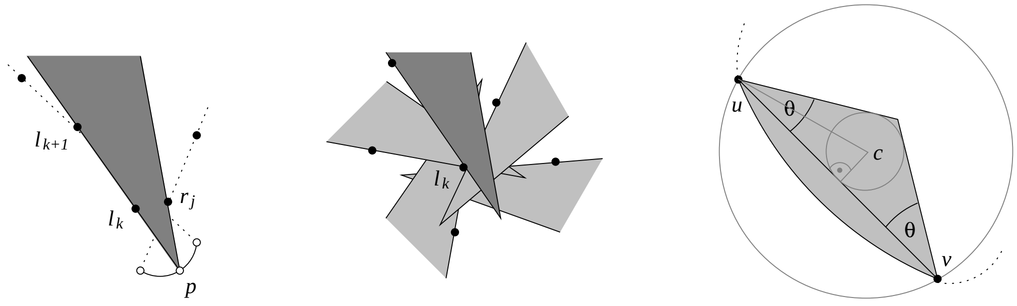


Figure 3.5: An end point p of two circular arcs in the boundary of the tunnel (left). A situation that is described in the proof of Theorem 3.6 (middle). The auxiliary construction that is described in the proof of Theorem 3.7 (right).

different cones because there are only $n-1$ different guards serving as the second guard on the left ray. On the other hand, we observe that the empty Θ-cones in this situation cannot intersect each other beyond the second guard on the left ray (see Figure 3.5, middle). Hence, there can be at most $\lfloor \frac{2\pi}{\Theta} \rfloor$ different cones in this special position. With the same argumentation for the right ray, we bound the number of arc end points per guard by $2\lfloor \frac{2\pi}{\Theta} \rfloor$. Therefore, the total number of arc end points is $O(\frac{n}{\Theta})$. □

We derive from the last Theorem that if the angle $\Theta > \delta$ is bounded by a constant $\delta > 0$, then the number of arcs in $\mathcal{C}$ is $O(n)$, and the complexity of the Θ-region is $O(n^2)$. Using an auxiliary construction, we improve this result even further.

Theorem 3.7. *Let $\delta > 0$ be a positive constant. If the angle $\Theta \in [\delta, \frac{\pi}{2})$, the complexity of the Θ-region is $O(n^{1+\varepsilon})$ for any $\varepsilon > 0$.*

Proof. We make use of the following construction. Let $a \in \mathcal{C}$ be an arc in the boundary of tunnel T_i, and let u and v be the end points of a. The line segment (u, v) and a are the boundary of a circular segment, say d_a. We next clue a triangle t_a at the edge (u, v) of d_a, which has an angle of $\min\{\Theta, \frac{\pi}{4}\}$ at u and v, and denote this new object by $F_a := d_a \cup t_a$ (see the right picture of Figure 3.5). We state and prove the following lemma: "The triangle t_a is a subset of T_i." Assume that t_a contains a guard $g \in G$. Then the angles $\angle guv$ and $\angle gvu$ are smaller than Θ. Two empty Θ-cones with apexes u and v must, therefore, belong to different tunnels. This is a contradiction to $a \subset \partial T_i$. As a consequence, t_a cannot contain any guards. Moreover, the union of empty Θ-cones with apexes at points in a covers t_a because of the angle at u and v. This completes the proof of the lemma.

We slightly change the construction above and collect the new objects F_a for all $a \in \mathcal{C}$ in the set $\mathcal{F}$. We repeat the definition of α-fatness from Efrat et al. [24]: *An object F is* α-fat *for some fixed $\alpha > 1$, if there exist two concentric disks $\mathcal{D} \subseteq F \subseteq \mathcal{D}'$ such that the ratio $\frac{\rho'}{\rho}$ between radius ρ' of $\mathcal{D}'$ and radius ρ of $\mathcal{D}$ is at most α.* We state that there is an $\alpha > 1$ such that the objects $F_a \in \mathcal{F}$ are α-fat. The worst-case scenario occurs when the arc a is almost a straight-line. Therefore, we concentrate on the proof that the triangle t_a is α-fat (see the right picture of Figure 3.5). Remember that the angle at u is $\min\{\Theta, \frac{\pi}{4}\}$. Then the ratio between the radius of the circumcircle and the radius of the inscribed circle is bounded by the constant α where[2]

$$\frac{\rho'}{\rho} \quad = \quad \frac{1}{\sin(\frac{1}{2} \cdot \min\{\Theta, \frac{\pi}{4}\})} \quad \leq \quad \frac{1}{\sin(\frac{1}{2} \cdot \min\{\delta, \frac{\pi}{4}\})} \quad =: \quad \alpha.$$

The main Theorem in Efrat et al. [24] states: The combinatorial complexity of the union of a collection $\mathcal{F}$ of α-fat objects, whose boundaries intersect pairwise in at most s points, is $O(|\mathcal{F}|^{1+\varepsilon})$ for any $\varepsilon > 0$; the constant of proportionality depends on ε, α, and s.

We have seen that α is a constant and that the objects in $\mathcal{F}$ are α-fat. The boundary of each convex object $F_a \in \mathcal{F}$ has three edges: two line segments and a circular arc. Therefore, the boundaries of each pair of objects in $\mathcal{F}$ intersect in at most $s = 10$ points (each line segment has at most 2 points of intersection, the arc has at most 6 points of intersection). Because Θ is bounded from below by the constant δ, we have $|\mathcal{C}| \in O(n)$, and consequently, $|\mathcal{F}| \in O(n)$. Therefore, the construction fulfills all preconditions to apply the Theorem of Efrat et al., which completes the proof. □

Our next result improves the worst-case complexity in the case $\frac{\pi}{2} \leq \Theta < \pi$.

Theorem 3.8. *The complexity of the Θ-region is $O(n)$ for $\frac{\pi}{2} \leq \Theta < \pi$.*

Proof. We use the following result of Kedem et al. [49]: Let $\mathcal{J}$ be a set of m Jordan curves, i.e., simply-closed curves. If any two curves in $\mathcal{J}$ intersect in at most two points, the complexity of their union is $O(m)$.

Remember the definition of U_i in Formula (3.4). We construct a Jordan curve J_i for each set U_i. Let J_i be the curved boundary of U_i from ℓ_{i_0} to r_{i_0} (i.e., the sequence of arcs), which is glued to the auxiliary half circle $C^{\frac{\pi}{2}}_{r_{i_0},\ell_{i_0}}$, that means,

$$J_i \quad := \quad \partial\left(U_i \,\cup\, D^{\frac{\pi}{2}}_{r_{i_0},\ell_{i_0}}\right).$$

We observe that the auxiliary half-circle lies in T_i because Θ is obtuse. We further observe that J_i is the boundary of a convex region and that J_i lies inside of the disc $D^{\frac{\pi}{2}}_{r_{i_0},\ell_{i_0}} \cup D^{\frac{\pi}{2}}_{\ell_{i_0},r_{i_0}}$.

We repeat the construction of Jordan curves J_i for all tunnels T_i with $i \in I$. By $\mathcal{J}$ we denote the total set of curves J_i. We prove the auxiliary statement: "Any two curves in $\mathcal{J}$ intersect at most twice." Assume that there are two curves $J_i, J_j \in \mathcal{J}$ that intersect at more than two points. We distinguish the cases that are shown in Figure 3.6. Let A_i (resp. A_j) denote the sequence of circular arcs of U_i (resp. U_j).

[2] Trigonometry: In a right-angled triangle, sine equals the quotient of the hypotenuse and the opposite side (see Papula [68]).

