Universiteit Utrecht

Faculty of Science
Department of Mathematics
Department of Information and Computing Sciences

Master's Thesis

Mathematical Sciences and Computing Science

A Framework for Studying the Complexity of General ε-Robust Problems

---

Lammert Westerdijk, BSc

Supervisors:

Dr. T. Miltzow
Universiteit Utrecht

Dr. L. A. Thompson
Universiteit Utrecht

Dr. J. H. Evertse
Universiteit Leiden

September 27, 2024

We study a class of robust versions of $\exists\mathbb{R}$-complete problems, parametrized by the robustness ε as part of the problem statement. By including ε in the problem statement and not the input, for some values of ε the problem can actually become easier than the original non-robust problem. Based on results for ε-Rudi, a robust version of unit disk intersection graph recognition, we derive a general framework for a wide family of ε-robust problems. The framework gives conditions on ε under which the ε-robust problem becomes polynomial-time solvable, or remains in $\exists\mathbb{R}$, or admits a polynomially sized witness, or even becomes undecidable. Specifically, sufficient conditions on ε are given, independent of the specific problem, under which the ε-robust problem is in NP. In particular, ε should be polynomial-time computable and badly approximable by all real algebraic numbers. We begin with a detailed study of such badly approximable numbers, including a comparision with other results on approximation such as in diophantine approximation. We present an example of a computable, badly approximable number and conjecture that a polynomial-time computable, badly approximable number exists. We also present two example framework applications on robust versions of Packing and the Art Gallery Problem.

Foar Heit en Mem

Contents

Introduction

Finding Robust Solutions

Algorithmic problems in computer science are often based on finding an optimal solution under some constraints. For instance, a public transport company might ask for a train schedule that minimizes the total delay, given a number of trains and available drivers, or a metalworking company might want to fit as many pieces onto a single sheet of metal to be cut by the plasma cutter, to minimize costs. Unfortunately, there is often a difference between the real world and the assumed optimal, spherical-cow computer model. A train driver might become sick, a train could break down, or the plasma cutter could have a misalignment. In these cases, the computed optimal solution is no longer possible, and if these potential issues have not been taken into account, serious problems could occur. To prevent this, instead of only looking for an optimal solution, we add the additional constraint that the solution should be *robust*. That is, some extra slack should be built into the solution to account for these unexpected issues.

We will look at so-called *ε-robust* problems, which are problems where the required robustness is parameterized by ε. In particular, we consider robust versions of $\exists\mathbb{R}$-complete problems. The complexity class $\exists\mathbb{R}$ encompasses all problems that are polynomial time equivalent to finding a real root of a polynomial with integer coefficients, and problems that are $\exists\mathbb{R}$-complete often require doubly exponential precision to encode a solution, which makes the solution highly fragile. Hence the interest to look at ε-robust versions of such problems. A more detailed introduction to the complexity class $\exists\mathbb{R}$ along with several examples of $\exists\mathbb{R}$-complete problems is given in Section 1.1.

Does adding ε-robustness to an $\exists\mathbb{R}$-complete problem make the problem easier or harder to solve? This is the main question of this thesis, and the answer turns out to be quite intricate. If the robustness ε is given as part of the input to the problem, then adding robustness cannot make the problem easier, as one could simply input $\varepsilon = 0$ to get the original problem without robustness. However, if instead the robustness ε is encoded as part of the problem statement, so that any ε-robust version of a problem is its own, distinct problem, the problem *can* become easier.

As each ε-robust version of a problem is now a separate problem for all $\varepsilon \geq 0$, perhaps not surprisingly the complexity of an ε-robust problem depends on the specific value of ε. While the original problems are $\exists\mathbb{R}$-complete, we will see that depending on the choice of ε, the resulting ε-robust problem can remain $\exists\mathbb{R}$-complete, or become solvable in polynomial time, become undecidable, and we even conjecture that for a specific ε^* the problem can become a member of NP. We first look at a seemingly unrelated problem.

A special number

The complexity class $\exists\mathbb{R}$ is related to determining whether some given formula has a solution over the reals. Such a formula consists of the constants 0 and 1, binary operators $+$ and $*$, relation symbols $\leq, <$ and $=$, logical operators $\wedge, \vee$ and $\neg$, a set of variables, and parentheses. An example of such a formula is

$$(X * X = 1 + 1) \wedge (X > 0) \wedge (Y > X + 1).$$

A formula is said to *define* a number α through a variable X if in any solution of the formula, we have $X = \alpha$. For instance, the formula above defines $\sqrt{2}$ through X, but defines no number through Y as Y can take different values. Given a real algebraic number α, we can measure its complexity by $L(\alpha)$, the shortest length of any formula that defines α. We are interested in real numbers for which any α that is close has to have a large complexity $L(\alpha)$.

Definition 1.2.17. A real number x is called *badly approximable* if there exist constants $p, C > 0$ such that for any $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$ of description length $L(\alpha)$ we have

$$|x - \alpha| \geq 2^{-CL(\alpha)^p},$$

with at most finitely many exceptions.

In particular, we are interested in whether there are (polynomial-time) computable, badly approximable numbers. A number x is computable if there exists an algorithm that can compute arbitrarily many digits of x in finite time. If this algorithm can compute the first k digits of x in time polynomial in k, the number x is polynomial-time computable. By studying the structure of the set $\mathcal{S}_p$ of all badly approximable numbers with a fixed exponent p, we derive the following theorem.

Theorem 1.5.8. *Let $p > 1$. The set $\mathcal{S}_p^C$ is Lebesgue-measurable and has Lebesgue measure 0.*

The main idea behind the proof of Theorem 1.5.8 is that there are only at most 2^L formulas of length L, and so at most 2^L algebraic numbers α with $L(\alpha) = L$. Each such α excludes an interval of length $2 \cdot 2^{-CL^p}$ in which a badly approximable ε cannot lie. For $p > 1$, the total length of at most 2^L of these intervals decreases exponentially in L, so that their total length becomes arbitrarily small. The result follows from the fact that each non-badly approximable ε has to lie in infinitely many of these intervals.

Theorem 1.5.8 implies that almost all numbers are badly approximable, and so it certainly seems plausible that a (polynomial-time) computable, badly approximable number should exist. Furthermore, any number that we encounter in daily life, such as $\frac{1}{3}$, $\sqrt{2}$, but also π and e, are polynomial-time computable. However, it turns out that in fact almost all numbers are not computable, as there are only countably many algorithms and therefore only countably many computable numbers. Thus, we cannot infer existence of a computable, badly approximable number from Theorem 1.5.8. However, by a different approach, we can derive the following main result.

Theorem 1.6.1. *There exists a computable, badly approximable number.*

We show this by two distinct approaches in Section 1.6. One of these approaches relies on a *badly approximable sequence*.

Definition 1.6.2. A badly approximable sequence $\{s_n\}_{n=1}^{\infty}$ is a sequence of non-negative integers with accompanying polynomial q such that all but finitely many n, for any $\beta \in \overline{\mathbb{Q}} \cap \left[s_n - \frac{1}{2}, s_n + \frac{1}{2}\right)$ we have $q(L(\beta)) \geq n$. A bounded badly approximable sequence satisfies the additional constraint that $s_n < 2^n$ for all n.

Our main result on badly approximable sequences is the following equivalence between badly approximable numbers and badly approximable sequences.

Theorem 1.6.3. *The existence of a (polynomial-time) computable badly approximable number is equivalent to the existence of a (polynomial-time) computable bounded badly approximable sequence.*

In fact, we present polynomial-time reductions that can transform a badly approximable number into a bounded badly approximable sequence, and vice versa. The idea of the proof of Theorem 1.6.3 is as follows. On one hand, suppose $\varepsilon \in [0, 1)$ is badly approximable, then the sequence $\lfloor \varepsilon 2^n \rfloor$ which corresponds to truncating the binary expansion of x is a badly approximable sequence. This follows relatively easily from the fact that ε is badly approximable. On the other hand, suppose we have a bounded badly approximable sequence $\{s_n\}_{n=1}^{\infty}$. Then by choosing $\mathtt{b}_0, \mathtt{b}_1, \ldots$ as the binary expansions of the elements of a suitable subsequence of $\{s_n\}_{n=1}^{\infty}$, the number with binary expansion

$$\varepsilon = 0.00\mathtt{b}_0 00\mathtt{b}_1 00\mathtt{b}_2 00\ldots$$

is badly approximable.

Unfortunately, these methods are not sufficient to show the existence of a polynomial-time computable, badly approximable number. We also compare the description length $L(\alpha)$ to other measures of complexity such as the minimal SLP length $\tau(n)$ and measures such as height $H(\alpha)$, degree $\deg(\alpha)$ and Mahler measure $M(\alpha)$ from the field of diophantine approximation. These measures are defined in detail in Section 1.3 and Section 1.4. Unfortunately, we cannot lift results from these fields to our study on badly approximable numbers, due to the fact that the derived bounds between these measures of complexity are not sufficient. On the other hand, we can derive bounds that would allow a result on badly approximable numbers to be lifted to the study of SLPs or diophantine approximation. The fact that related problems in these fields have been unsolved for some time illustrates the difficulty in finding a polynomial-time computable, badly approximable number.

Robust unit disk intersection graphs

Next, we return to the study of ε-robust problems. In particular, in Chapter 2 we look at the problem in computational geometry known as ε-RUDI, short for *ε-robust unit disk intersection graph recognition.* Given a set of unit disks in the plane, the corresponding *intersection graph* is a graph with a vertex for each unit disk, and an edge between two vertices if and only if the corresponding unit disks intersect. An example of such a unit disk intersection graph is given in Figure 1.

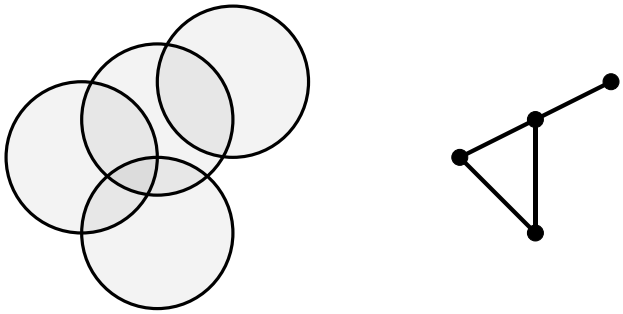


Figure 1: A set of unit disks and their corresponding intersection graph.

The inverse problem of determining whether a given graph is a unit disk intersection graph for some set of disks is known as UDI, short for *unit disk intersection.* It is known that this problem is $\exists\mathbb{R}$-complete [1][2]. In particular, note that deciding whether a graph G is a unit disk intersection graph is equivalent to determining whether some formula has a solution over the reals, as we can add variables for the midpoint coordinates and construct the formula by taking the conjunction of all the distance constraints imposed by G. In ε-RUDI, we still require that vertices that share an edge have distance ≤ 1, but vertices that do not share an edge should now have distance $> 1 + \varepsilon$. A similar corresponding formula can be derived for ε-RUDI, except that we might need the additional constant ε if ε is transcendental.

The first main result on the complexity of ε-RUDI that we showcase is the following theorem.

Theorem 2.1.2. *For each $\varepsilon > \frac{\sqrt{6}+\sqrt{2}}{2} - 1 \approx 0.932$, ε-RUDI is polynomial time solvable.*

The main idea behind this result is the fact that any ε-robust graph with ε this large cannot contain any vertices of degree 3 or higher, so that any graph can consist only of a set of disjoint cycles and lines, which is easy to verify. The next result we show is that for almost all sufficiently small ε, the problem ε-RUDI is undecidable.

Theorem 2.2.1. *ε-RUDI is decidable for only countably many $0 \leq \varepsilon < \frac{2}{\sqrt{3}} - 1 \approx 0.155$.*

While this result may be relatively surprising, the essence is similar to the fact that almost all numbers are not computable. Again, there are only a countable number of algorithms, while there is an uncountable number of ε to choose from in this interval. Thus, if every instance of ε-RUDI in fact needs a different algorithm, the result follows.

To see that every instance of ε-RUDI requires a different algorithm, the idea is to show that for any $0 \leq \varepsilon < \varepsilon' < \frac{2}{\sqrt{3}} - 1$ there is a graph that has an ε-robust representation, but not an ε'-robust one. This result is shown in detail in Section 2.2, but the essence is that a set of vertices that share no edges corresponds to a set of disjoint disks in the plane of diameter $1 + \varepsilon$. Now, if we increase ε to ε', the total area of these disjoint disks increases quadratically. By forcing a cycle of the graph to lie around these disjoint disks, and the fact that the length of this cycle does not increase when increasing ε, as the distance bound ≤ 1 remains the same for edges, the increase in area of the disjoint disks can be forced to be impossible, so that an ε'-robust representation does not exist.

If a graph has an r-robust representation with $r > \varepsilon$, we can slightly perturb the points of the representation, and still retain an ε-robust representation. From this insight, we derive the following *gap theorem.*

Theorem 2.3.1. *Let G be a graph with an r-robust unit disk representation, then there is an ε-robust unit disk representation on a d-grid, as long as*

$$d \leq \sqrt{1/2} \approx 0.707 \text{ and } \varepsilon + 3\sqrt{2}d \leq r.$$

Using this gap theorem, we finally arrive at the relation between badly approximable numbers and ε-robust problems.

Theorem 2.4.1. *Let ε be badly approximable. Then ε-*Rudi *admits a binary witness.*

A *binary witness* is similar to a polynomial witness that is used in proving NP-membership. A binary witness either states that a graph G has no ε-robust disk representation, or contains an arrangement of disks that is polynomially sized in the size of G, showing that G does have an ε-robust disk representation. Contrary to a polynomial witness used in proving NP-membership, we do not require a polynomial-time binary verification algorithm to verify the witness. Instead, we allow for a real verification algorithm that also has access to the constant ε.