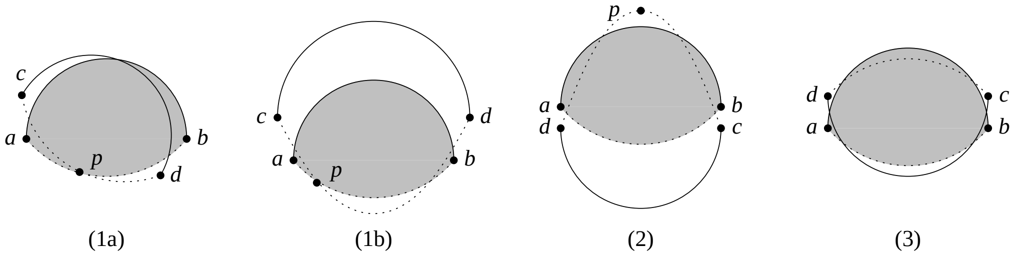


Figure 3.6: All cases that possibly imply more than just two intersection points between the curves J_i and J_j in the proof of Theorem 3.8. Dotted curves illustrate the sequence of arcs of U_i and U_j from a to b and from c to d, respectively. Solid arcs illustrate the auxiliary half-circles.

In Case 1, the sequence of circular arcs A_i and A_j intersect in point p as is shown in Pictures (1a) or (1b). That means, for each tunnel T_i and T_j, there exists an empty Θ-cone with apex in p. Therefore, the angle $\angle apb$ has to be at least 2Θ which is at least π. This is a geometrical contradiction.

In Case 2, we consider a point p that lies on the sequence of arcs A_j outside J_i as shown in Picture (2). The angle $\angle cpd$ is at least Θ. By construction, the angle $\angle bpa$ is larger than $\angle cpd$, and hence, $\angle apb > \Theta$. It is a contradiction that p does not lie inside J_i.

In Case 3, we consider an empty Θ-cone with apex c through tunnel T_j. Assume this cone passes between a and b. Then the angle $\angle bca$ is at least Θ, and hence, c must lie inside J_i. This is a contradiction. The cases where the empty Θ-cone with apex c does not pass between a and b, but between b and c, or between a and d, lead to similar geometric contradictions.

Further cases are excluded because no guard can lie inside of J_i or J_j. Because the assumption of more than two points of intersection leads to a contradiction in all cases, the auxiliary statement is true. Therefore, we can apply the mentioned result of Kedem et al. to conclude that the right side in

$$\Theta\text{-region} \quad \subset \quad \partial \bigcup_{i \in I} \left(U_i \ \cup \ D^{\frac{\pi}{2}}_{r_{i_0}, \ell_{i_0}} \right)$$

has complexity $O(n)$. This completes the proof. □

We have seen that the worst-case complexity of the Θ-region is linear for angles $\frac{\pi}{2} \leq \Theta < \pi$. Because the region does not have to be connected in this case, we have implicitly proven that the number of connected components is at most $O(n)$.

Example 3.1. Let G be the set of vertices of a regular 5-sided polygon. We reuse the definition of α in the proof of Lemma 3.2. Here, $\alpha = \frac{2\pi}{5} < \frac{\pi}{2}$. The center of the 5-sided polygon is Θ-guarded for angles $\frac{\pi}{2} \leq \Theta < \pi$. Be aware that this fact is independent of the volume of the polygon used in the construction; the statement is true for any *arbitrarily small* regular polygon with at least 5 sides.

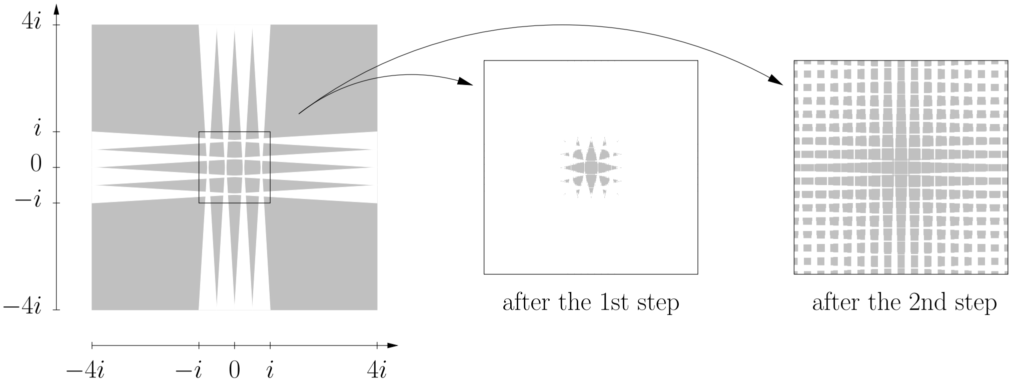


Figure 3.7: We generate the highly fragmented Θ_i-region for G_i *at the center* of the point set. The left picture shows the simplified Θ_2-region. The middle picture shows the center of the Θ_8-region *after the first step* of the construction. The right picture shows the center of the Θ_8-region *after the second step* of the construction.

Let $n \in \mathbb{N}$. We construct $\lfloor \frac{n}{5} \rfloor$ regular (at least) 5-sided polygons next to each other such that their center points are disjoint and lie on a straight line. The vertices of the polygons mark the location of the guards. Because the angle $\frac{\pi}{2} \leq \Theta < \pi$ is given, we can scale the polygons down until the Θ-region disconnects between the polygons. In this way, we construct a set G of size n that has a Θ-region with $O(n)$ components. ○

3.3.3 Lower Bound on the Worst-Case Complexity

We prove that there is a sequence of inputs such that the asymptotic bound on the complexity of the corresponding Θ-guarded regions is $\Omega(n^2)$. For this purpose, we develop a generic construction for point sets G_i with n_i guards and angles Θ_i for all $i \in \mathbb{N}$ with the property: The complexity of the Θ_i-region of the point set G_i is lower bounded by $c \cdot n_i^2$ for some constant c and $\lim_{i\to\infty} n_i = \infty$. In fact, n_i is a linear function in i, and Θ_i is of order $\frac{1}{i}$. Therefore, the complexity bound can also be interpreted as $\Omega(\frac{n}{\Theta})$.

First, we motivate the construction for a given $i \in \mathbb{N}$. The main aim is to construct the point set G_i such that the Θ_i-region is fragmented into $c \cdot n_i^2$ connected components *at the center* of the point set where the complexity of each component is constant. The left picture of Figure 3.7 illustrates the desired fragmentation. We plan to force this decomposition by long, thin tunnels that penetrate the center almost axis parallel from above, below, left, and right. More precisely, the medial axis of the cones that enter these tunnels deepest are parallel to the principal axes. In the *first step* of the construction, we define these wanted tunnels by certain guards. Unfortunately, the same guards that define these tunnels define an even larger number of unwanted diagonal tunnels, which can enter this area as well. Many components

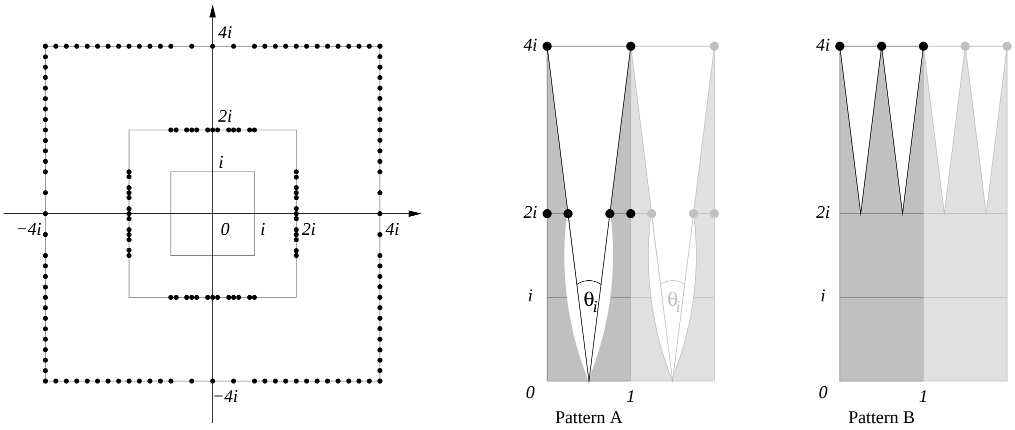


Figure 3.8: Placement of the guards in the first step for $i = 2$ (left). The placement follows the guard patterns A and B (right).

of the desired Θ_i-region in the center are erased by these unwanted tunnels; for example, see the middle picture of Figure 3.7. Therefore, we have to place additional guards in the *second step* of the construction with the intention to prevent unwanted diagonal tunnels from entering this area. This way, we achieve the desired total complexity; compare the example with the right picture of Figure 3.7. We announce that, because of the construction in the second step, the actual convex hull of G_i is huge compared to the box in which we count the connected components. The shape of the Θ_i-region outside of the center is neglected since we cannot expect a significant contribution to the asymptotic bound on the complexity. The detailed construction is presented below.