As we will show, all graphs G have a rigidity value r, which is the supremum of all ε such that G is ε-robust, and this rigidity value is algebraic. Now, if ε is badly approximable then all rigidity values have to lie relatively far from ε, so that the gap theorem implies that we can find a witness on a relatively coarse grid. This witness is of polynomial size. Furthermore, if ε is polynomial-time computable, the real verification algorithm of the binary witness reduces to a binary polynomial-time verification algorithm. This leads to the following corollary, which shows why we are so interested in finding a polynomial-time computable, badly approximable number.

Corollary 2.4.4. *Let ε be badly approximable and polynomial-time computable. Then ε-*Rudi *is in* NP.

Finally, we show a result on $\exists\mathbb{R}$-completeness. We look at a slightly different variant of ε-Rudi, which instead of requiring a distance $> 1 + \varepsilon$ for a non-edge, only requires a distance of $\geq 1 + \varepsilon$. We denote this problem by ε-Rudi*. By allowing equality, we can reason about certain rigid structures, deriving the following.

Theorem 2.5.2. *The problem $\left(\sqrt{1+\frac{1}{4}}-1\right)$-*Rudi* *is $\exists\mathbb{R}$-complete.*

This result follows by constructing a rigid needle, as in Figure 2, and constructing addition and inversion gadgets to reduce from an $\exists\mathbb{R}$-complete problem known as Etr-Inv[3].

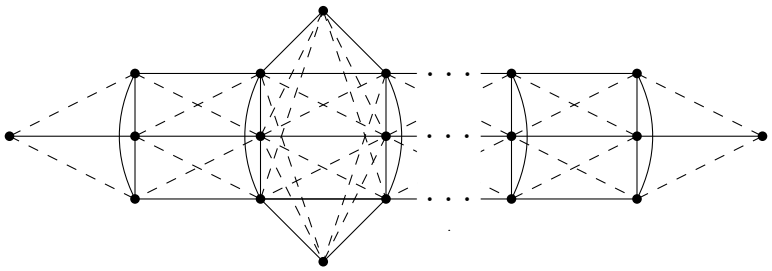


Figure 2: A rigid needle.

A general framework for ε-robust problems

Finally, in Chapter 3, we derive a general framework for ε-robust problems, generalizing the results we saw for ε-Rudi to general ε-robust problems. In particular, in correspondence with the undecidability result on ε-Rudi, we show the following.

Theorem 3.1.2. *If the rigidity values of ε-*Robust Problem *lie dense in the range $[a, b]$ then ε-*Robust Problem *is decidable for only countably many $\varepsilon \in [a, b]$.*

Similarly to the gap result for ε-Rudi, to show results on NP-membership, the ε-robust problem should have the *gap property*.

Definition 3.1.3. We say that ε-Robust Problem has the *gap property*, if the following holds for all $r > \varepsilon$. Pick any $I \in L(r)$ and suppose it has size n. Then there exists a binary witness for I of size polynomial in $n + \log 1/\delta$, where $\delta = r - \varepsilon$. The specific polynomial is allowed to depend on ε.

Similar to ε-Rudi, we use the definition of a badly approximable number to show the following theorem and corollary on NP-membership.

Theorem 3.1.4. *Assume that* ε-Robust Problem *has the gap property and let* ε *be badly approximable. Then* ε-Robust Problem *admits a binary witness.*

Corollary 3.1.6. *Assume that* ε-Robust Problem *has the gap property and let* ε *be badly approximable and polynomial-time computable. Then* ε-Robust Problem *is in* NP*.*

Results on the existence of a simple range, and $\exists\mathbb{R}$-completeness are also derived, but these are mainly problem-specific. To illustrate this, we give two examples of a framework application.

Robust packing

The first ε-robust problem we consider is ε-Robust Packing, which is a robust version of Packing. The problem of Packing asks whether a set of simple polygons with rational vertices fit inside a square without overlap. This problem was shown to be $\exists\mathbb{R}$-complete, even if only convex pieces are allowed, by Abrahamsen, Miltzow and Seiferth[4]. The ε-robust version of this problem, denoted by ε-Robust Packing requires that a configuration exists where the polygons can be arbitrarily shifted by at most a distance ε, still without overlap. As we can scale pieces arbitrarily, which relatively reduces ε, we have the following equivalence.

Theorem 3.2.1. *Let* $\varepsilon \in \mathbb{R}_{\geq 0}$*. For any* $q \in \mathbb{Q}_{>0}$*, the problems* ε-Robust Packing *and* $q\varepsilon$-Robust Packing *are computationally equivalent under polynomial time reductions.*

Using the framework we can show that ε-Robust Packing is often undecidable.

Theorem 3.2.2. *The problem of* ε-Robust Packing *is decidable for only countably many* $\varepsilon \in \mathbb{R}_{\geq 0}$*.*

This result is much easier to show than for ε-Rudi, as a unit square that we want to pack into a square of side length $1 + 2r$ has rigidity r. Also, as we want for NP-membership results, ε-Robust Packing has the gap property.

Theorem 3.2.4. ε-Robust Packing *has the gap property.*

This is shown in Section 3.2.3, and requires a bound on Pythagorean triples to ensure that we can also encode the rotation of the pieces in a polynomial numbers of bits. The related corollaries as indicated by the framework follow immediately.

Corollary 3.2.7. *If* $\varepsilon \geq 0$ *is selected uniformly at random, then* ε-Robust Packing *has a binary witness almost surely.*

Corollary 3.2.8. *If* ε *is badly approximable and polynomial-time computable, then* ε-Robust Packing *is in* NP*.*

Finally, we present a modified version of the $\exists\mathbb{R}$-hardness proof of Packing given by Abrahamsen, Miltzow and Seiferth[4]. This is done in Section 3.2.4.

Theorem 3.2.9. *If* ε *is a real algebraic number then* ε-Robust Packing *is in* $\exists\mathbb{R}$*. If* $\varepsilon \in \mathbb{Q}_{\geq 0}$*, then* ε-Robust Packing *is* $\exists\mathbb{R}$*-complete.*

Robust art gallery problem

The second, and final framework application we consider is the problem of ε-Robust Art Gallery Problem; a robust variant of the Art Gallery Problem where perturbing all of the guards by a distance of at most ε does not change the validity of the solution. The Art Gallery Problem asks whether a certain polygon can be *guarded* by a set of points, so that every point p in the polygon has a guard g such that the segment pg is fully contained in the polygon. The problem was shown to be $\exists\mathbb{R}$-complete by Abrahamsen, Adamaszek and Miltzow[3]. As ε-Robust Art Gallery Problem also allows for arbitrary scaling of the gallery, which decreases the relative influence of ε, we have an identical result to that for ε-Robust Packing.

Theorem 3.3.1. *Let* $\varepsilon \in \mathbb{R}_{\geq 0}$*. For any* $q \in \mathbb{Q}_{>0}$*, the problems* ε-Robust Art Gallery Problem *and* $q\varepsilon$-Robust Art Gallery Problem *are computationally equivalent under polynomial time reductions.*

Results on undecidability and NP-membership also hold for the ε-Robust Art Gallery Problem.

Theorem 3.3.2. *The problem of* ε-Robust Art Gallery Problem *is decidable for only countably many* $\varepsilon \in \mathbb{R}_{\geq 0}$.

Again, the results on rigidity values here are much easier to show than for ε-Rudi. As ε-Robust Art Gallery Problem also has the gap property, the corollaries on binary witnesses and NP-membership follow as well.

Theorem 3.3.5. ε-Robust Art Gallery Problem *has the gap property.*

Corollary 3.3.6. *If* $\varepsilon \geq 0$ *is selected uniformly at random, then* ε-Robust Art Gallery Problem *has a binary witness almost surely.*

Corollary 3.3.7. *If* ε *is badly approximable and polynomial-time computable, then* ε-Robust Art Gallery Problem *is in* NP.

We do not prove any $\exists\mathbb{R}$-completeness results for ε-Robust Art Gallery Problem, as these are again very problem specific and involved, and not that insightful for the workings of the framework.

Throughout this thesis, we use log to denote the base 2 logarithm, and ln to denote the natural logarithm.

Acknowledgements

I would like to sincerely thank my supervisors dr. Miltzow, dr. Thompson and dr. Evertse for the interesting discussions and great suggestions. In addition, I would also like to thank dr. Miltzow once more for proposing this very interesting question on the boundary between mathematics and computer science. While the problem initially seemed quite abstract and daunting, it has been a great challenge to work on. My gratitude also goes out to prof. Linda Kleist, for the useful discussions on the construction of rigid needles for ε-Rudi, and the other coauthors of [5]. This paper was written in parallel with this thesis, and where in this thesis we showcase results of [5] that were shown by a coauthor, this will be clearly indicated by referencing [5].

Finally I would like to thank my parents for all the things they have done for me while I was working on this thesis, despite all the other stressful circumstances they have had to worry about during this period.

Chapter 1

A Special Number

In mathematics, constants such as π and e are often defined in an abstract manner with definitions like "π is the ratio of a circle's circumference to its diameter", and "e is the unique positive number such that the derivative of the function e^x is itself". When working with these constants, mathematicians often leave the symbols in place or bound them very crudely. In computer science however, to perform accurate calculations with such constants, we require accurate approximations of their digits. It is well known that $\pi = 3.14\ldots$ and $e = 2.718\ldots$, but how can we actually compute these digits?

On a computer, all numbers have to be described by a finite number of 1's and 0's. There are many possible ways of describing a number with 1's and 0's, but the simplest for performing calculations is through storing a part of the *binary expansion*. This is comparable to the *decimal expansions* $\pi = 3.14\ldots$ and $e = 2.718\ldots$, but only uses two digits instead of ten. Numbers that have an algorithm that can compute their binary expansion are called *computable numbers*. Numbers that have an algorithm that can do this quickly are called *polynomial-time computable numbers*. Both π and e are polynomial-time computable numbers. In fact, it is very likely that all numbers you will have ever thought about in your life are polynomial-time computable. However, if you were to pick a truly random real number, the probability that this number is polynomial-time computable is 0.

Another way to describe numbers on a computer is through a describing formula. Such formulas are studied in the *existential theory of the reals* and only use the constants 0 and 1, variables, and a series of logical and arithmetic operators. For example, a formula that defines $\frac{1}{2}$ is

$$Z * (1 + 1) = 1.$$

It turns out that not all numbers can be described exactly by such a formula. In fact, both π and e cannot be described in this way. However, we can describe approximations of these numbers. Take e for instance; one of the definitions of e is as the infinite sum $e = \frac{1}{1} + \frac{1}{1} + \frac{1}{1\cdot 2} + \frac{1}{1\cdot 2\cdot 3} + \ldots + \frac{1}{1\cdot 2\cdot \cdots \cdot n} + \ldots$. If we view only the first couple of terms in this infinite sum, we can approximate e by $\frac{1}{1} + \frac{1}{1} + \frac{1}{1\cdot 2} + \frac{1}{1\cdot 2\cdot 3} + \ldots + \frac{1}{1\cdot 2\cdot \cdots \cdot n}$. This number can be described by the following formula:

$$\begin{gathered}(X_1 = 1) \wedge (X_2 = X_1 + 1) \wedge \ldots \wedge (X_n = X_{n-1} + 1) \wedge \\ (Y_0 = 1) \wedge (Y_0 = Y_1 * X_1) \wedge \ldots \wedge (Y_{n-1} = Y_n * X_n) \wedge \\ (Z = Y_0 + Y_1 + \ldots + Y_n)\,.\end{gathered}$$

To encode this formula in 0's and 1's, we need $O(n \log n)$ bits. On the other hand, this approximation is still $\Omega\left(\frac{1}{1\cdot 2\cdot \cdots \cdot (n+1)}\right) = \Omega\left(2^{-n \log n}\right)$ off from the actual value of e. Suppose our computer only has L available 0's and 1's to encode an approximation of e. By this approach, we can only construct approximations of e with an error of at least 2^{-L}. This leads us to one of the important questions of this thesis: can we do better?

It is known that in general one cannot do much better, in the sense that if we again pick a truly random real number the probability that we can do better is 0. However, the probability of the randomly picked number being polynomial-time computable was also 0. The main question of this project therefore asks whether there is a polynomial-time computable number that cannot be approximated better than inverse exponentially.

The existence of such a number, affectionately called "the one to rule them all" by my supervisor dr. Miltzow, would lead to an important result in the complexity of certain problems in computational geometry, and is not apparent by the probabilistic methods discussed above. Instead, we focus on a more explicit, constructive approach, where we attempt to reason from structural results on the set of numbers that can be approximated more effectively than inverse exponentially through ETR formulas.

1.1 The Existential Theory of the Reals and $\exists\mathbb{R}$

We start by defining the *existential theory of the reals*, which is the backbone behind most of the problems that we will study. As the existential theory of the reals is concerned with whether specific formulas have solutions, we first give a formal definition of a formula.

Definition 1.1.1. A *formula*, or more formally a *formula of the first-order theory of the reals*, is a finite sentence using the following symbols:

- The constants 0 and 1;
- The binary operators $+, -$ and $*$;
- The relation symbols $\leq, <, \geq, >, =, \neq$;
- The logical operators $\wedge, \vee, \neg$ and $\Leftrightarrow$;
- The quantifiers $\forall$ and $\exists$;
- Symbols for variables;
- Parentheses.

A formula without any quantifiers is called *quantifier-free*.

The existential theory of the reals is now defined as follows.

Definition 1.1.2. The *existential theory of the reals*, or ETR for short, is defined as the set of true sentences of the form

$$(\exists X_1, \ldots, X_n \in \mathbb{R}^n)\phi(X_1, \ldots, X_n),$$

where ϕ is a quantifier-free formula on the variables $X_1, \ldots, X_n$.

In other words, there is a one-to-one correspondence between elements of ETR and quantifier-free formulas ϕ that have a satisfying assignment over the reals. Hence, determining whether a formula is in ETR is equivalent to determining whether a formula has a satisfying assignment over the reals. The latter problem is oftentimes very useful when studying other algorithmic problems, especially in computational geometry.

Several problems in computational geometry, such as Udi and ε-Rudi discussed in Chapter 2, admit a reduction to deciding whether a certain formula has a real solution. This formula often naturally arises from the constraints of the problem. For Udi and ε-Rudi, the problem asks whether a set of points can be placed in the plane such that certain distance constraints hold. By taking the conjunction of all the distance constraints, we obtain a single formula whose satisfiability is equivalent to the original problem.

In terms of open and closed solution sets, these algorithmic problems reduce to two types of formulas: those that allow equality, and those that only have strict inequalities. The main result shown by Schaefer and Štefankovič[1] is that these two variants of ETR are computationally equivalent by showing that the problem Etr, deciding whether a sentence is in ETR, reduces to Strict Ineq, the variant of Etr only allowing strict inequalities. Thus Schaefer and Štefankovič define the complexity class $\exists\mathbb{R}$ for problems that reduce to ETR, an unambiguous definition as the types of ETR that such problems reduce to are all computationally equivalent.

One of the first problems in computational geometry that was shown to be $\exists\mathbb{R}$-complete is Seg, recognizing whether a given graph can be represented as the intersection graph of a collection of segments in the plane. The $\exists\mathbb{R}$-completeness of Seg was first shown by Kratochvíl and Matoušek[6] through a reduction from Simple Stretchability, another $\exists\mathbb{R}$-complete problem. Other intersection graphs recognition problems that are known to be $\exists\mathbb{R}$-complete include recognizing intersection graphs of convex sets[7], ellipses[7], unit balls[8], (unit) disks[9], 3D segments[10] or unit segments[11]. Many more problems are known to be $\exists\mathbb{R}$-complete; see the compendium by Cardinal, Miltzow and Schaefer[12] for a complete overview.

The $\exists\mathbb{R}$-complete problems of geometric Packing[4] and the Art Gallery Problem[3], together with the unit disk intersection graph recognition problem Udi will be discussed in more detail as we will study ε-robust versions of these problems in the later chapters.

1.2 Defining Formulas and Description Length

As a measure for the complexity of a real algebraic number α, we define its *description length*, which is the length of the shortest formula defining α. For this, we first need to define what we mean by the length of a formula.

Definition 1.2.1. Given a binary encoding of the set of possible symbols in a formula of the first-order theory of the reals, the *length* $L(\phi)$ of a formula ϕ is defined as the total number of bits required to encode ϕ.

Remark 1.2.2. We assume the encoding of symbols is done efficiently, so that the minimal number of bits is needed. In this case, the length of a formula ϕ is basically proportional to the number of symbols it has, with the exception of a formula containing many variables, in which case a formula with n symbols could require up to $O(n \log n)$ bits to encode. However, as the bounds we consider are exponential or even doubly exponential, in most cases this distinction can be ignored, as is done in most of the literature[2]. Thus in most cases, the arguments given in the results that follow can be extended to this alternate definition of length. We stick to the binary definition as it makes counting arguments on the number of formulas of length up to a given bound more concise. We will make extensive use of such arguments.

Having defined the length of a formula, we can extend this definition specifically to real algebraic numbers, as each real algebraic number α satisfies a minimal polynomial f_α which can be translated into a formula. The resulting formula does however not need to be the shortest possible formula for α. In order to properly define the length of a real algebraic number α, we first define what it means for a formula to *define* α.

Definition 1.2.3. A *defining formula* of a real number α is a quantifier-free formula $\Phi(X_1, \ldots, X_n, Y)$ such that $\exists \mathbf{X} \Phi(\mathbf{X}, Y)$ is true if and only if $Y = \alpha$. We say Φ *defines* α *through* Y.

For example, the formula $(Y = X + 1) \wedge (X + X = 1)$ defines $\frac{3}{2}$ through Y, but the formula $(Y * Y = 1)$ is not defining as it is true for both $Y = -1$ and $Y = 1$. We can now define the length of a real algebraic number.

Definition 1.2.4. For any real algebraic number α, we define the *length* $L(\alpha)$ as the minimal length of any defining formula for α.

Remark 1.2.5. Any real number defined by a formula must be algebraic, so that $L(\alpha)$ can only be defined for $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$. On the other hand, for all such α, we can find a defining formula by rewriting the minimal polynomial as a formula. Hence, $L(\alpha)$ is defined precisely for $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$.

A relatively trivial, but useful upper bound on the length of an algebraic number constructed from other algebraic numbers is the following.

Lemma 1.2.6. *Suppose $\Phi(X_1, X_2, \ldots, X_n)$ is a formula with n free variables. There exists an absolute constant C_{add} such that if $\{\alpha_i\}_{i=1}^n$ is a set of real algebraic numbers with minimal description lengths $L(\alpha_i)$, then the minimal description length of $\Phi(\alpha_1, \alpha_2, \ldots, \alpha_n)$ is at most*

$$(1 + \log(n+1))(\sum_{i=1}^{n} L(\alpha_i) + L(\Phi) + C_{add}).$$

The statement of Lemma 1.2.6 is very general. As an example, we can choose Φ as the summation formula such that $\Phi(\alpha_1, \alpha_2, \ldots, \alpha_n) = \alpha_1 + \alpha_2 + \ldots + \alpha_n$. However, as we can also introduce additional variables in Φ, we can also use Lemma 1.2.6 to reason about the description length of $1/\alpha_1 + \alpha_2/\alpha_3 + \ldots + \alpha_n$, for instance. The proof is relatively straightforward.

Proof. Let ϕ_i be minimal ETR formulas defining α_i, respectively. Suppose that their variables are named in such a way that ϕ_i defines α_i through $X_{i,0}$. Then the following ETR formula defines $\Phi(\alpha_1, \alpha_2, \ldots, \alpha_n)$ through X.

$$(X = \Phi(X_{1,0}, X_{2,0}, \ldots, X_{n,0})) \wedge \bigwedge_{i=1}^{n} (\phi_i).$$

To determine the length of this formula, we note that the total number of symbols is at most $\sum_{i=1}^{n} L(\alpha_i) + L(\Phi) + C_{\text{add}}$ for some constant C_{add}. Note that C_{add} is an effective constant that can be computed from the binary encodings. To ensure the encoding works with all variables, we re-encode the variables as

follows. We add an additional string of bits to the encoding of every variable in its original formula, denoting from which of the $n+1$ formulas $\phi_1, \phi_2, \ldots, \phi_n$ or Φ the variable comes. This can be done by adding $\log(n+1)$ additional bits to every variable, and hence every symbol is encoded by at most $\log(n+1)$ additional bits. Thus, the total formula length is bounded by

$$(1+\log(n+1))(\sum_{i=1}^{n} L(\alpha_i) + L(\Phi) + C_{\text{add}}).$$

□

The above proof of Lemma 1.2.6 shows that one should be careful when combining formulas. The length of the new formula is often not simply the sum of the initial formulas, as the new variable encoding has to account for a larger number of variables. Lemma 1.2.6 gives an upper bound in the worst case, and it can sometimes be useful to explicitly try to use as few variables as possible in a formula, so that when combining formulas the new variable encoding becomes less of an issue and one can derive a stronger upper bound.

Again, we note that Lemma 1.2.6 is very general. In practice we will use the following, more restricted, but slightly tighter version.

Lemma 1.2.7. *There exists an absolute constant C_{add} such that if α and β are real algebraic numbers with description length $L(\alpha)$ and $L(\beta)$, then the minimal description lengths of $\alpha+\beta, \alpha-\beta$ and $\alpha*\beta$ are all at most $2(L(\alpha)+L(\beta)+C_{add})$*

Proof. For the algebraic numbers $\alpha+\beta, \alpha-\beta$ and $\alpha*\beta$, the formula $\Phi(X_1, X_2)$ as defined in Lemma 1.2.6 itself contains no variables. Thus we can replace the $\log(n+1)$ in the result by $\log n = \log 2 = 1$. Analogous to the proof of Lemma 1.2.6, the result follows. □

1.2.1 Bounds on description length

While it is generally hard to reason about the minimal description length of a specific algebraic number, there are several bounds that can be of help. One of these bounds, similarly to how we can encode any number n in about $\log n$ bits in binary, implies that any integer has a minimal description length that is at most logarithmically large.

Lemma 1.2.8. *There exists an absolute constant C_{bin} such that for any $n \in \mathbb{N}$ we have $L(n) < C_{bin}(1+\log n)$.*

Proof. The idea is to construct a defining formula for n that is completely analogous to the binary expansion of n. Unfortunately, we cannot do this by simply computing every required power of 2 and summing these, as that would require a logarithmic number of variables as well, resulting in a total length of $O(\log n \log\log n)$. Instead, we construct the formula without any variables, except for the final variable defining n.

We define a sequence of formulas Ψ_n such that the formula $\Phi_n = (X = \Psi_n)$ defines n through the variable X. That is, Ψ_n reduces to the constant n. We proceed by induction. Clearly $\Psi_1 = 1$ works, and for $n > 1$ we distinguish between even and odd n. If n is even, say $n = 2k$, then we can choose

$$\Psi_{2k} = ((1+1) * \Psi_k).$$

On the other hand, if n is odd, say $n = 2k+1$, then we can choose

$$\Psi_{2k+1} = (1 + ((1+1) * \Psi_k)).$$

As both add only a constant number of symbols, and at least halve n, we see that Ψ_n reduces to n, has no variables, and has length $O(1+\log n)$. Thus $\Phi_n = (X = \Psi_n)$ is a defining formula for n of length $O(1+\log n)$. □

Remark 1.2.9. From Lemma 1.2.8, we can also derive upper bounds on the description length of rational numbers. Namely, if we define Ψ_n as in the proof of Lemma 1.2.8, the formula

$$\Phi_{\frac{a}{b}} = (\Psi_b * X = \Psi_a)$$

defines $\frac{a}{b}$ through X. Thus we have $L\left(\frac{a}{b}\right) \leq L(\Phi_{\frac{a}{b}}) \leq C_{\text{bin}}(2 + \log a + \log b)$, perhaps needing to slightly increase C_{bin} to account for the additional parentheses in $\Phi_{\frac{a}{b}}$. Similar reasoning can be employed to show similar upper bounds on the length of any real algebraic number based on its minimal polynomial. This is done in Section 1.4.2.

While it is difficult to find effective lower bounds for the description length $L(\alpha)$, some double logarithmic bounds have been derived. We conclude this section by showing a double logarithmic lower bound on $L(\alpha)$ that can be derived from the following theorem by Schaefer and Štefankovič[1].

Theorem 1.2.10 (Schaefer, Štefankovič [1])**.** *If two semi-algebraic sets in $\mathbb{R}^n$ each of complexity at most $L \geq 5n$ have positive distance (for example, if they are disjoint and compact), then that distance is at least $2^{-2^{L+5}}$.*

The condition that $L \geq 5n$ is somewhat restrictive, but if we define and use all n variables, it is not hard to see that this lower bound is not a problem in most cases. Nonetheless, we can derive the following result from the proof of Theorem 1.2.10.

Theorem 1.2.11. *If two semi-algebraic sets in $\mathbb{R}^n$ each of complexity at most L with $L \geq n$ and $L \geq 3$ have positive distance, then that distance is at least $2^{-(2L^5+2L+3)4^{L+1}}$.*

Proof. Following the proof of Corollary 3.4 in [1], we derive that this distance is at least $2^{-(\tau n^4+2n+3)4^{n+1}}$, with $\tau \leq 2L$. Using the assumption that $n \leq L$, the result follows. □

From this result we can derive the following non-trivial lower bound on the description length of real algebraic numbers close to 0.

Lemma 1.2.12. *Let $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$ be a real algebraic number with $0 < |\alpha| < 1$. Then $L(\alpha) \geq \frac{1}{6} \log \log |\alpha|^{-1}$.*

Proof. Set $L = L(\alpha)$ and let ϕ be a defining ETR formula for α of length L. In particular, choose variables in such a way that if ϕ is true then $X_0 = \alpha$. Clearly ϕ has at most L variables, as each requires a unique symbol. Thus we can set $n = L$ and view the semi-algebraic set $A \subseteq \mathbb{R}^n$ of complexity L defined by ϕ. As $\alpha \neq 0$, ϕ must contain at least one equation and hence $L \geq 3$. Define $B \subseteq \mathbb{R}^n$ as the semi-algebraic set given by $X_0 = 0$, of complexity $3 \leq L$. As any element of A has $X_0 = \alpha \neq 0$ and any element of B has $X_0 = 0$, the sets A and B are disjoint. Furthermore, as B contains all points of $\mathbb{R}^n$ with $X_0 = 0$ and A is non-empty, the two semi-algebraic sets have distance precisely $|\alpha|$. On the other hand, applying Theorem 1.2.11 to A and B implies that A and B have distance at least $2^{-(2L^5+2L+3)4^{L+1}}$. Thus, $|\alpha| \geq 2^{-(2L^5+2L+3)4^{L+1}}$ and so $\log |\alpha|^{-1} \leq (2L^5 + 2L + 3)4^{L+1} \leq 2^{6L}$, where we use the fact that $L \geq 3$. The result follows by taking the logarithm on both sides. □

Remark 1.2.13. The bound of $(2L^5 + 2L + 3)4^{L+1} \leq 2^{6L}$ used above is of course very crude. However, the resulting bound on $L(\alpha)$ is easy to remember and will suffice for our purposes.

Remark 1.2.14. The lower bound of Lemma 1.2.12 is close to optimal in the sense that we can for instance construct a defining formula of length $O(n \log n)$ defining 2^{2^n} through X_n by repeatedly squaring:

$$(X_n = X_{n-1} * X_{n-1}) \wedge (X_{n-1} = X_{n-2} * X_{n-2}) \wedge \cdots \wedge (X_1 = X_0 * X_0) \wedge (X_0 = 1 + 1).$$

As a corollary of Lemma 1.2.12, two real algebraic numbers that lie close together cannot both have small defining formulas.