First step: Determining the ***wanted*** tunnels

By B_i we denote the square of edge length $2i$ that is centered at the origin and is oriented parallel to the principal axes. In this step, we place guards of G_i on the boundary of the boxes B_{4i} and B_{2i}. The area where we expect the connected components is B_i (see the left picture in Figure 3.8). The entire construction is symmetric to the origin as well as to the principal axes. For this reason, we only give the construction for the upper half of the box B_{4i}; the constructions for the lower, left, and right half of this box are done analogously.

We now introduce the guard patterns A and B (see Figure 3.8, right) which define two ways to place guards inside of a cell with width 1 and height $4i$. We use these patterns to mark the location of guards in the upper half of B_{4i}. We define Θ_i as the angle[3] between the rays emanating from $(\frac{1}{2}, 0)$ through the points $(0, 4i)$ and $(1, 4i)$. To get guard pattern A, we place four guards on the boundary of this cone: two with $y = 2i$ and two with $y = 4i$. These four guards define a wanted tunnel

[3] We remark that Θ_i is related to i in the way $\Theta_i = \arctan(\frac{1}{8i}) \leq \frac{1}{8i}$.

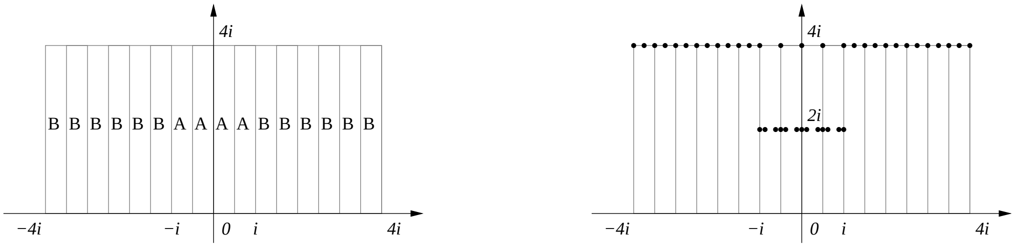


Figure 3.9: Subdivision of the upper half of B_{4i} in $8i$ cells and their guard patterns (left). The resulting placement of guards in the upper half (right).

that is thin in the sense that the boundary of the tunnel stays inside of the lowest quarter of the cell (i.e., for $0 \leq y \leq i$); remember that the focus of the analysis is the interior of B_i. We add guards at $(0, 2i)$ and $(1, 2i)$ to avoid some unwanted diagonal tunnels between adjacent cells with guard pattern A. (These guards do not provide a general protection.) To avoid tunnels inside of the lowest quarter of a cell, we use pattern B: Three guards that are placed equidistant on the top edge of the pattern are sufficient to protect against Θ_i-cones. How do we use these patterns to place the guards? We subdivide the upper half of the box B_{4i} in $8i$ cells of width 1 (see Figure 3.9). Then the medial quarter is stamped with pattern A, and the remaining cells are stamped with pattern B.

This way, we guarantee $2i$ wanted tunnels from above that penetrate B_i and touch the x-axis. Afterwards, we repeat this construction for the lower, left, and right half of B_{4i} with appropriately rotated guard patterns. Then tunnels from above and below touch at the x-axis, and tunnels from the left and the right touch at the y-axis. This follows immediately from the symmetric construction. If we can manage to remove *only these tunnels* from the center of the box B_i, the center is fragmented into $(2i + 1)^2$ connected components.

Two guards are sometimes placed at the same location because of the symmetric construction. It is naturally sufficient to place only one guard there. Therefore, the number of guards in G_i that are placed in the first step sums up to $80i + 4$.

Second step: Excluding the *unwanted* tunnels

Again, we begin with the construction for the upper half of B_{4i}. Figure 3.10 illustrates the guards of the $2i$ cells with patterns A. In particular, we consider pairs of guards through which empty cones can enter B_i from above. We denote the guard pairs on the line $y = 4i$ by $P_1, \ldots, P_{2i}$ and the guard pairs on the line $y = 2i$ by $Q_1, \ldots, Q_{2i}$. An empty cone that enters B_i from above, therefore, has to pass through P_k and Q_ℓ for some $k, \ell \in \{1, \ldots, 2i\}$. If $k = \ell$, the tunnel is wanted. If $k \neq \ell$, however, we need a counter-measure to prevent the cone from entering B_i. We formalize the problem.

Definition 3.1. Let t be the tunnel through P_k and Q_ℓ in the top part of B_{4i}.

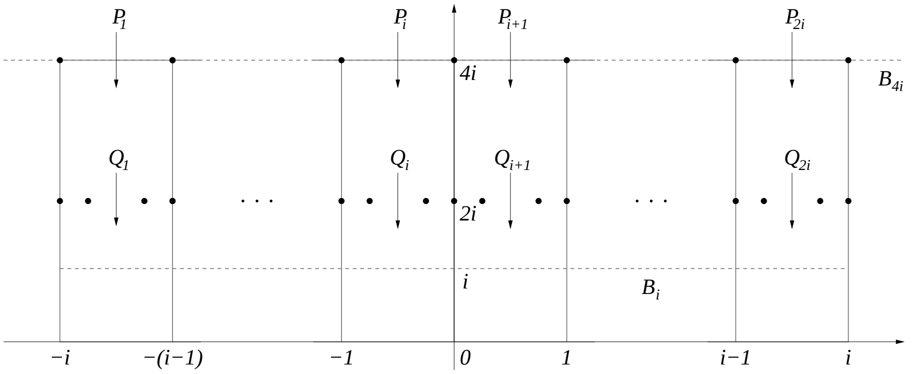


Figure 3.10: Guard pairs through which empty cones can enter B_i from above.

We say that an empty cone enters tunnel t *the deepest* if the y-value of its apex is minimal among all empty cones in t. We define deepest cones analogously for the other three sides of B_{4i}.

Note that the deepest cone in a tunnel is unique and is tangent to at least one guard on each ray. We define the *slope of a cone* as the slope of its medial axis. Because of the regular structure of the cells with guard pattern A, we make the observation that the slope of a deepest cone through P_j and Q_{j+h} is independent of j and is implicitly given by the offset h. We formalize this observation.

Lemma 3.9. *Let $h \in \{0, \ldots, 2i-1\}$. For all $j \in \{1, \ldots, 2i-h\}$, let c_j be the deepest cone through P_j and Q_{j+h}, and let d_j be the deepest cone through P_{j+h} and Q_j. Then the cones c_j for all j have the same slope, and the cones d_j for all j have the same slope.*

For a given offset h, we consider the set of cones c_j. (A similar argumentation is true for the cones d_j.) We derive from Lemma 3.9 that the intersection of all deepest cones c_j is a cone with the same slope (see the shaded region in the left picture of Figure 3.11). Assume that we place a guard at a random position inside of this intersection. Then none of the cones c_j are empty anymore. That means, in this altered situation, the *new* deepest cones must have different slopes and must be tangent to the new guard (see Figure 3.11, middle). Imagine we pull the new guard in the direction of the medial axis of the former deepest cones towards infinity. While we move this guard, the *current* deepest cones are rotated and pushed away from the x-axis. Because of the construction, we can enforce a rotation by an angle that is arbitrarily close to half of the apex angle, i.e., $\frac{\Theta_i}{2} - \varepsilon$ for any $\varepsilon > 0$ (see Figure 3.11, right). Be aware that this construction is possible in general for all $h = \{0, \ldots, 2i-1\}$. Of course, we *may not* place guards in the *union* of the deepest cones for $h = 0$ since these tunnels are wanted. Nevertheless, we used this case in Figure 3.11 since it depicts the worst-case scenario, which is considered in the proof later on.

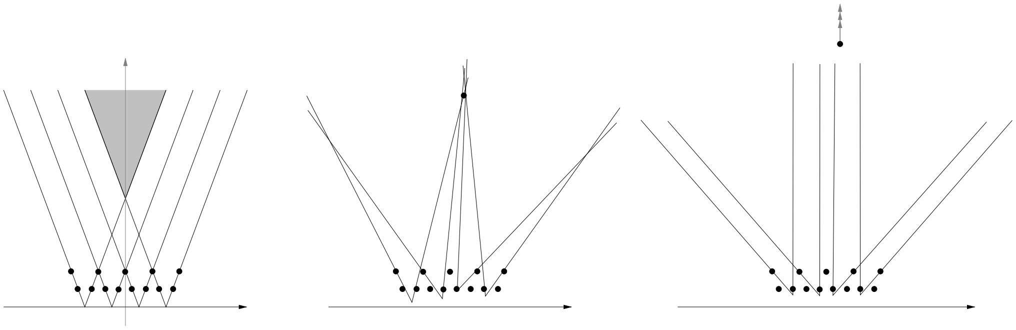


Figure 3.11: A set of deepest cones with the same slope (left). The new deepest cones after the insertion of an additional guard (middle). The limit of the rotation (right).

We are now able to complete the construction. First, we compute the $4i-2$ slopes of the deepest cones c_j and d_j of Lemma 3.9 for all $h = \{1, \ldots, 2i-1\}$. For each s of these slopes, we then place an additional guard at the intersection point of the ray emanating from the origin with slope s and the boundary of a new huge box B_x (see Figure 3.12, left). We have seen that the box B_x must be large enough to guarantee that each intersection point lies

1. outside of the union of the wanted cones and

2. inside of the intersection of the deepest cones of the given slope.

The existence of such a box B_x follows from the discussion above. It remains to show that the additional guards are sufficient to prevent all diagonal tunnels from entering B_i.

Lemma 3.10. *Box B_x can be chosen large enough such that no empty Θ_i-cone can intersect B_i (with the exception of the wanted cones).*

Proof. It is sufficient to prove the claim for the deepest cones whose apex y-value is minimal amongst all deepest cones according to Lemma 3.9. Although these are the cones for $h = 1$, we consider $h = 0$ here because it simplifies the proof. (Be aware that we do not block tunnels for $h = 0$ in practice.)

Remember that we can place the additional guard such that the deepest cones are rotated by an angle arbitrarily close to $\frac{\Theta_i}{2}$. Without loss of generality, we assume that the cone is rotated clockwise.

Therefore, we consider an empty cone c with apex $a = (\frac{1}{4}, i)$ inside a cell with guard pattern A (shaded region in Figure 3.12, right). Assume that its angle is *maximum*, that means, one ray passes through the point $(\frac{1}{4}, 2i)$, and the other ray passes through the point $(1, 4i)$. This cone touches the boundary of B_i, and its left ray is vertical as it is the case for maximal rotated deepest cones. If we can show that the angle of c is smaller than Θ_i, it follows that c cannot enter B_i.

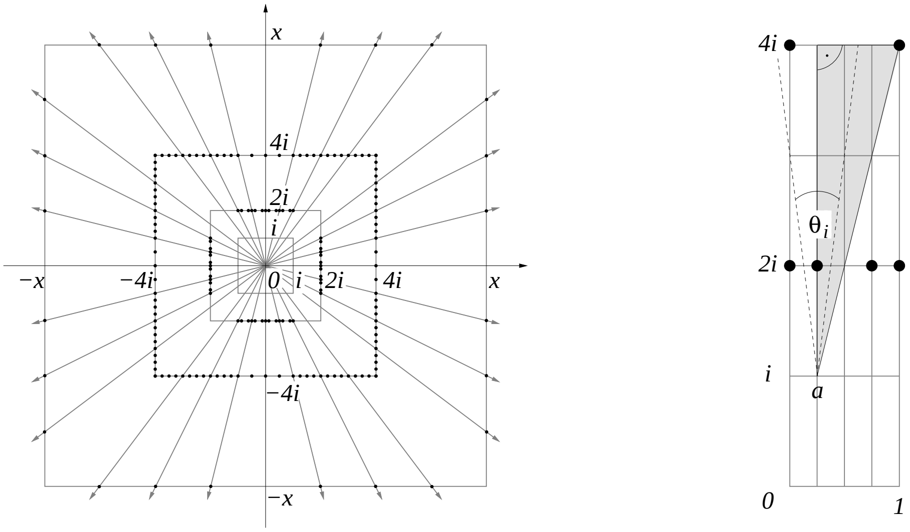


Figure 3.12: Final placement of the guards in both steps for $i = 2$ (left). Auxiliary construction in the proof of Lemma 3.10 (right).

We crop c at the line $y = 4i$ to make it a rectangular triangle. In addition, we draw a (non-empty) vertical Θ_i-cone with apex a. We now divide the shaded triangle along the right boundary of the Θ_i-cone (dashed line) through the point $(\frac{5}{8}, 4i)$; the point of intersection is deduced from guard pattern A (see Figure 3.8). Then the left shaded sub-triangle has angle $\frac{\Theta_i}{2}$ at point a. Since the side of the shaded triangle, which is opposite to a, is divided in the middle, the total angle of the shaded triangle at a must be less than Θ_i. That means, vertical Θ_i-cones cannot enter the box B_i if they are rotated by an angle of $\frac{\Theta_i}{2}$.

As a consequence, it is possible to prevent a series of empty Θ_i-cones *with the same slope* from entering B_i by a single additional guard. □

After repeating this construction for the lower, left, and right halves, we have placed $16i - 8$ additional guards. Together with the guards from the first step, they define the set G_i with $n_i = 96i - 4$ guards in total. We draw the conclusion that the series of examples with guards G_i and angle Θ_i for $i \in \mathbb{N}$ proves the following Theorem.

Theorem 3.11. *The pairs (G_i, Θ_i) for $i \in \mathbb{N}$ define a sequence of Θ-regions whose asymptotic bound on the complexity is $\Omega(n^2)$ where n is the number of guards.*

3.3.4 Computation

We also present a way to compute the boundary of the Θ-region. An overview of the steps is given in Table 3.1. We remark that for any n and any Θ, there are sets G for which the Θ-region is empty or extremely simple, for example, if the guards lie on a straight line. Unfortunately, the presented algorithm has to consider the potential

Step 1: compute the set of relevant arcs (determine $\mathcal{C}'$)
- convex hull ($CH(G)$)
- Θ-maxima ($g_{\min}, g_{\max}$)
- extended partition tree ($\mathcal{T}$)
- querying the extended partition tree (g_ℓ, g_r)

Step 2: compute the arrangement of these arcs (determine $\mathcal{A}(\mathcal{C}')$)
Step 3: determine the Θ-guarded cells (color $\mathcal{A}(\mathcal{C}')$)
Step 4: report the boundary of the Θ-region (traverse $\mathcal{A}(\mathcal{C}')$)

Table 3.1: Instructions for computing the Θ-guarded regions.

arcs in $\mathcal{C}$ and is not output-sensitive in general. We further remark that there is a slight modification of the algorithm: Instead of $\mathcal{C}$, we consider the set $\mathcal{C}'$ of arcs that are longer on one side, i.e., $|\mathcal{C}| = |\mathcal{C}'|$ and $\bigcup \mathcal{C} \subset \bigcup \mathcal{C}'$.