Corollary 1.2.15. *If $\alpha, \beta \in \overline{\mathbb{Q}} \cap \mathbb{R}$ are two real algebraic numbers with $0 < |\alpha - \beta| < 1$ then at least one of α or β has a description length of at least $\frac{1}{24} \log \log |\alpha - \beta|^{-1} - \frac{1}{2} C_{add}$.*

Proof. Assume to the contrary that $L(\alpha), L(\beta) < \frac{1}{24} \log \log |\alpha - \beta|^{-1} - \frac{1}{2} C_{\text{add}}$. Then from Lemma 1.2.7 we find that $L(\alpha - \beta) < \frac{1}{6} \log \log |\alpha - \beta|^{-1}$. From Lemma 1.2.12, this is a clear contradiction. □

The following result may be somewhat simpler to work with.

Corollary 1.2.16. *There exist absolute constants C_{diff}, c_{diff} such that if $\alpha, \beta \in \overline{\mathbb{Q}} \cap \mathbb{R}$ are two real algebraic numbers with $0 < |\alpha - \beta| < C_{diff}$ then at least one of α or β has a description length of at least $c_{diff} \log \log |\alpha - \beta|^{-1}$.*

Proof. Apply Corollary 1.2.15 and for instance set $C_{\text{diff}} = 2^{-2^{24 \cdot C_{\text{add}}}}$ and $c_{\text{diff}} = \frac{1}{48}$. □

1.2.2 Badly approximable numbers

Both Lemma 1.2.12 and Corollary 1.2.16 imply that a sequence of real algebraic numbers $\alpha_1, \alpha_2, \ldots$ can only approximate a different real number at most doubly exponentially fast in terms of $L(\alpha_i)$. In fact, we will see in a later section that any real algebraic number can also be approximated with exactly this doubly exponential speed, similar to Remark 1.2.14. However, we are interested in the real numbers for which this doubly exponential approximation cannot be realized, and the best approximation is in fact only exponential. We call such real numbers *badly approximable*.

Definition 1.2.17. A real number x is called *badly approximable* if there exist constants $p, C > 0$ such that for any $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$ of description length $L(\alpha)$ we have

$$|x - \alpha| \geq 2^{-CL(\alpha)^p},$$

with at most finitely many exceptions.

Badly approximable numbers will be the main focus of the remainder of this chapter. We are especially interested in what it takes for a real number to be badly approximable, so that we can perhaps find a computable or even polynomial-time computable, badly approximable number. In the later chapters it will become clear why such a number would have great impact on the complexity of a certain family of algorithmic problems, but for now we will simply study the structure of badly approximable numbers without worrying about the specific use-cases.

However, before we delve into the overall structure of badly approximable numbers, we present a specific example to illustrate how the derived bounds on description length can be used to reason about whether a given number is badly approximable.

1.2.3 Partial sums of a specific series

Using Corollary 1.2.16 we can prove lower bounds on the description lengths of particular sums. For example, consider the number

$$x = \sum_{k=0}^{\infty} 2^{-2^k}$$

and denote its partial sums as $\alpha_n = \sum_{k=0}^{n} 2^{-2^k}$. We can for instance show that almost all α_n have complexity cn for some absolute constant $c > 0$.

Definition 1.2.18. We say that a property P holds for *almost all* elements of a sequence $\{a_n\}_{n=0}^{\infty}$ if

$$\lim_{N \to \infty} \frac{|\{a_n | 0 \leq n < N, P \text{ holds for } a_n\}|}{N} = 1.$$

For specific series we can show that almost all partial sums have complexity cn for some absolute constant $c > 0$.

Theorem 1.2.19. *Let $\{b_n\}_{n=0}^{\infty}$ be a sequence of algebraic numbers, and define $a_n = \sum_{k=0}^{n} b_n$ to be the sequence of partial sums of $\{b_n\}_{n=0}^{\infty}$. If there is a positive real number $C > 0$ such that $|b_n| \leq 2^{-2^{Cn}}$ for all but finitely many n, then there exists an effective constant $c > 0$ depending only on C such that $L(a_n) \geq cn$ for almost all a_n.*

Proof. Pick $N \geq 0$ such that all exceptions of $b_n \leq 2^{-2^{Cn}}$ have $n \leq N$. Then for $i > j > N$ we have

$$|a_i - a_j| = \left| \sum_{n=j+1}^{i} b_n \right| \leq \sum_{n=j+1}^{i} |b_n| \leq \sum_{n=j+1}^{i} 2^{-2^{Cn}} < \sum_{n=j+1}^{\infty} 2^{-2^{Cn}}.$$

Rounding each Cn down we can bound this sum as

$$|a_i - a_j| < \sum_{n=j+1}^{\infty} 2^{-2^{Cn}} \leq \left\lceil \frac{1}{C} \right\rceil \sum_{n=\lfloor C(j+1) \rfloor}^{\infty} 2^{-2^n} \leq 2 \left\lceil \frac{1}{C} \right\rceil 2^{-2^{\lfloor C(j+1) \rfloor}}.$$

As C is constant, the bound decreases doubly exponentially in j. Hence, for j sufficiently large, say $j > M$, we can compute an effective constant C' such that

$$|a_i - a_j| < 2^{-2^{C'j}}.$$

From Corollary 1.2.16, it follows that there exists an effective constant c' depending only on C such that for $i > j$ with j large enough, at least one of a_i or a_j has a description length of at least $c'j$. Set $c = \frac{1}{2}c'$. We show that $L(a_n) < cn$ can hold for at most one $n \in [k, 2k)$ for $k > M$, which implies the theorem. Suppose to the contrary that there are $2k > i > j \geq k$ such that $L(a_i) < ci$ and $L(a_j) < cj$. Then $L(a_j) < c'j$ and $L(a_i) < ci < 2ck = c'k \leq c'j$. This is a contradiction with the fact that at least one of a_i or a_j has a description length of at least $c'j$. □

Remark 1.2.20. While Theorem 1.2.19 provides a lower bound on the ETR complexity of almost all partial sums of a quickly decreasing sequence $\{b_n\}_{n=0}^{\infty}$, this quick decrease also implies that $\{a_n\}_{n=0}^{\infty}$ converges doubly exponentially fast. Thus, the theorem is of little use to showing that $\lim_{n\to\infty} a_n$ is a badly approximable number. In fact, we will see later that

$$\sum_{k=0}^{\infty} 2^{-2^k}$$

is in fact not a badly approximable number.

The main takeaway from this section is that it is likely very hard to find a specific example of a badly approximable number x, as one would need exponential lower bounds on almost every real algebraic number in relation to x. The difficulty of the situation becomes apparent when we recall that the best-known lower bounds in the general case are doubly exponential, as in Lemma 1.2.12, and no exponential bounds are known even for a specific x.

Comparing this with the fact that we only need a single sequence $\alpha_1, \alpha_2, \ldots$ that approximates x faster than exponentially to show that x is not badly approximable, we see that it is much easier to show that a specific x is *not* badly approximable, than to show that it is. This is why we will attempt to reason about the entire set of badly approximable numbers, rather than one specific element.

However, before we study the structure of the set of badly approximable numbers, we look at measures of complexity other than description length to see whether there are specific approximation results known for these measures that could perhaps be lifted to our definition of badly approximable numbers.

1.3 Straight Line Programs

The first different measure of complexity we look at, stems from the study of *straight line programs*.

Definition 1.3.1. A *straight line program*, or SLP for short, is a sequence of univariate integer polynomials $\{a_0, a_1, \ldots, a_\ell\}$ with $a_0 = 1$, $a_1 = x$ and for all $2 \leq i \leq \ell$, we have $a_i = a_j \circ a_k$ for $0 \leq j, k < i$ with $\circ$ an operator being either $+, -$ or $*$. The SLP is said to *define* the polynomial a_ℓ and has length ℓ. Note that the polynomial a_ℓ can also be a constant polynomial, so that we can define integers with an SLP.

Adhering with the literature, we denote by $\tau(n)$ the minimal length of an SLP defining n. For instance, we have $\tau(11) = 5$ since the SLP

$$a_0 = 1,\ a_1 = x,\ a_2 = a_0 + a_0,\ a_3 = a_0 + a_2,\ a_4 = a_3 * a_3,\ a_5 = a_2 + a_4$$

defines 11 and has length 5, while no SLP of length less than 5 defines 11.

To give a better idea of how $\tau(n)$ behaves, Table 1.1 includes values of $\tau(n)$ up to 100. Additionally, Table 1.2 includes values of $\tau^*(n)$, which is the minimal length of an SLP defining n, while only using the operators $+$ and $*$. These tables were calculated by brute-forcing over all possible SLPs of fixed length. Similar sequences are also present in the OEIS as A173419 and A230697, where both sequences are always one less than $\tau(n)$ and $\tau^*(n)$, respectively. Related to these sequences we find the minimal lengths of *addition chains*, which are equivalent to SLPs where $+$ is the only allowed operator.

n	$\tau(n)$	n	$\tau(n)$	n	$\tau(n)$	n	$\tau(n)$	n	$\tau(n)$
1	1	21	6	41	7	61	7	81	5
2	2	22	6	42	6	62	6	82	6
3	3	23	6	43	7	63	6	83	6
4	3	24	5	44	7	64	5	84	6
5	4	25	5	45	6	65	6	85	7
6	4	26	6	46	7	66	6	86	7
7	5	27	5	47	7	67	7	87	7
8	4	28	6	48	6	68	6	88	7
9	4	29	6	49	6	69	7	89	7
10	5	30	6	50	6	70	7	90	6
11	5	31	6	51	7	71	7	91	7
12	5	32	5	52	7	72	6	92	7
13	6	33	6	53	7	73	7	93	7
14	5	34	6	54	6	74	7	94	7
15	5	35	6	55	7	75	6	95	7
16	4	36	5	56	6	76	7	96	6
17	5	37	6	57	7	77	7	97	7
18	5	38	6	58	7	78	6	98	7
19	6	39	6	59	7	79	6	99	6
20	5	40	6	60	6	80	6	100	6

Table 1.1: Minimal SLP description length using the operators $\{+, -, *\}$.

n	$\tau^*(n)$	n	$\tau^*(n)$	n	$\tau^*(n)$	n	$\tau^*(n)$	n	$\tau^*(n)$
1	1	21	6	41	7	61	7	81	5
2	2	22	6	42	6	62	7	82	6
3	3	23	7	43	7	63	7	83	6
4	3	24	5	44	7	64	5	84	6
5	4	25	5	45	6	65	6	85	7
6	4	26	6	46	7	66	6	86	7
7	5	27	5	47	7	67	7	87	7
8	4	28	6	48	6	68	6	88	7
9	4	29	6	49	6	69	7	89	7
10	5	30	6	50	6	70	7	90	6
11	5	31	7	51	7	71	8	91	7
12	5	32	5	52	7	72	6	92	7
13	6	33	6	53	7	73	7	93	7
14	6	34	6	54	6	74	7	94	8
15	5	35	6	55	7	75	6	95	8
16	4	36	5	56	7	76	7	96	6
17	5	37	6	57	7	77	7	97	7
18	5	38	6	58	7	78	7	98	7
19	6	39	6	59	8	79	8	99	6
20	5	40	6	60	6	80	6	100	6

Table 1.2: Minimal SLP description length using the operators $\{+, *\}$.

In a similar manner to what we study for description length, there has been research on lower and upper bounds for $\tau(n)$, both in the general case, and for more specific n.

1.3.1 Existing literature

In this section, we give an overview of three existing pieces of literature regarding lower and upper bounds for $\tau(n)$, and straight line programs in general. Recall that we denote the base 2 logarithm by log.

On Asymptotic Estimates for Arithmetic Cost Functions

The paper "On Asymptotic Estimates for Arithmetic Cost Functions" by Moreira[13] provides bounds on the minimal length $\tau(n)$ of an SLP defining a positive integer n. Like with $L(\alpha)$, a double logarithmic lower bound exists for all n, namely

$$\tau(n) \geq \log\log n + 1.$$

This follows from the fact that with the operators $+, -$ and $*$, we can only at most square the maximal element in the SLP at every step. However, also like with $L(\alpha)$, on average the value of $\tau(n)$ will be much greater. Improving upon a result by De Melo and Svaiter[14], Moreira shows that for almost all $n \in \mathbb{N}$ we have

$$\tau(n) \geq \frac{\log n}{\log\log n},$$

which is shown by a counting argument, as is often the case with such proofs. In Section 1.5, we implicitly use a similar counting argument to show that almost all real numbers are badly approximable. On the other hand, Moreira shows an upper bound on $\tau(n)$ that is close to this average lower bound, in the following sense. For any $\varepsilon > 0$ we have

$$\tau(n) \leq (1+\varepsilon)\frac{\log n}{\log\log n}$$

for n sufficiently large. By explicitly constructing n by its binary expansion, we see that $\tau(n) = O(\log n)$. Moreira improves this by choosing a different base than 2, namely $C = B^{\lfloor \log\log n \rfloor}$ with $B = \left\lfloor \frac{\log n}{(\log\log n)^3} \right\rfloor$, and computing the base C digits of n by expanding these in base B.

Remark 1.3.2. Note that Moreira's upper bound on $\tau(n)$ can be used to show the logarithmic upper bound on $L(n)$ as in Lemma 1.2.8. Translating Moreira's SLP defining n into a defining formula in the naive way yields the formula

$$(X_0 = 1) \wedge (X_1 = X_0 \circ X_0) \wedge \cdots \wedge (X_i = X_j \circ X_k) \wedge \ldots \wedge (X_\ell = X_{j'} \circ X_{k'}),$$

where each $\circ$ is one of the operators $+, -$ or $*$ chosen in accordance to the SLP. This formula defines n through X_ℓ and has $O(\ell) = O\left(\frac{\log n}{\log\log n}\right)$ symbols and variables. Thus, the encoding length of the defining formula is

$$O\left(\frac{\log n}{\log\log n}\right)\log\left(O\left(\frac{\log n}{\log\log n}\right)\right) = O(\log n).$$

However, this is actually a weaker result than the defining formula given in the proof of Lemma 1.2.8, as that defining formula uses no variables. Hence, that defining formula can be substituted into another formula without worry, while the defining formula derived from Moreira's result requires a re-encoding of variables, which increases the total length of the formula.

Valiant's Model and the Cost of Computing Integers

In our search for badly approximable numbers, we would like upper bounds for specific sequences of algebraic numbers that are better than the doubly exponential bound. A similar question has been studied for $\tau(n)$. The paper "Valiant's Model and the Cost of Computing Integers" by Koiran[15] relates non-trivial lower bounds on $\tau(x_n)$ for specific sequences of integers x_n to other open problems.

A sequence x_n is said to be *easy to compute* if $\tau(x_n) \leq \text{poly}(\log n)$ for some fixed polynomial relation on $\log n$. The sequence x_n is said to be *ultimately easy to compute* if there is another sequence a_n of positive integers such that the sequence $a_n x_n$ is easy to compute. If x_n is exponential in n, being easy to compute is somewhat equivalent to the known double logarithmic lower bound, which we know is almost never attained. Thus, it is natural to conjecture that sequences such as $x_n = n!$ and $x_n = \lfloor 2^n \ln 2 \rfloor$ are not easy to compute. It has been shown by Shub and Smale that if $n!$ is ultimately easy to compute, then P $\neq$ NP over the field of complex numbers[16].

Koiran shows that if $\lfloor 2^n \ln 2 \rfloor$ is easy to compute, then a superpolynomial lower bound exists for the arithmetic circuit size of the permanent polynomial, and if $n!$ is easy to compute then either a superpolynomial lower bound exists for the arithmetic circuit size of the permanent polynomial or P $\neq$ PSPACE, all of which are open problems. An arithmetic circuit is similar to an SLP, but starts with more variables than just $a_1 = x$. The permanent polynomial of an $n \times n$ matrix X is

$$\mathrm{Per}_n(X) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} x_{i\sigma(i)}.$$

Note that the permanent is similar to the determinant, except that the signs of the permutations are not taken into account. While the specifics of these results on their own are again not of much use to solving our main research question, the fact that a lower bound on the SLP description length of certain sequences of integers that is even slightly better than the double logarithmic bound would implicitly prove these problems that have been open for decades shows that bounding any specific sequence by something better than the double logarithmic bound is in itself already a very difficult problem.

One additional interesting note by Koiran is that if computing remainder and quotient is allowed in the SLP, $n!$ becomes easy to compute. It is interesting to see if this approach could be modified to work for defining formulas, as an SLP is practically a restricted version of a defining formula.

On the Hardness of PosSLP

To give an idea of the current state of research on SLPs, we discuss "On the Hardness of PosSLP" by Bürgisser and Jindal. The authors study the problem of determining whether an integer computed by a straight line program, or SLP, is positive[17]. While the results themselves are not related to any bounds on $\tau(n)$, the paper gives an insight on the current techniques of manipulating SLPs, which could prove useful when studying ETR as well.

Initially, one might suppose that it is easy to test whether an integer defined by an SLP is positive; simply compute the resulting integer. However, by repeatedly squaring, the size of the defined integer can be doubly exponential in the length of the SLP, thus requiring exponential time to compute the integer defined by the SLP. Bürgisser and Jindal show that, under a certain conjecture, if PosSLP is "easy", then so is every problem in NP. The key idea of the proof is transforming a 3SAT formula W into an SLP defining a univariate polynomial F whose roots correspond to satisfying assignments of W. Using oracle calls to PosSLP, we can then sample F at random rational points, and if F ever takes two different signs, by continuity we know that F must have a root, and so W must have a satisfying assignment. This would give a randomized algorithm to decide 3SAT in polynomial time with oracle calls to PosSLP.

One of the main components in the construction of an SLP corresponding to a 3SAT formula are the *Chebyshev polynomials*.

Definition 1.3.3. The *Chebyshev polynomials* T_k are univariate polynomials defined by $T_0(x) = 1$, $T_1(x) = x$ and

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x).$$

The two most important properties of T_k, used in the proof by Bürgisser and Jindal are the fact that the roots of T_k are exactly $\left\{\cos\left(t\frac{\pi}{2k}\right) | t \in \{1, 3, \ldots, 2k-1\}\right\}$ and the fact that $T_{pq} = T_p \circ T_q$, where $\circ$ denotes the composition of functions. Using the recursive definition of T_k, we can construct an SLP of size $O(k)$ defining T_k. However, as we can encode function composition of two functions defined by SLPs by replacing the input $a_1 = x$ of one of the SLPs by the output of the other SLP, by the fact that $T_{pq} = T_p \circ T_q$ we can often encode a Chebyshev polynomial much more efficiently. In particular, if $M = p_1 p_2 \cdots p_n$ is the product of n primes, we can define T_M by an SLP of size $O(p_1 + p_2 + \ldots + p_n)$. This efficiency is similar to the fact that an SLP can encode an integer that is doubly exponential in the size of the SLP, which is why the problem of PosSLP could be hard to begin with. Thus, these Chebyshev polynomials could be useful in a proof that PosSLP should be hard as they can encode a lot of information in terms of their zeros, and can be defined with a relatively small SLP.

To convert a given 3SAT formula W on n boolean variables $x_1, \ldots, x_n$ into an SLP defining a univariate polynomial that encodes the satisfying assignments of W into its set of roots, Bürgisser and Jindal choose n distinct primes $p_1, p_2, \ldots, p_n$, one for each variable.

Letting $M = p_1 p_2 \cdots p_n$, the idea is to define a polynomial whose roots are a subset of the roots of T_M corresponding to the set of satisfying assignments of W. For each satisfying assignment ϕ of W, the number $\alpha(\phi)$ is defined as $\prod_{\phi_i=\top} p_i$. That is, $\alpha(\phi)$ is the product of precisely those p_i with x_i being set to true in ϕ. The set of roots of T_M that correspond to ϕ is defined as

$$S_M(\phi) = \left\{ \cos\left(t\frac{\pi}{2k}\right) | t \in \{1, 3, \ldots, 2k-1\}, \gcd(t, M) = \alpha(\phi) \right\},$$

and the univariate polynomial $\mathrm{POLYSAT}_M(W)$ is defined as the monic polynomial with precisely those roots $S_M(\phi)$ for all satisfying assignments of W. If $C = x_i \wedge x_j \wedge x_k$ is a 3SAT clause, it is not hard to show that $\mathrm{POLYSAT}_M(C) = \frac{T_{M/p_i p_j p_k}}{2^{M/p_i p_j p_k - 1}}$. As $M/p_i p_j p_k$ has a lot of distinct prime factors, we can efficiently define $T_{M/p_i p_j p_k}$, an integer multiple of $\mathrm{POLYSAT}_M(C)$, by an SLP. By the principle of inclusion-exclusion, we can also construct efficient SLPs for the other possible clauses so that for each clause C_i of W we have an SLP of polynomial size defining $F_M(C_i)$, a nontrivial integer multiple of $\mathrm{POLYSAT}_M(C_i)$.

To combine the clauses, note that $\mathrm{POLYSAT}_M(W)$ consists of precisely those roots that are roots of all the $\mathrm{POLYSAT}_M(C_i)$, as any satisfying assignment of W satisfies all its clauses C_i. Thus, the polynomial $P_M(W) = \sum_{i=1}^{m} (F_M(C_i))^2$ has the same roots as $\mathrm{POLYSAT}_M(W)$, but with even multiplicity. While we have an SLP of polynomial size that defines $P_M(W)$, sampling the sign of $P_M(W)$ by calls to PosSLP is not useful to determine whether $P_M(W)$ has a real root, as it is the sum of squares and thus never negative. We would like to modify the SLP defining $P_M(W)$ to an SLP defining a polynomial $F_M(W)$ with the same real roots as $P_M(W)$, but with multiplicity one. For this, Bürgisser and Jindal use the following conjecture.

Conjecture 1.3.4 (Constructive univariate radical conjecture, Dutta, Saxena and Sinhababu[18])**.** *For any polynomial $f \in \mathbb{Z}[X]$, the minimal length of an SLP defining* $\operatorname{rad} f$*, the radical of f can be polynomially bounded in the minimal length of an SLP defining f. Moreover, there is a randomized polynomial time algorithm which, given an SLP of size s computing f, constructs an SLP for* $\operatorname{rad}(f)$ *of polynomial size on s with success probability at least $1 - \frac{1}{\Omega(s^{1+\varepsilon})}$ for some $\varepsilon > 0$.*

Using the constructive univariate radical conjecture, we can take $F_M(W) = \operatorname{rad} P_M(W)$, and we have a probabilistic algorithm to find an SLP of polynomial size defining $F_M(W)$. Now, all real roots of $F_M(W)$ have multiplicity one, and we can apply PosSLP for a randomized algorithm to determine if $F_M(W)$, and thus $\mathrm{POLYSAT}_M(W)$, has any real roots. However, the roots of $F_M(W)$ could be spaced in such a way that the function is only negative on some very small intervals, and we have no guarantee on the probability with which we miss an oppposite sign in every call to PosSLP, even though the 3SAT formula W has a satisfying assignment. This is remedied by Bürgisser and Jindal by first modifying W to a 3SAT formula W' having only a single satisfying assignment if W is satisfiable, and no satisfying assignments otherwise. Then it is shown that both the sets of intervals in $[-1, 1]$ where $F_M(W)$ is negative and where $F_M(W)$ is positive, both have total length at least $\frac{1}{\pi}$, by several analytical results on the spacing of the roots of T_M, which we know are cosines. Hence, we have a guarantee on the probability that PosSLP returns a different sign if W is satisfiable. This results in a randomized polynomial time reduction from 3SAT to PosSLP. Hence, if PosSLP can be solved by a randomized algorithm in polynomial time, so can all problems in NP.

Again, while the main result of Bürgisser and Jindal is of little use to our research on badly approximable numbers, the methods they employ and especially the use of Chebyshev functions could be useful for constructions, even though the results of Bürgisser and Jindal are on SLPs, and we are mainly interested in ETR formulas. Also, Bürgisser and Jindal expose several conjectures on SLP lengths and the description lengths of factors of a polynomial. Besides the constructive univariate radical conjecture, they also expose two different conjectures.

Conjecture 1.3.5 (Factor Conjecture)**.** *For a polynomial $f \in \mathbb{F}[X_1, \ldots, X_n]$ we define $L'(f)$ as the size of the smallest arithmetic circuit computing f from the variables $X_1, \ldots, X_n$ and any constants in the field $\mathbb{F}$. An arithmetic circuit is a generalization of an SLP; a directed acyclic graph with $X_1, \ldots, X_n$ and any constants in the leaves, and where each non-leaf node acts as an arithmetic gate on the values of the outward edges. Over a field $\mathbb{F}$ of characteristic 0, the factor conjecture states that $L'(g) \leq \mathrm{poly}(L'(f) + \deg g)$ for any factor g of $f \in \mathbb{F}[X_1, \ldots, X_n]$.*

An explicit result that can be interpreted as a weaker version of Conjecture 1.3.5 is also given in the paper by Bürgisser and Jindal. Another conjecture that is exhibited by Bürgisser and Jindal is related to the constructive univariate radical conjecture.

Conjecture 1.3.6 (Radical Conjecture)**.** *For a nonzero polynomial $f \in \mathbb{F}[X_1, \ldots, X_n]$ we have*

$$\min\{\deg(\mathrm{rad}(f)), L'(\mathrm{rad}(f))\} \leq \mathrm{poly}(L'(f)).$$

While we will not use either of these conjectures for our results, the fact that these are only conjectures further illustrates the difficulty in reasoning about lower bounds on SLP length. As SLPs can be seen as a restricted variant of ETR formulas, we will have similar difficulty when reasoning about description length.

1.3.2 Relating description length to SLP length

In order to transfer results on description length to SLP length, and vice versa, we require bounds relating SLP length and description length. We restrict ourselves to integers, as these can be defined by both SLPs and ETR formulas. A bound which we have already informally introduced is the following.

Lemma 1.3.7. *For any integer n we have*

$$L(n) = O(\tau(n) \log \tau(n)).$$

Proof. Turning the SLP of length $\tau(n)$ defining n into a defining formula in the naive way yields the formula

$$(X_0 = 1) \wedge (X_1 = X_0 \circ X_0) \wedge \cdots \wedge (X_i = X_j \circ X_k) \wedge \ldots \wedge (X_{\tau(n)} = X_{j'} \circ X_{k'}),$$

where each $\circ$ is one of the operators $+, -$ or $*$ chosen in accordance to the SLP. This formula defines n through $X_{\tau(n)}$ and has $O(\tau(n))$ symbols and variables. By choosing a minimal encoding of variables the result follows. □

Now, if we have any novel lower bound on the description length of a specific integer, Lemma 1.3.7 implies an equally novel lower bound on the SLP length of that integer.

On the other hand, we were unable to derive a similar upper bound on $\tau(n)$ in terms of $L(n)$, aside from the trivial exponential bound that follows from combining the logarithmic upper bound on $\tau(n)$ with the doubly logarithmic lower bound on $L(n)$. However, such a bound would only serve to lift a novel lower bound on $\tau(n)$ to a lower bound on $L(n)$. Besides the fact that we know of no such non-trivial lower bounds on $\tau(n)$ for any specific sequence of integers, this would also only lead to a lower bound on description length for integers, and not any other real algebraic number. Hence, even if we had a very tight bound on $\tau(n)$ in terms of $L(n)$, this result would not really be useful for our purposes.