Step 1 (determine $\mathcal{C}'$). We distinguish two types of arcs: Arcs that are purely induced by vertices of the convex hull and arcs that are induced by at least one guard from the interior of the hull. Therefore, we begin with the computation of the convex hull $CH(G)$ in $O(n \log n)$ time [18]. The determination of arcs induced by hull edges is quite simple: For each hull edge (u, v) in clockwise order, we add the circular arc $C^{\Theta}_{u,v}$ to the set $\mathcal{C}'$.

The determination of the second type of arcs is much costlier. For each guard g that lies in the interior of $CH(G)$, we compute all empty cones of maximal angle with apex g together with two guards (witnesses) $g_{\min}$ and $g_{\max}$ per empty cone that lie on its rays (see the light-shaded cone in Figure 3.13). Of course, we are only interested in empty cones whose angle is at least Θ; we disregard cones with smaller angles. We determine the empty cones of maximal angle (also known as Θ-maxima) with the algorithm of Avis et al. [3] in $O(n \log n)$ time and $O(n)$ space for $\frac{\pi}{2} \leq \Theta < \pi$ and in $O(\frac{n}{\Theta} \log n)$ time and $O(n)$ space in general.

Following the proof of Theorem 3.6, we find the arcs in $\mathcal{C}$ via their end points. If we move an empty Θ-cone with apex g and its left ray through $g_{\min}$ along the line through g and $g_{\min}$ until the first time that another guard, say g_r, touches the other ray, the new apex marks an end point p_r of two arcs in the set $\mathcal{C}$ (see Figure 3.13 and remind the left picture of Figure 3.5). Since we do not know the other end points of the arcs, we add the piece of $C^{\Theta}_{g_{\min},g_r}$ to $\mathcal{C}'$ that ends in g_r and p_r, and we add the piece of C^{Θ}_{g,g_r} to $\mathcal{C}'$ that ends in g and p_r. A similar construction for the line through g and $g_{\max}$ adds two more arcs to $\mathcal{C}'$. But how do we determine g_r and g_ℓ?

By fixing the line through g and $g_{\min}$, we could find g_r naively by simply checking all guards in G, and we could find g_ℓ in a similar way. However, we compute the guards g_r and g_ℓ faster with the help of the well-known *Partition Theorem*. We cite the theorem for a planar point set.

Theorem 3.12. *(Partition Theorem [56].) Any set S of n points in the plane can be partitioned into $O(r)$ disjoint classes by a simplicial partition such that every*

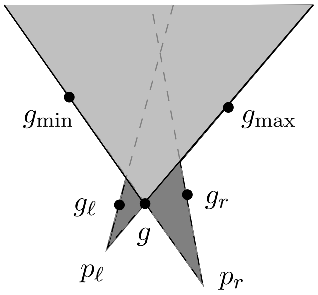


Figure 3.13: From each maximal empty cone at g with angle at least Θ, we derive two arc end points p_ℓ and p_r and four incident arcs.

simplex (i.e., triangle) contains between $\frac{n}{r}$ and $\frac{2n}{r}$ points, and every line crosses at most $O(r^{\frac{1}{2}})$ simplices (crossing number). Moreover, for any $\xi > 0$, such a simplicial partition can be constructed in $O(n^{1+\xi})$ time.

Using this theorem recursively, we construct a tree, called a *partition tree*. That means, the root of the tree is associated with the entire set S, it has $O(r)$ children, each child is a triangle and is associated with a subset of the points from the previous level, and so on. We assume that r is a constant. Then the partition tree is of $O(n)$ size and can be constructed in $O(n^{1+\xi})$ time for any $\xi > 0$. We now claim that it is possible to construct an *extended partition tree* $\mathcal{T}$ with the following property.

Lemma 3.13. *For any $\xi > 0$, there is a data structure of $O(n \log n)$ size and $O(n^{1+\xi})$ construction time such that we can compute the guards g_ℓ (resp. g_r) for a query line through g and $g_{\min}$ (resp. g and $g_{\max}$) in $O(n^{\frac{1}{2}+\xi})$ time.*

Proof. Assume we are given a partition tree. We consider the line through g and $g_{\max}$. Due to Theorem 3.12, the number of first level triangles that intersect this line is bounded by $O(r^{\frac{1}{2}})$. Recurring on those triangles leads to a total of $O(n^{\frac{1}{2}})$ triangles which are intersected by the line. That means, if g_ℓ belongs to one of these triangles, it can be found in time $O(n^{\frac{1}{2}})$.

On the other hand, g_ℓ can also belong to a triangle lying completely to the left of the line $gg_{\max}$. There are $O(r)$ such triangles. For each triangle in the partition tree, we precompute the convex hull of the points associated with the triangle. This increases the total space of the extended partition tree to $O(n \log n)$ since every level in the tree is now of $O(n)$ size. Then, for every triangle that lies to the left of $gg_{\max}$, guard g_ℓ can be found as an extreme point of the precomputed convex hulls. g_ℓ is extreme in the direction perpendicular to the line that forms the Θ-cone with the line through $gg_{\max}$. The additional time to determine the extreme points is $O(\log n)$ in total (see [67], Section 7.9). This way, we avoid recurring on the triangles that lie completely to the left of $gg_{\max}$.

Analogously, we prove the case for the line through g and $g_{\min}$. □

We query the extended partition tree $\mathcal{T}$ to determine g_ℓ and g_r for each Θ-maximum. We use these guards to determine four arcs of $\mathcal{C}'$ as explained above. Remember that the number of Θ-maxima is bounded by $\lfloor \frac{2\pi}{\Theta} \rfloor$ per guard. Therefore, the querying takes total time $O(n^{\frac{3}{2}+\xi}/\Theta)$. We summarize the construction of $\mathcal{C}'$.

Lemma 3.14. *For any $\xi > 0$, the set $\mathcal{C}'$ can be computed in $O(n^{\frac{3}{2}+\xi}/\Theta)$ time and $O(n \log n)$ space.*

Proof. The *total construction time* is the sum of the construction time for the convex hull of G, all Θ-maxima, and the extended partition tree $\mathcal{T}$, plus the query time for all touching points. In the given order, this leads to

$$O\left(n \log n \;+\; \frac{n}{\Theta} \log n \;+\; n^{1+\xi} \;+\; \frac{n^{\frac{3}{2}+\xi}}{\Theta}\right) \;=\; O\left(\frac{n^{\frac{3}{2}+\xi}}{\Theta}\right).$$

The construction that consumes the most space is the extended partition tree $\mathcal{T}$. □

Step 2 (determine $\mathcal{A}(\mathcal{C}')$). We now compute the arrangement $\mathcal{A}(\mathcal{C}')$ from the arc set $\mathcal{C}'$. By m we denote the number of arcs in $\mathcal{C}'$, by ψ we denote the number of cells in $\mathcal{A}(\mathcal{C}')$, and by μ we denote the total complexity of the arrangement $\mathcal{A}(\mathcal{C}')$, which upper bounds the complexity of the Θ-region. Then $\mu \geq m$. Edelsbrunner et al. [21, Theorem 5] showed that μ is at most

$$O\left(\psi^{\frac{1}{2}} \cdot m \cdot 2^{\alpha(m)}\right) \;=\; O\left(\psi^{\frac{1}{2}} \cdot \frac{n}{\Theta} \cdot 2^{\alpha(\frac{n}{\Theta})}\right)$$

where $\alpha(\cdot)$ is the inverse Ackermann function, which is an extremely slow-growing function. Moreover, the arrangement $\mathcal{A}(\mathcal{C}')$ can be constructed in

$$O\left((m + \mu) \log m\right) \;=\; O\left(\mu \log \frac{n}{\Theta}\right)$$

time by the plane-sweep algorithm of Bentley and Ottmann [6].