1.3.3 The SLP length of friable integers

To finish our discussion on SLP length, we show an improvement on the upper bound for $\tau(n)$ given by Moreira[13] when the integer n is of a specific form. For example, if n were a perfect power of 2, say 2^k, then through exponentiation by squaring we see that $\tau(n) = O(\log k) = O(\log \log n)$. The idea of the improvement is to apply a similar reasoning when n is not necessarily a perfect power, but consists of the product of several perfect powers. This is certainly the case when n consists only of relatively small prime factors. As a basic example, we show an improvement upper bound given by Moreira in the case of $\tau(n!)$.

Theorem 1.3.8. *We have $\tau(n!) = O\left(\frac{n}{\ln n}\right)$.*

Proof. Denote the prime factorization of $n!$ by $p_1^{e_1} p_2^{e_2} \cdots p_m^{e_m}$. As the prime factors of $n!$ are precisely all primes $p \leq n$, we have $m = \pi(n)$, where $\pi(n)$ denotes the prime-counting function. Our first claim is that we can compute the primes $p_1, p_2, \ldots, p_m$ in an SLP of length $O(m)$.

From Goldbach's weak conjecture, as proved by Helfgott[19], any odd number greater than 5 can be written as the sum of three primes. In particular, any prime greater than 5 can be written as the sum of three smaller primes. Thus, if we have an SLP computing the first k primes $p_1, p_2, \ldots, p_k$, we can compute p_{k+1} by expanding the SLP by at most 2 additions. As $p_1 = 2, p_2 = 3$ and $p_3 = 5$ can trivially be computed in an SLP of finite length, we can inductively construct an SLP of length $O(m)$ computing the first m primes $p_1, p_2, \ldots, p_m$.

Next, we extend the SLP computing $p_1, p_2, \ldots, p_m$ to compute the powers $p_1^{e_1}, p_2^{e_2}, \ldots, p_m^{e_m}$. Using exponentiation by squaring, each of these powers $p_i^{e_i}$ can be computed in $O(\ln e_i)$ additional operations as p_i has already been computed. Thus, all of the powers $p_1^{e_1}, p_2^{e_2}, \ldots, p_m^{e_m}$ can be computed in $O\left(\sum_{i=1}^{m} \ln e_i\right)$ additional operations. By counting all the numbers $\leq n$ that are divisible by p_i^j we have

$$e_i = \sum_{j=1}^{\infty} \left\lfloor \frac{n}{p_i^j} \right\rfloor \leq \sum_{j=1}^{\infty} \frac{n}{p_i^j} = \frac{n}{p_i - 1} \leq \frac{2n}{p_i}$$

for all $1 \leq i \leq m$, and it follows that

$$\sum_{i=1}^{m} \ln e_i \leq \sum_{i=1}^{m} \ln \frac{2n}{p_i} = \sum_{i=1}^{m} \ln 2 + \ln n - \ln p_i = \pi(n) \ln 2 + \pi(n) \ln n - \vartheta(n),$$

where $\vartheta(n)$ is the Chebyshev function $\vartheta(n) = \sum_{p \leq n} \ln p$. As $\pi(n) = \frac{n}{\ln n} + O\left(\frac{n}{\ln^2 n}\right)$ and $\vartheta(n) = n + O\left(\frac{n}{\ln n}\right)$ by results on the Prime Number Theorem[20][21], we see that

$$\sum_{i=1}^{m} \ln e_i \leq \pi(n) \ln 2 + \pi(n) \ln n - \vartheta(n) = O\left(\frac{n}{\ln n}\right) + \left(n + O\left(\frac{n}{\ln n}\right)\right) - \left(n + O\left(\frac{n}{\ln n}\right)\right) = O\left(\frac{n}{\ln n}\right).$$

Finally, multiplying the powers $p_1^{e_1}, p_2^{e_2}, \ldots, p_m^{e_m}$ to compute $n!$ takes an additional m operations, so that the constructed SLP computing $n!$ has length

$$O(m) + O\left(\sum_{i=1}^{m} \ln e_i\right) + O(m) = O\left(\frac{n}{\ln n}\right) + O(\pi(n)) = O\left(\frac{n}{\ln n}\right).$$

□

Remark 1.3.9. Note that Theorem 1.3.8 is in fact an improvement over the upper bound given by Moreira as Moreira's result yields

$$\tau(n!) \leq (1 + \varepsilon) \frac{\log n!}{\log \log n!} = (1 + \varepsilon) \frac{O(n \log n)}{\Omega(\log n)} = O(n).$$

To generalize the result for $n!$ to other n with relatively small prime factors, we first define what it means to have small prime factors.

Definition 1.3.10. A y-friable number $n \in \mathbb{Z}$ is an integer such that all prime factors of n are at most y.

For example, the integer $60 = 2^2 \cdot 3 \cdot 5$ is 5-friable and 11-friable, but not 3-friable. As another example, all 2-friable numbers are powers of 2.

In a similar way to Theorem 1.3.8, if a number n is y-friable with small y, we can find an SLP computing n that is shorter than the expected $O\left(\frac{\log n}{\log \log n}\right)$.

Theorem 1.3.11. *Let $y \geq 2$ and let $n \in \mathbb{Z}$ be a y-friable number. Denote by $n = p_1^{e_1} p_2^{e_2} \cdots p_m^{e_m}$ the prime factorization of n, then*

$$\tau(n) = O\left(\frac{y \ln \ln n}{\ln y}\right).$$

More specifically, there exists a constant $C > 0$ such that for any n

$$\tau(n) \leq C \left(\frac{y}{\ln y} + m \ln \ln n - m \ln m - \sum_{i=1}^{m} \ln \ln p_i\right).$$

Proof. As n is y-friable, we have $p_1, p_2, \ldots, p_m \leq y$. Again by the Prime Number Theorem and Helfgott's proof of the weak Goldbach conjecture, there is some way of computing all primes up to y, including $p_1, p_2, \ldots, p_m$, in an SLP of size $O\left(\frac{y}{\ln y}\right)$. Note that we have no explicit construction of this SLP, but in

any case it must exist and that is enough to show an upper bound. Applying exponentiation by squaring each power $p_i^{e_i}$ can be computed in $O(\ln e_i)$ operations, so that the final SLP computing n is of length

$$O\left(\frac{y}{\ln y} + \sum_{i=1}^{m} \ln e_i\right).$$

Now as $n \geq p_i^{e_i}$ and $p_i \geq 2$, we have $e_i \leq \ln n$ for any $1 \leq i \leq m$, so that we can roughly bound

$$\tau(n) = O\left(\frac{y}{\ln y} + \sum_{i=1}^{m} \ln e_i\right) = O\left(\frac{y}{\ln y} + m \ln \ln n\right) = O\left(\frac{y \ln \ln n}{\ln y}\right),$$

where the final inequality follows from the fact that $p_1, p_2, \ldots, p_m$ are m distinct primes, all at most y, so that by the Prime Number Theorem $m = O\left(\frac{y}{\ln y}\right)$. For the more subtle bound on $\tau(n)$, we use the method of Lagrange multipliers. As $n = p_1^{e_1} p_2^{e_2} \cdots p_m^{e_m}$ we have $\ln n = e_1 \ln p_1 + e_2 \ln p_2 + \ldots + e_m \ln p_m$. Dropping the constraint that all e_i are positive integers, at the maximum of $\sum_{i=1}^{m} \ln e_i$ the gradient vectors of $f(\mathbf{e}) = \sum_{i=1}^{m} \ln e_i$ and $g(\mathbf{e}) = e_1 \ln p_1 + e_2 \ln p_2 + \ldots + e_m \ln p_m - \ln n$ must be a multiple of each other, i.e. there exists a $\lambda \in \mathbb{R}$ with $\frac{1}{e_i} = \lambda \ln p_i$ for all $1 \leq i \leq m$. As $\ln n = e_1 \ln p_1 + e_2 \ln p_2 + \ldots + e_m \ln p_m$ it follows that $\lambda = \frac{m}{\ln n}$ so that $\sum_{i=1}^{m} \ln e_i$ is maximal when

$$e_i = \frac{\ln n}{m \ln p_i}$$

for all $1 \leq i \leq m$. Hence, for any combination of $e_1, e_2, \ldots, e_m$ we have

$$\sum_{i=1}^{m} \ln e_i \leq \sum_{i=1}^{m} \ln\left(\frac{\ln n}{m \ln p_i}\right) = m \ln \ln n - m \ln m - \sum_{i=1}^{m} \ln \ln p_i,$$

from which the result follows. □

Remark 1.3.12. As $n!$ is an n-friable number, Theorem 1.3.11 implies that

$$\tau(n!) \leq C\left(\frac{n}{\ln n} + m \ln \ln n! - m \ln m - \sum_{i=1}^{m} \ln \ln p_i\right).$$

The prime factors of $n!$ are precisely all primes at most n, so that by the Prime Number Theorem we have $m = \pi(n) = \frac{n}{\ln n} + O\left(\frac{n}{\ln^2 n}\right)$. By Stirling's approximation we have $\ln(n!) = n \ln n - n + O(\ln n)$ so that for some constant $C' > 0$ we have

$$\tau(n!) \leq C'\left(\frac{n}{\ln n} + m\left(\ln n + \ln \ln n\right) - m\left(\ln n - \ln \ln n\right) - \sum_{i=1}^{m} \ln \ln p_i\right) = O\left(\frac{n \ln \ln n}{\ln n}\right),$$

which is slightly worse than the upper bound $O\left(\frac{n}{\ln n}\right)$ obtained in Theorem 1.3.8. This is due to the fact that we assume the worst-case distribution of the e_i in Theorem 1.3.11, whereas we know the precise distribution of the e_i in the case of $n!$, leading to a better bound. In general more structural knowledge on the exponents e_i leads to a better upper bound on $\tau(n)$.

1.4 Diophantine Approximation

In the field of diophantine approximation, an important problem is how closely we can approximate algebraic numbers by rational numbers, similar to our question on badly approximable numbers. There have been many results in this area, with the results for rational approximations crowned by the following famous result of Roth[22].

Theorem 1.4.1 (Roth [22])**.** *Let α be a real, irrational algebraic number. Then for every $\kappa > 2$ the inequality*

$$\left|\alpha - \frac{p}{q}\right| < \frac{1}{q^{\kappa}}$$

only has finitely many solutions $\frac{p}{q} \in \mathbb{Q}$.

Other questions in the field of diophantine approximation include the approximation of transcendental numbers such as e and π by rational[23] or algebraic[24][25] numbers. The relation with whether e or π are badly approximable numbers is apparent. However, where we use the description length $L(\alpha)$ as a measure of the complexity of an algebraic number α, results in diophantine approximation often use measures such as the *height*, *degree*, and *Mahler measure*, which are based on the minimal polynomial of α. Similar to what we did for the minimal SLP length $\tau(n)$, in an attempt to relate badly approximable numbers to results in diophantine approximation, we will also compare $L(\alpha)$ with the height, degree, and Mahler measure of α.

1.4.1 Existing literature

While Roth's theorem is a famous result on the inapproximability of algebraic numbers by rationals, it is of little use to us as we study approximations of real numbers by algebraic numbers. The following works are concerned with approximations of real numbers by rational and algebraic numbers. There is not much to say about these papers and books other than that the results they state could be useful for us if we can find a suitable translation of our problem into the problems of diophantine approximation, and their measures of complexity.

Rational Approximations to e

In the paper "Rational Approximations to e" by Davis[23], an optimal bound is derived on how well e can be approximated by rationals. For $\varepsilon > 0$ there are infinitely many $\frac{p}{q} \in \mathbb{Q}$ with

$$\left| e - \frac{p}{q} \right| \leq (\frac{1}{2} + \varepsilon) \frac{\log \log q}{q^2 \log q},$$

but, of more interest to our research, there are only finitely many $\frac{p}{q} \in \mathbb{Q}$ with

$$\left| e - \frac{p}{q} \right| \leq (\frac{1}{2} - \varepsilon) \frac{\log \log q}{q^2 \log q}.$$

Similar bounds are given for rational approximations of $e^{2/t}$ with t an integer different from 0. The latter bound could be of use to us if we can show a logarithmic lower bound on the description length of q. As we saw from the conditional results on SLP lengths by Koiran[15], such a lower bound is likely very difficult to show. However, the similar bounds for $e^{2/t}$ might allow for a counting argument.

Approximation by Algebraic Numbers

Perhaps more interesting than approximations with rationals is the approximation of real numbers by algebraic numbers and so-called *transcendence measures*. In particular, the book "Approximation by Algebraic Numbers" by Bugeaud[24] provides an excellent overview of the results in this area. An important idea are so-called *transcendence measures*. A transcendence measure for a transcendental number ξ is a function $\omega(n, H)$ so that for any n, H sufficiently large we have $|P(\xi)| \geq e^{-\omega(n, \ln H)}$ for all non-zero polynomials $P = \sum_{i=0}^{m} c_i X^i$ with $\deg P = m \leq n$ and $H(P) = \max_{i=0}^{m} |c_i| \leq H$[24][25]. By choosing P to be the minimal polynomial f_α of an algebraic number α, we can derive a lower bound on $|\xi - \alpha|$ assuming we have a bound on the derivative of f_α around α. A transcendence measure for π is

$$\omega(n, H) = 2^{40} n (\ln H + n \ln n)(1 + \ln n),$$

as exhibited by Bugeaud. Transcendence measures for other transcendental numbers such as e are also known[25]. These transcendence measures are quite good and could perhaps be used to show that e or π are specific badly approximable numbers. However, in order for us to make use of these results, we should show efficient bounds on $H(\alpha) = H(f_\alpha)$ and $\deg \alpha = \deg f_\alpha$ in terms of the description length $L(\alpha)$. Unfortunately, the following section shows that this hope is relatively futile.

1.4.2 Relating description length to height, degree and Mahler measure

In diophantine approximation, often used complexity measures for algebraic numbers include *height*, *degree* and the *Mahler measure*[26]. These are all based on the *minimal polynomial*.

Definition 1.4.2. The *minimal polynomial* f_α of an algebraic number α is the monic polynomial with rational coefficients of minimal degree, having α as a root.

For example, the minimal polynomial of $\frac{1}{3}$ is $X - \frac{1}{3}$ and the minimal polynomial of $\sqrt{2}$ is $X^2 - 2$. Note that, in contrast to the minimal defining formula for α, the minimal polynomial of α is allowed to have real roots other than α. From the minimal polynomial f_α, we can derive the complexity measures mentioned above.

Definition 1.4.3. For an algebraic number α let $f_\alpha(X) = \sum_{i=0}^{n} c_i X^i = \prod_{j=1}^{n}(X - \alpha_j) \in \mathbb{Q}[X]$ be its minimal polynomial, of degree n. Then the *degree* of α is $\deg \alpha = n = \deg f_\alpha$. Furthermore, let $F_\alpha(X) = \sum_{i=0}^{n} z_i X^i \in \mathbb{Z}[X]$ be the unique integer multiple of f_α such that $z_n > 0$ and $\gcd(z_0, z_1, \ldots, z_n) = 1$. In other words, $F_\alpha = C f_\alpha$ with C the least common multiple of the denominators of all c_i. Then the *height* of α is defined as $H(\alpha) = \max_{0 \le i \le n} |z_i|$ and the *Mahler measure* of α is $M(\alpha) = z_n \prod_{|\alpha_j| \ge 1} |\alpha_j|$.

As mentioned before, approximation bounds in diophantine approximation are often stated in terms of height, Mahler measure, degree, or any combination of the three. Most bounds, such as the bound by Roth[22], are polynomial in height and Mahler measure, or exponential in the degree. As the bound we want to show is exponential with polynomial exponent in $L(\alpha)$, we might hope that there are bounds on $H(\alpha)$ and $M(\alpha)$ that are exponential with polynomial exponent in $L(\alpha)$, or a bound on $\deg \alpha$ that is polynomial in $L(\alpha)$. We will now crush those hopes. For $n \in \mathbb{N}$, consider the polynomial $f \in \mathbb{Z}[X]$ given by

$$f(x) = x^{2^n} - \left(2^{2^n} - 2\right).$$

As $n \ge 1$, we have $2^{2^n} - 2 \equiv 2 \pmod 4$, so that f is irreducible over $\mathbb{Q}$ by the Eisenstein criterion for $p = 2$. Clearly, f is also monic. Let $\alpha \in \mathbb{R}_{>0}$ be the only positive real root of f. Then f is the minimum polynomial of α and hence $H(\alpha) = 2^{2^n} - 2$, $\deg \alpha = 2^n$ and $M(\alpha) = 2^{2^n} - 2$, denoting the height, degree and Mahler measure of α, respectively. The result on the Mahler measure follows from the fact that all roots α_j of f are 2^n-th roots of $2^{2^n} - 2$, so that $|\alpha_j| > 1$. Then

$$M(\alpha) = \prod_{|\alpha_j| \ge 1} |\alpha_j| = \prod_{j=1}^{n} |\alpha_j| = |c_0| = 2^{2^n} - 2.$$

On the other hand, a defining formula for α is given by

$$(Z \ge 0) \wedge (Z * Z = X_{n-1}) \wedge (X_{n-1} * X_{n-1} = X_{n-2}) \wedge \cdots \wedge (X_1 * X_1 = C_n - 2) \wedge$$
$$(C_n = C_{n-1} * C_{n-1}) \wedge (C_{n-1} = C_{n-2} * C_{n-2}) \wedge \cdots \wedge (C_1 = C_0 * C_0) \wedge (C_0 = 1 + 1),$$

defining α through Z. The length of this formula is $O(n \log n)$, so that $L(\alpha) = O(n \log n)$. But, $H(\alpha) = M(\alpha) = 2^{2^n} - 2 = \Omega(2^{2^n})$ and $\deg(\alpha) = 2^n$. Hence, it is impossible to find any universal bounds on $H(\alpha)$ and $M(\alpha)$ that are exponential with polynomial exponent in $L(\alpha)$, or a universal bound on $\deg \alpha$ that is polynomial in $L(\alpha)$.

However, it is likely that we cannot do much worse than the example above. In particular, we make the strong assumption that the defining formula for α can be reduced to a univariate polynomial $g(x) = \sum_{i=0}^{n} a_i X^i \in \mathbb{Z}[X]$ with $a_n \neq 0$ and the zero set given by the zero set of the defining formula, i.e. $\{\alpha\}$, without negatively affecting the length too much. That is, $g(\alpha) = 0$ and $L(\alpha) = \Omega(L(g)) = \Omega(\log(n) + \max_{i=0}^{n} L(a_i))$. This assumption is strong in the sense that quantifier elimination can result in a quantifier-free formula that is doubly exponential in the size of the original formula[2]. Note that we need $\Omega(\log(n))$ symbols to describe the degree of f, since the symbols $\{+, -\}$ do not change the total degree and each $\{*\}$ at most doubles the degree. Hence we need at least $\log(n)$ symbols $*$. Even allowing other logical operators will not change this as they can be expressed as arithmetic relations which increase the degree of the polynomial by at most some constant C. The bound $\Omega(L(a_i))$ comes from the fact that we can also use g to describe a_i with length $O(L(g))$. Denote by $f \in \mathbb{Z}[X]$ the minimum primitive polynomial of α. Note that we must have $f \mid g$ over $\mathbb{Q}$.

For the degree of α, note that as $f \mid g$ we have $\deg(\alpha) = \deg f \le \deg g = n$. As $L(\alpha) = \Omega(\log(n) + \sum_{i=0}^{n} \log\log(a_i))$ we have $L(\alpha) = \Omega(\log(n)) = \Omega(\log \deg(\alpha))$ and so $\deg(\alpha) = 2^{O(L(\alpha))}$.

From the Landau-Mignotte[27] bound we find $H(\alpha) = \|f\|_\infty \leq 2^n\sqrt{n+1}\|g\|_\infty$ so that

$$\log\log H(\alpha) \leq \log(n + \log\sqrt{n+1} + \log\|g\|_\infty) \leq \log(2n + \log\|g\|_\infty)$$

for n sufficiently large. By splitting some maxima we can derive

$$\log\log H(\alpha) = O(\log n + \log\log\|g\|_\infty).$$

Now let $a_i = \|g\|_\infty$ be the coefficient of g with maximum absolute value. We know that $\log\log\|g\|_\infty = \log\log a_i = O(L(a_i))$, and so

$$\log\log H(\alpha) = O(\log n + L(a_i)) = O(L(\alpha)).$$

It follows that $H(\alpha) = 2^{2^{O(L(\alpha))}}$. For the Mahler measure we use the well-known upper bound[26] in terms of $H(\alpha)$ to bound $M(\alpha) \leq \sqrt{\deg(\alpha)+1}H(\alpha) \leq \sqrt{n+1}H(\alpha) \leq 2^n(n+1)\|g\|_\infty$ and we can use the same asymptotics to prove $M(\alpha) = 2^{2^{O(L(\alpha))}}$.

On the other hand, similar to our comparison between SLP length and description length, we can provide a relatively good bound in the other direction. By transforming the minimal polynomial f_α to an ETR formula, we can derive a defining formula for α. Intuitively, it should take roughly $O(\deg\alpha\log H(\alpha))$ bits to describe f_α as an ETR formula, and so we can derive an upper bound for $L(\alpha)$ in terms of $\deg\alpha$ and $H(\alpha)$. As a defining formula has to have a unique solution while f_α can have multiple roots, we do have to be slightly more careful in how we define a defining formula for α from f_α; we have to specifically select α from the set of roots of f_α. For this we use the following theorem due to Mahler[28].

Theorem 1.4.4. *Let α be an algebraic number of degree $d = \deg\alpha$ and height $H = H(\alpha)$. Then*

$$\text{sep}(\alpha) \geq \sqrt{3}(d+1)^{-(2d+1)/2}H^{-d+1},$$

where $\text{sep}(\alpha)$ *is the minimal distance between any two roots of f_α.*

With this bound on root separation, we can show an upper bound on $L(\alpha)$ in terms of the height and degree of α.

Theorem 1.4.5. *Let $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$ be a real algebraic number of degree $d = \deg\alpha$ and height $H = H(\alpha)$. Then*

$$L(\alpha) = O(d\log d + d\log H).$$

Proof. We construct a defining formula for α, being careful to introduce a minimal number of variables. Denote by $F_\alpha(X) = \sum_{i=0}^{d} z_i X^i \in \mathbb{Z}[X]$ the unique integer multiple of f_α such that $z_d > 0$ and $\gcd(z_0, z_1, \ldots, z_d) = 1$. By definition we have $|z_i| \leq H$ for all $0 \leq i \leq d$. From the proof of Lemma 1.2.8, we have expressions Ψ_i of length $O(\log z_i) = O(\log H)$ without any variables, that reduce to z_i. By rewriting $F_\alpha(X)$ to an ETR formula,

$$\psi_f = (\Psi_0 + X * (\Psi_1 + X * (\Psi_2 + X * (\ldots))) = 0)$$

is a formula with $O(d\log H)$ symbols and a single variable that has a solution if and only if $F_\alpha(X) = 0$. What remains is to isolate $X = \alpha$ as the only solution. To this end we add the bounds $L \leq X < U$ for L and U defined as suitable rationals ℓ and u. Defining $q = (d+1)^{d+1}H^d$ and $p = \lfloor\alpha q\rfloor$, we take $\ell = \frac{p}{q}$ and $u = \frac{p+1}{q}$. Clearly $\ell \leq \alpha < u$, and as $\text{sep}(\alpha) \geq \frac{1}{q}$ by Theorem 1.4.4, we cannot have $\ell \leq \alpha^{(i)} < u$ for any conjugate $\alpha^{(i)} \neq \alpha$ of α.

To finish the proof we show that ℓ and u can be defined through relatively short formulas. Note that as $q = (d+1)^{d+1}H^d$ we have an expression Ψ_q reducing to q with $O(d\log d + d\log H)$ symbols and no variables. Writing $p = kq + r$ with $0 \leq r < q$ we have $|k| \leq |\alpha| \leq 1 + H$, where the last bound is a consequence of Cauchy's bound on polynomial roots[29]. Thus we have an expression Ψ_k reducing to k $O(\log(1+H))$ symbols and no variables, and an expression Ψ_r reducing to r with $O(d\log d + d\log H)$ symbols and variables. Now the formula

$$\psi = (\psi_f) \wedge (\Psi_r \leq \Psi_q * (X - \Psi_k) < \Psi_r + 1)$$

defines α through X and has $O(d\log d + d\log H)$ symbols and a single variable. □

Remark 1.4.6. Note that Theorem 1.4.5 can be seen as a generalization of the results on description length for integers and rationals given in Lemma 1.2.8 and Remark 1.2.9. In fact, for every fixed degree d any algebraic number has $L(\alpha) = O_d(\log H)$.

The term $d \log d$ in the bound of Theorem 1.4.5 stems from the separation bound Theorem 1.4.4. If an improved separtion bound in terms of d holds, we can show that $L(\alpha) = O(d \log H)$. In particular, we conjecture the following.

Conjecture 1.4.7. *There exists a constant $C > 0$ such that for any algebraic number α of degree $d = \deg \alpha$ and height $H = H(\alpha)$ we have*

$$\operatorname{sep}(\alpha) > (2H)^{-Cd}.$$

As mentioned, this conjecture would imply that $L(\alpha) = O(d \log H)$. Unfortunately, the available literature on root separation seems mainly focused on improving the exponent of H in the separation bound, which only worsens the dependency on d. On the other hand, a counterexample is also not obvious, as one would need a sequence of polynomials of fixed height, or at least with $H \ll d$, whose root separation decreases faster than 2^{-d}. The family of polynomials that seems closest to a counterexample in the literature is due to Mignotte[29]:

$$p_d(X) = X^d - 2(2X - 1)^2$$

has height 8 and root separation that decreases like $2^{-d/2}$.
Similar to the results for SLP length, we see that we can derive a relatively good upper bound on $L(\alpha)$ in terms of height and degree, but we cannot hope for a good general lower bound on $L(\alpha)$. As with our study on SLP length, it follows that we cannot use the results on transcendence measures and other results from diophantine approximation to show general results on badly approximable numbers. On the other hand, we *can* extend results on badly approximable numbers to the setting of diophantine approximation. This once again illustrates the expected difficulty of finding general results on badly approximable numbers.

1.5 The Set $\mathcal{S}_p$

We return to the topic of badly approximable numbers. While we are mainly interested in finding a single badly approximable number that is polynomial-time computable, it is interesting and useful to look at badly approximable numbers in a set-theoretic manner to discover more about their structure. The hope is that such structural insights might give rise to a polynomial-time computable example of a badly approximable number.

While we were unfortunately not able to find such a special number, in this section we do find examples of badly approximable numbers that are computable, just not in polynomial time. Furthermore, we derive an equivalence between badly approximable numbers and badly approximable sequences, which further indicates that finding a polynomial-time computable, badly approximable number is far from trivial.

We begin by defining the set $\mathcal{S}_p$ of badly approximable numbers.

Definition 1.5.1. For a real number $p > 0$, let $\mathcal{S}_p \subseteq \mathbb{R}$ denote the set of reals x for which there exists a constant $C(x) > 0$ such that for any $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$ of description length $L = L(\alpha)$ we have

$$|x - \alpha| \geq 2^{-C(x)L^p},$$

with at most finitely many exceptions.

The remainder of this section is dedicated to study the structure of $\mathcal{S}_p$.

1.5.1 Basic properties

The first important structural result on $\mathcal{S}_p$ is the fact that both $\mathcal{S}_p$ and its complement $\mathcal{S}_p^C$ are closed under addition with real algebraic numbers.