Step 3 (color $\mathcal{A}(\mathcal{C}')$). We explain the computation of the Θ-guarded region from the arrangement of circular arcs $\mathcal{A}(\mathcal{C}')$. The arcs in $\mathcal{C}'$ contain the bounding circular segments from Formula (3.3). Therefore, the cells in the arrangement $\mathcal{A}(\mathcal{C}')$ have the property that they are either *entirely* Θ-guarded or *entirely* non-Θ-guarded. That means, it suffices to query a random point from the interior of a cell to find out if the entire cell is guarded or not. Hence, we chose a sample point from the interior of each of the ψ different cells in $\mathcal{A}(\mathcal{C}')$ and denote the set of sample points by P. To check which cells belong to the Θ-region, we use the following lemma.

Lemma 3.15 (Avis et al. [3])**.** *Let G be a set of n guards, and let P be a set of ψ query points in $\mathbb{R}^2$. The Θ-unguarded points of P can be reported together with their witnesses $g_{\min}$ and $g_{\max}$ in $O(\frac{n+\psi}{\Theta} \log(n + \psi))$ time.*

Proof. Avis et al. [3, Theorem 2] presented an algorithm to compute all Θ-unguarded guards of G in $O(\frac{n}{\Theta}\log n)$ time. So far, the set of query points and the set of guards are the same. But since their algorithm actually distinguishes between query points and guards, it can be extended immediately: In Steps 2 and 3 of procedure UNORIENTED MAXIMA on page 284f only guards are inserted into the convex hull constructions, while tangents to these convex hulls are only computed through query points. The running time of the algorithm is dominated by sorting the points in $G \cup P$ for $\frac{\pi}{\Theta}$ many directions, which takes $O(\frac{n+\psi}{\Theta}\log(n+\psi))$ time. For more details, see Sections 2 and 6 (Appendix) in [3]. □

This way, we color guarded cells black and non-guarded cells white.

Step 4 (traverse $\mathcal{A}(\mathcal{C}')$). First, we remove all arcs from the arrangement $\mathcal{A}(\mathcal{C}')$ that separate cells of the same color. Then we traverse $\mathcal{A}(\mathcal{C}')$ to detect and report the connected components of the Θ-region. Each time we detect a black cell (i.e., a component of the Θ-region), we report the clockwise boundary of this cell as the sequence of arc end points and arcs. We mark the cell as processed (we color it gray) and continue traversing the arrangement until we have visited each arc once. We can solve this task in $O(\mu)$ time.

We summarize the construction of the four steps in the theorem below.

Theorem 3.16. *For $\Theta < \pi$, the Θ-region can be computed in time*

$$O\left(\frac{n^{\frac{3}{2}+\xi}}{\Theta} + \mu\log\frac{n}{\Theta} + \frac{n+\psi}{\Theta}\log(n+\psi)\right)$$

where $\xi > 0$, ψ is the number of cells in $\mathcal{A}(\mathcal{C}')$, and μ is the complexity of the arrangement $\mathcal{A}(\mathcal{C}')$.

We refine the algorithm for angles $\frac{\pi}{2} \leq \Theta < \pi$ such that it has running time $O(n\log n)$. For these angles, the Θ-region may resemble the convex hull. For example, let G be the vertex set of a regular n-sided polygon where $n \geq 5$; then, the Θ-region is non-empty, has complexity n, and each arc of the Θ-region can be associated with a convex hull edge of $CH(G)$. In such cases, the running time is optimal.

Theorem 3.17. *For $\frac{\pi}{2} \leq \Theta < \pi$, the Θ-region can be computed in time $O(n\log n)$.*

Proof. We slightly change the construction and show that each step takes at most $O(n\log n)$ time.

Step 1: Avis et al. [3] have shown that we can determine the Θ-maxima for angles $\frac{\pi}{2} \leq \Theta < \pi$ in $O(n\log n)$ time. Their procedure UNORIENTED MAXIMA on page 284f reports unguarded points *per convex hull edge* e (as a result of procedure CANDIDATES). Furthermore, these points are sorted along e (the first step of UNORIENTED MAXIMA). Therefore, for each convex hull edge $e = (u, v)$, we can construct the chain of arcs with inscribed angles Θ from u through the ordered set of unguarded candidates in G to v without extra time. We denote this chain by γ_e and the set of all such chains by Γ. Because $\frac{\pi}{2} \leq \Theta$, chain γ_e cannot be self-intersecting. Now, $\mathcal{C}'$ is the set of all arcs in Γ. Each guard is a candidate at most four times. Hence, the total complexity of $\mathcal{C}'$ is linear.

Step 2: Computing $\mathcal{A}(\mathcal{C}')$ is equivalent to computing $\mathcal{A}(\Gamma)$. We also use the algorithm of Bentley and Ottmann [6] in this case. The running time depends on the output complexity, which we consider next. We have said that γ_e is a chain of arcs that connects two vertices u and v of the convex hull. Imagine that we glue the arc chains in Γ together at vertices of the convex hull. Then we obtain a cycle that possibly intersects itself. Let a_i and a_j be two intersecting arcs in the cycle Γ. Then the clockwise arc chain from a_i to a_j cannot intersect the clockwise arc chain from a_j to a_i since it would imply a geometric contradiction (four angles sum up to more than 2π). Therefore, the complexity of $\mathcal{A}(\mathcal{C}')$ is linear and the algorithm of Bentley and Ottmann takes $O(n \log n)$ time.

Step 3 and 4: During the computation of $\mathcal{A}(\mathcal{C}')$, we color faces that lie on the inscribed side of an arc white. That means, points in white-colored faces are unguarded. The unbounded face is clearly colored white, too. After the computation of $\mathcal{A}(\mathcal{C}')$, we color each uncolored face black and report its boundary clockwise. The additional time is bounded by the complexity of the arrangement, which is $O(n)$. □

3.4 Summary: Complexity Bounds

We have seen that the boundary of the Θ-guarded region depends on the location of the guards and the angle Θ. For all angles $0 \leq \Theta < 2\pi$ and $n \in \mathbb{N}$, there are sets of n guards whose Θ-regions are empty; this is particularly true if all guards lie on a straight line.

The worst-case complexity, however, is defined by a set of at most $O(\frac{n}{\Theta})$ circular arcs. The difficulty in the complexity analysis of the Θ-region itself appeared while arguing about the complexity of the union of convex sets U_i, which are bounded by these arcs (cf. Formula 3.3). We summarize our results on the worst-case complexity of the Θ-region in dependence on the angle in Table 3.2. Furthermore, we have given a series of inputs with decreasing angle and increasing number of guards whose asymptotic complexity is $\Omega(n^2)$.

angle Θ	Θ-region	worst-case complexity
2π	$\mathbb{R}^2$	$O(1)$
$\pi < \Theta < 2\pi$	$\supset CH(G)$	$O(\lvert CH(G)\rvert)$
π	$CH(G)$	$\lvert CH(G)\rvert$
$\frac{\pi}{2} \leq \Theta < \pi$	$\subset CH(G)$	$O(n)$
$\delta < \Theta < \frac{\pi}{2}$ for $\delta > 0$	$\subset CH(G)$	$O(n^{1+\varepsilon})$ for $\varepsilon > 0$
$\frac{2\pi}{n} < \Theta$	$\subset CH(G)$	$O(\frac{n^2}{\Theta^2})$ and $\Omega(n^2)$
$0 \leq \Theta \leq \frac{2\pi}{n}$	$\emptyset$	$O(1)$

Table 3.2: *The worst-case complexity of the Θ-region in dependence on the angle Θ.*

Bibliography

[1] M. Abellanas, A. Bajuelos and I. Matos. Some Problems Related to Good Illumination, *International Conference on Computational Science and Its Applications*, pp. 1–14, 2007.

[2] M. Abellanas, M. Claverol and I. P. Matos. The α-Embracing Contour. *International Conference on Computational Science and Its Applications*, pp. 365–372, 2008.

[3] D. Avis, B. Beresford-Smith, L. Devroye, H. Elgindy, E. Guévremont, F. Hurtado and B. Zhu. Unoriented Θ-Maxima in the Plane: Complexity and Algorithms, *SIAM Journal on Computing,* Vol. 28, pp. 278–296, 1999.

[4] D. Avis, D. Bremner and R. Seidel. How Good Are Convex Hull Algorithms? In *Computational Geometry: Theory and Applications*, Vol. 7, pp. 265–301, 1997.

[5] F. Avnaim, J.-D. Boissonnat, O. Devillers, F. P. Preparata and M. Yvinec. Evaluating Signs of Determinants Using Single-Precision Arithmetic. In *Algorithmica*, Vol. 17(2), pp. 111-132, 1997.

[6] J. L. Bentley and T. Ottmann. Algorithms for reporting and counting geometric intersections, *IEEE Transactions on Computers C*, Vol. 28, pp. 643–647, 1979.

[7] E. Berberich, A. Eigenwillig, M. Hemmer, S. Hert, L. Kettner, K. Mehlhorn, J. Reichelt, S. Schmitt, E. Schömer and Nicola Wolpert. EXACUS: Efficient and Exact Algorithms for Curves and Surfaces. In *13th Annual European Symposium on Algorithms,* pp. 155–166, 2005.

[8] M. de Berg, O. Cheong, M. van Kreveld and M. Overmars. *Computational Geometry: Algorithms and Applications.* Springer-Verlag, 3nd edition, 2008.

[9] H. Brönnimann and M. Yvinec. Efficient Exact Evaluation of Signs of Determinants. In *Algorithmica*, Vol. 27(1), pp. 21–56, 2000.

[10] Ch. Burnikel. *Exact computation of Voronoi diagrams and line segment intersections.* PhD Thesis, Max-Planck-Institut für Informatik, Universität des Saarlandes, 1996.

[11] Ch. Burnikel, St. Funke, and M. Seel. Exact Geometric Computation Using Cascading. In *International Journal of Computational Geometry and Applications*, pp. 245–266, 2001; preliminary version *Symposium on Computational Geometry*, pp. 175–183, 1998.

[12] Ch. Burnikel, K. Mehlhorn and St. Schirra. On Degeneracy in Geometric Computations. In *Symposium on Discrete Algorithms*, pp. 16–23, 1994.

[13] M. Caroli. *Evaluation of a Generic Method for Analyzing Controlled-Perturbation Algorithms.* Master's Thesis, Universität des Saarlandes, 2007.

[14] CGAL - *User and Reference Manual: All Parts.* Release 3.9, 2011. `http://www.cgal.org/Manual/latest/doc_pdf/cgal_manual.pdf`

[15] B. Chazelle. On the convex layers of a point set, *IEEE Transactions on Information Theory,* Vol. 31, No. 4, pp. 509–517, 1985.

[16] V. Chvátal. A Combinatorial Theorem in Plane Geometry, *Journal of Combinatorial Theory B,* Vol. 18, pp. 39–41, 1975.

[17] R. Cole, M. Sharir and C. Yap. On k-hulls and related problems, *SIAM Journal on Computing,* Vol. 16(1), pp. 61–67, 1987.

[18] T. H. Cormen and C. E. Leiserson and R. L. Rivest and C. Stein. *Introduction to Algorithms.* The MIT Press and McGraw-Hill, 1990.

[19] O. Deiser. *Einführung in die Mengenlehre.* Springer-Verlag, 2. Auflage, 2004.

[20] P. Deuflhard and A. Hohmann. *Numerische Mathematik I: Eine algorithmisch orientierte Einführung.* de Gruyter Lehrbuch, 3. Auflage, 2002.

[21] H. Edelsbrunner, L. Guibas, J. Pach, R. Pollack, R. Seidel and M. Sharir. Arrangements of curves in the plane—topology, combinatorics, and algorithms, *Theoretical Computer Science,* Vol. 92, Issue 2, pp. 319–336, 1992.

[22] H. Edelsbrunner, D. G. Kirkpatrick and R. Seidel. On the Shape of a Set of Points in the Plane, *IEEE Transactions on Information Theory,* Vol. 29, No. 4, pp. 551–559, 1983.

[23] H. Edelsbrunner and E. P. Mücke. Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms. In *ACM Transactions on Graphics*, Vol. 9(1), pp. 66–104, 1990.

[24] A. Efrat and M. Sharir. On the complexity of the union of fat objects in the plane, *SCG '97: ACM Proceedings of the thirteenth annual symposium on Computational geometry,* pp. 104–112, 1997.

[25] I. Z. Emiris and J. F. Canny. A General Approach to Removing Degeneracies. In *SIAM Journal on Computing*, Vol. 24(3), pp. 650–664, 1995.

[26] I. Z. Emiris, J. F. Canny and R. Seidel. Efficient Perturbations for Handling Geometric Degeneracies. In *Algorithmica*, Vol. 19(1), pp. 219–242, 1997.

[27] Euklid von Alexandria. *Die Elemente. Bücher I-XIII.* Ostwalds Klassiker der Exakten Wissenschaften, Band 235. Verlag Harri Deutsch, 3. Auflage, 1997.

[28] A. Fabri, G.-J. Giezeman, L. Kettner, St. Schirra and S. Schönherr. On the design of CGAL a computational geometry algorithms library *Software Practice and Experience*, Vol. 30(11), pp. 1167–1202.

[29] W. Fischer and I. Lieb. *Funktionentheorie – komplexe Analysis in einer Veränderlichen.* Vieweg Studium, 9. Auflage, 2005.

[30] O. Forster. *Analysis 1: Differential- und Integralrechnung einer Veränderlichen.* Vieweg-Verlag, 8. Auflage, 2006.

[31] O. Forster. *Analysis 3: Maß- und Integrationstheorie, Integralsätze im* $\mathbb{R}^n$ *und Anwendungen.* Vieweg+Teubner, 6. Auflage, 2011.

[32] G. E. Forsythe. Pitfalls in Computation, or why a Math Book isn't Enough. In *The American Mathematical Monthly*, Vol. 77(9), 931–956, 1970. Or in *Technical Report No. CS 147*, Computer Science Department, School of Humanities and Sciences, Stanford University, 1970.

[33] S. Fortune and C. van Wyk. Static analysis yields efficient exact integer arithmetic for computational geometry. In *ACM Transactions on Graphics*, Vol. 15, pp. 223–248, 1996; preliminary version in 7th ACM Conference on Computational Geometry, pp. 163–172, 1993.

[34] St. Funke. *Exact Arithmetic using Cascaded Computation*, Master's Thesis, Universität des Saarlandes, 1997.

[35] St. Funke, Ch. Klein, K. Mehlhorn, and S. Schmitt. Controlled perturbation for Delaunay triangulations. In *Symposium on Discrete Algorithms*, pp. 1047–1056, 2005.

[36] C. G. Gibson. *Elementary geometry of algebraic curves.* Cambridge University Press, 1998.

[37] D. Goldberg. What Every Computer Scientist Should Know About Floating-Point Arithmetic. In *ACM Computing Surveys*, Vol. 23(1), pp. 5–48, 1991.

[38] P. Hachenberger and L. Kettner. Boolean operations on 3D selective Nef complexes: optimized implementation and experiments. In *Symposium on Solid and Physical Modeling*, pp. 163–174, 2005.

[39] D. Halperin and E. Leiserowitz. Controlled perturbation for arrangements of circles. In *International Journal of Computational Geometry and Applications*, Vol. 14(4), pp. 277–310, 2004.

[40] D. Halperin and S. Raab. Controlled perturbation for arrangements of polyhedral surfaces with application to swept volumes. In *Symposium on Computational Geometry*, pp. 163–172, 1999.

[41] D. Halperin and Ch. R. Shelton. A perturbation scheme for spherical arrangements with application to molecular modeling. In *Computational Geometry: Theory and Applications*, Vol. 10, pp. 183–192, 1998.

[42] M. Held. VRONI: An engineering approach to the reliable and efficient computation of Voronoi diagrams of points and line segments. In *Computational Geometry: Theory and Applications*, Vol. 18(2), pp. 95–123, 2001.

[43] G. Hotz. *Einführung in die Informatik.* Leitfäden und Monographien der Informatik, Teubner, 1990.

[44] *IEEE Standard 754-2008 for Floating-Point Arithmetic.* 2008.

[45] T. Imai. A topology oriented algorithm for the Voronoi diagram of polygons. In *Proceeding of the 8th Canadian Conference on Computational Geometry*, Carleton University Press, Ottawa, Canada, pp. 107–112, 1996.

[46] K. Jänich. *Topologie.* Springer-Verlag, 7. Auflage, 2001.

[47] M. Jünger, G. Reinelt, and D. Zepf. Computing correct Delaunay triangulations. In *Computing*, Vol. 47, pp. 43–49, 1991.

[48] M. Karasick, D. Lieber, and L.R. Nackman. Efficient Delaunay triangulation using rational arithmetic. In *ACM Transactions on Graphics*, Vol. 10(1), pp. 71–91, 1991.

[49] K. Kedem, R. Livne, J. Pach and M. Sharir. On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles, *Discrete and Computational Geometry,* Vol. 1, pp. 59–71, 1986.

[50] L. Kettner, M. Mehlhorn, S. Pion, St. Schirra, and C.-K. Yap Classroom Examples of Robustness Problems in Geometric Computations. In *Computational Geometry: Theory and Applications*, Vol. 40, pp. 702–713, 2008.

[51] L. Kettner and St. Näher. Two Computational Geometry Libraries: LEDA and CGAL. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry,* second edition, pp. 1435-1463, 2004.

[52] Ch. Klein. *Controlled Perturbation for Voronoi Diagrams.* Master's Thesis, Universität des Saarlandes, 2004.

[53] E. Lamprecht. *Lineare Algebra I und II.* Birkhäuser, 1993.

[54] C. Li, C. Yap, S. Pion and Z. Du. *Core Library Tutorial.* Courant Institute of Mathematical Sciences, New York University, 2002.
`http://www.cs.nyu.edu/exact/core/doc/tutorial.ps.gz`

[55] D. Matijević and R. Osbild. *Finding the Theta-guarded region.* In *Computational Geometry: Theory and Applications*, Vol. 43(2), pp. 207–218, 2010.

[56] J. Matoušek. Efficient Partition Trees, *Discrete and Computational Geometry,* Vol. 8, pp. 315–334, 1992.

[57] K. Mehlhorn and S. Näher. The Implementation of Geometric Algorithms. In *Proceedings of the 13th International Federation for Information Processing World Computer Congress*, Vol. 1, pp. 223–231, Elsevier, 1994.

[58] K. Mehlhorn and S. Näher. *The LEDA Platform for Combinatorial and Geometric Computing.* Cambridge University Press, 1999.
`http://www.mpi-inf.mpg.de/~mehlhorn/LEDAbook.html`

[59] K. Mehlhorn, R. Osbild and M. Sagraloff. Reliable and Efficient Computational Geometry via Controlled Perturbation. In *International Colloquium on Automata, Languages and Programming*, Vol. 4051 of LNCS, pp. 299–310, 2006.

[60] K. Mehlhorn, R. Osbild and M. Sagraloff. A General Approach to the Analysis of Controlled Perturbation Algorithms. In *Computational Geometry: Theory and Applications*, Vol. 44(9), pp. 507–528, 2011.

[61] D. Michelucci. An epsilon-Arithmetic for Removing Degeneracies. In *Proceedings of the 12th Symposium on Computer Arithmetic*, pp. 230– 1995.

[62] R. Motwani and P. Raghavan. *Randomized algorithms.* Cambridge University Press, 1995.

[63] *MPFI 1.0 - Multiple Precision Floating-Point Interval Library.* SPACES, INRIA Lorraine and Arenaire, INRIA Rhone-Alpes, 2002.
`http://perso.ens-lyon.fr/nathalie.revol/mpfi_toc.html`

[64] The MPFR team. *GNU MPFR - The Multiple Precision Floating-Point Reliable Library.* Edition 3.1.0, 2011.
`http://www.mpfr.org/mpfr-current/mpfr.pdf`

[65] K. Mulmuley. *Computational geometry - an introduction through randomized algorithms.* Prentice Hall, 1994.

[66] J. O'Rourke. *Art gallery theorems and algorithms,* Oxford University Press Inc., 1987.

[67] J. O'Rourke. *Computational Geometry in C*, second edition, Cambridge University Press, 2000.

[68] L. Papula. *Mathematische Formelsammlung für Ingenieure und Naturwissenschaftler.* Vieweg + Teubner, 10. Auflage, 2009.

[69] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, New York, 1985.

[70] M. Sagraloff and C.-K. Yap. A simple but exact and efficient algorithm for complex root isolation. In *The International Symposium on Symbolic and Algebraic Computation*, pp. 353–360, 2011.

[71] M. Seel. *An Accurate Arithmetic Implementation of Line Segment AVDs.* Technical Report, Max-Planck-Institut für Informatik, 1996.

[72] R. Seidel. The Nature and Meaning of Perturbations in Geometric Computing. In *Discrete and Computational Geometry*, Vol. 19(1), pp. 1–17, 1998.

[73] J. R. Shewchuk. Adaptive Precision Floating-Point Arithmetic and Fast Robust Geometric Predicates. In Discrete and Computational Geometry, Vol. 18(3), pp. 305–368, 1997.

[74] K. Sugihara and M. Iri. Construction of the Voronoi diagram for "one million" generators in single-precision arithmetic. In *Proceedings of the IEEE*, Vol. 80(9), pp. 1471–1484, 1992.

[75] K. Sugihara, M. Iri, H. Inagaki and T. Imai. Topology-Oriented Implementation - An Approach to Robust Geometric Algorithms. In *Algorithmica*, Vol. 27(1), pp. 5–20, 2000.

[76] J. Urrutia. Art Gallery and Illumination Problems, in Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, pp. 973–1027, 2000.

[77] C.-K. Yap. Geometric Consistency Theorem for a Symbolic Perturbation Scheme. In *Journal of Computer and System Sciences*, Vol. 40(1), pp. 2–18, 1990.

[78] C.-K. Yap. Symbolic Treatment of Geometric Degeneration. In *Journal of Symbolic Computation*, Vol. 10(3), pp. 349–370, 1990.

[79] C.-K. Yap. Towards exact geometric computation. In *Computational Geometry: Theory and Applications*, Vol. 7(1), pp. 3–23, 1997.