Lemma 1.5.2. *The set $\mathcal{S}_p$ is closed under addition with real algebraic numbers $\beta \in \overline{\mathbb{Q}} \cap \mathbb{R}$.*

Proof. Let $x \in \mathcal{S}_p$ and $\beta \in \overline{\mathbb{Q}} \cap \mathbb{R}$ and define $y = x + \beta \in \mathbb{R}$. As $x \in \mathcal{S}_p$, there exists a constant $C(x) > 0$ such that $|x - \alpha| \geq 2^{-C(x)L(\alpha)^p}$ for any $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$. Now, for all but finitely many $\gamma \in \overline{\mathbb{Q}} \cap \mathbb{R}$, it follows that

$$|y - \gamma| = |x - (\gamma - \beta)| \geq 2^{-C(x)L(\gamma-\beta)^p}$$

as $\gamma - \beta \in \overline{\mathbb{Q}} \cap \mathbb{R}$ for any real algebraic number γ. From Lemma 1.2.7 we know that $L(\gamma - \beta) \leq 2L(\gamma) + 2L(\beta) + 2C_{\text{add}}$. Now, since β is fixed, there are only finitely many $\gamma \in \overline{\mathbb{Q}} \cap \mathbb{R}$ with $L(\gamma) < L(\beta) + C_{\text{add}}$, as there is a finite number of formulas of bounded length, and each can define at most one real algebraic number. Hence, excluding only finitely many $\gamma \in \overline{\mathbb{Q}} \cap \mathbb{R}$, we may assume that $L(\gamma) \geq L(\beta) + C_{\text{add}}$, implying that $L(\gamma - \beta) \leq 2L(\gamma) + 2L(\beta) + 2C_{\text{add}} \leq 4L(\gamma)$. It follows that

$$|y - \gamma| \geq 2^{-C(x)L(\gamma-\beta)^p} \geq 2^{-4^p C(x)L(\gamma)^p}$$

for all but finitely many $\gamma \in \overline{\mathbb{Q}} \cap \mathbb{R}$. Setting $C(y) = 4^p C(x)$ we see that $y \in \mathcal{S}_p$. □

Corollary 1.5.3. *The complement $\mathcal{S}_p^C \subseteq \mathbb{R}$ is also closed under addition with real algebraic numbers $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$.*

Proof. Suppose to the contrary that for some $x \in \mathcal{S}_p^C$ there is an $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$ such that $x + \alpha \notin \mathcal{S}_p^C$. Then by definition $x + \alpha \in \mathcal{S}_p$, but $x + \alpha + (-\alpha) = x \notin \mathcal{S}_p$ and $-\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$, contradicting Lemma 1.5.2. □

Because some ETR formulas can define doubly exponentially small numbers, another corollary of Lemma 1.5.2 is the fact that no algebraic number is badly approximable.

Corollary 1.5.4. *The set $\mathcal{S}_p$ contains no algebraic numbers.*

Proof. We show that $0 \notin \mathcal{S}_p$. The result then follows immediately from Corollary 1.5.3. View the ETR formula

$$(X_n = X_{n-1} * X_{n-1}) \wedge (X_{n-1} = X_{n-2} * X_{n-2}) \wedge \ldots \wedge (X_1 = X_0 * X_0) \wedge (X_0 + X_0 = 1)\,,$$

defining $\alpha = 2^{-2^n}$ through X_n. Its length is $O(n \log n)$, so that $L = L(\alpha) = O(n \log n)$ and we have

$$|0 - \alpha| = \alpha = 2^{-2^n} < 2^{-2^{\sqrt{L}}} < 2^{-L^p}$$

for L, n suitably large. Note that we only have finitely many α with L or n too small. Hence, $0 \notin \mathcal{S}_p$ and so by Corollary 1.5.3, no real algebraic number is a member of $\mathcal{S}_p$. □

This shows that all real algebraic numbers must lie in $\mathcal{S}_p^C$. However, these are certainly not the only elements of $\mathcal{S}_p^C$.

Theorem 1.5.5. *The set $\mathcal{S}_p^C$ contains uncountably many real numbers. In particular, $\mathcal{S}_p^C$ contains transcendental numbers.*

Proof. Consider real numbers of the form

$$x = \sum_{k=0}^{\infty} a_k 2^{-2^k}$$

with $a_k \in \{0, 1\}$ for all $k \geq 0$. Through a diagonal argument, there are uncountably many such x. We will show that all these x lie in $\mathcal{S}_p^C$. Denote the partial sums as $\alpha_n = \sum_{k=0}^{n} 2^{-2^k}$. Note that the ETR formula

$$(Y = X_n + X_{n-1} + \ldots + X_0) \wedge (X_n = X_{n-1} * X_{n-1}) \wedge \ldots \wedge (X_1 = X_0 * X_0) \wedge (X_0 + X_0 = 1)$$

has length $O(n \log n)$ and defines α_n through Y if we leave out the X_k with $a_k = 0$ from the summation for Y, so that $L(\alpha_n) = O(n \log n)$. Now note that

$$|x - \alpha_n| = \sum_{k=n+1}^{\infty} a_k 2^{-2^k} \leq 2^{-2^n} < 2^{-2^{\sqrt{L}}} < 2^{-L^p}$$

for L, n suitably large. Note that we only have finitely many α_n with L or n too small. If there are infinitely many a_k equal to 1, there are infinitely many distinct α_n. Hence, $x \notin \mathcal{S}_p$. On the other hand, if there are finitely many a_k equal to 1, the number x is rational, and thus $x \notin \mathcal{S}_p$ by Corollary 1.5.4.

Thus, $\mathcal{S}_p^C$ contains uncountably many real numbers. As there are only countably many algebraic numbers, $\mathcal{S}_p^C$ must contain transcendental numbers as well. Explicitly, the number

$$x = \sum_{k=0}^{\infty} 2^{-2^k}$$

is an element of $\mathcal{S}_p^C$ which is transcendental due to a result of Mahler[30]. □

In the following sections we will mainly consider $\mathcal{S}_p$ with $p > 1$. The following lemma and corollary show that it is not interesting to study $\mathcal{S}_p$ for $p < 1$.

Lemma 1.5.6. *There exists an absolute constant C such that for any $x \in \mathbb{R}$ there are infinitely many $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$ of description length $L = L(\alpha)$ with*

$$|x - \alpha| < 2^{-CL}$$

Proof. If $x \in \mathbb{Q}$ then $x \in \overline{\mathbb{Q}} \cap \mathbb{R}$ and the result follows from Corollary 1.5.4. Hence, we may assume that $x \notin \mathbb{Q}$. Then, by Dirichlet's approximation theorem there are infinitely many $\frac{a}{b} \in \mathbb{Q}$ with $\left|x - \frac{a}{b}\right| < \frac{1}{b^2}$. Let $\alpha = \frac{a}{b} \neq 0$ be such an approximation. Without loss of generality assume that $a, b > 0$, as otherwise we can add a minus sign in the ETR formulas used below without altering the asymptotic number of symbols. From Remark 1.2.9 it follows that there exists an absolute constant $c > 0$ such that $L(\alpha) \leq c(1+\log a+\log b)$. Now, as $\left|x - \frac{a}{b}\right| < \frac{1}{b^2}$, we know that $a < b|x| + \frac{1}{b} \leq b|x| + 1 \leq b(|x|+1)$. Thus it follows that $L(\alpha) \leq c(1 + \log a + \log b) < c(1 + \log(b(|x| + 1)) + \log b) = c(1 + \log(|x| + 1) + 2\log b)$. As there are infinitely many possible approximations $\frac{a}{b}$, and a is bounded in terms of b, b can become arbitrarily large. Only consider those b large enough so that $L = L(\alpha) \leq c(1 + \log(|x| + 1) + 2\log b) < 3c \log b$. It follows that

$$|x - \alpha| = \left|x - \frac{a}{b}\right| < \frac{1}{b^2} < 2^{-\frac{2}{3c}L}.$$

As there are infinitely many such approximations, choosing $C = \frac{2}{3c}$ proves the lemma. □

Corollary 1.5.7. *For $p < 1$ we have $\mathcal{S}_p = \emptyset$.*

Proof. Suppose to the contrary that there is an $x \in \mathcal{S}_p$ for some $p < 1$. Then by definition there is a constant $C(x) > 0$ such that there are only finitely many $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$ of description length $L = L(\alpha)$ with

$$|x - \alpha| < 2^{-C(x)L^p}.$$

On the other hand, by Lemma 1.5.6, there are infinitely many $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$ of description length $L = L(\alpha)$ with

$$|x - \alpha| < 2^{-CL}.$$

As $p < 1$ and $C(x)$ is constant for fixed x, there are only finitely many $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$ with $CL(\alpha) \leq C(x)L(\alpha)^p$. Hence, for infinitely many $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$ of description length $L = L(\alpha)$ we have

$$|x - \alpha| < 2^{-CL} < 2^{-C(x)L^p},$$

contradicting our assumption that $x \in \mathcal{S}_p$. □

1.5.2 The measure of $\mathcal{S}_p^C$

Thus far we have only seen examples of real numbers not in $\mathcal{S}_p$. Perhaps surprisingly, this is the exception rather than the rule. For $p > 1$ the following theorem shows that, in a measure-theoretic sense, $\mathcal{S}_p$ contains almost all reals.

Theorem 1.5.8. *Let $p > 1$. The set $\mathcal{S}_p^C$ is Lebesgue-measurable and has Lebesgue measure 0.*

Proof. Through a counting argument, we show that the set $\mathcal{T}_p \subseteq \mathbb{R}$ of reals x for which there exist infinitely many $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$ of description length $L = L(\alpha)$ with

$$|x - \alpha| < 2^{-L^p}$$

is Lebesgue-measurable and has Lebesgue measure 0. This corresponds to all x for which we cannot choose $C(x) = 1$ to show that $x \in \mathcal{S}_p$. Hence, $\mathcal{S}_p^C \subseteq \mathcal{T}_p$ and thus, if we show that $\mathcal{T}_p$ is Lebesgue-measurable with Lebesgue measure 0, so is $\mathcal{S}_p^C$.

Define $\mathcal{A}_n \subseteq \overline{\mathbb{Q}} \cap \mathbb{R}$ as the set of real algebraic numbers α with $L(\alpha) = n$. As each defining ETR formula defines exactly one real algebraic number, and there are at most 2^n valid ETR formulas of length n, we have $|\mathcal{A}_n| \leq 2^n$. We also define

$$T_{p,n} = \bigcup_{L \geq n} \bigcup_{\alpha \in \mathcal{A}_L} \left(\alpha - 2^{-L^p}, \alpha + 2^{-L^p}\right) = \bigcup_{\substack{\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R} \\ L(\alpha) \geq n}} \left(\alpha - 2^{-L(\alpha)^p}, \alpha + 2^{-L(\alpha)^p}\right).$$

As each $\mathcal{A}_L$ is finite, the sets $T_{p,n}$ are countable unions of open intervals and therefore are Lebesgue measurable with Lebesgue measure

$$\lambda\left(T_{p,n}\right) \leq \sum_{L \geq n} \sum_{\alpha \in \mathcal{A}_L} 2 \cdot 2^{-L^p} \leq \sum_{L \geq n} 2^L 2^{1-L^p} = \sum_{L \geq n} 2^{1+L-L^p}.$$

As $p > 1$, for L sufficiently large we can bound $1 + L - L^p \leq -\frac{1}{2}L^p < -\frac{1}{2}L$. Thus, for n sufficiently large, we have

$$\lambda\left(T_{p,n}\right) \leq \sum_{L \geq n} 2^{1+L-L^p} < \sum_{L \geq n} 2^{-\frac{1}{2}L} = \left(\frac{1}{\sqrt{2}}\right)^n \frac{1}{1 - \frac{1}{2}\sqrt{2}},$$

so that $\lim_{n \to \infty} \lambda\left(T_{p,n}\right) = 0$. We claim that $\mathcal{T}_p \subseteq \bigcap_{n \geq 1} T_{p,n}$. Take any $x \in \mathcal{T}_p$, so that by definition there are infinitely many $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$ of description length $L = L(\alpha)$ with $|x - \alpha| < 2^{-L^p}$. As for any $n \in \mathbb{N}$ there are only finitely many α with $L(\alpha) < n$, there must also be such α with $L(\alpha) \geq n$. By definition of α we have $x \in \left(\alpha - 2^{-L(\alpha)^p}, \alpha + 2^{-L(\alpha)^p}\right)$, so that $x \in \mathcal{T}_{p,n}$ for any $n \geq 1$. It follows that $\mathcal{T}_p \subseteq \bigcap_{n \geq 1} T_{p,n}$. As $\bigcap_{n \geq 1} T_{p,n}$ is a countable intersection of Lebesgue-measurable sets, it itself is Lebesgue-measurable with Lebesgue measure

$$\lambda\left(\bigcap_{n \geq 1} T_{p,n}\right) \leq \inf_{n \geq 1} \lambda\left(T_{p,n}\right) = 0,$$

as $\lambda\left(T_{p,n}\right) \geq 0$ for all $n \geq 1$ and $\lim_{n \to \infty} \lambda\left(T_{p,n}\right) = 0$. Hence $\bigcap_{n \geq 1} T_{p,n}$ has Lebesgue measure 0. As $\mathcal{S}_p^C \subseteq \mathcal{T}_p \subseteq \bigcap_{n \geq 1} T_{p,n}$ is a subset of a Lebesgue-measurable set with Lebesgue measure 0, it follows that $\mathcal{S}_p^C$ is also Lebesgue-measurable with Lebesgue measure 0. □

Theorem 1.5.8 implies that almost any real number is an element of $\mathcal{S}_p$. However, as the proof is non-constructive, we still have no explicit examples of such a number. It is for instance not clear whether $\mathcal{S}_p$ contains any computable numbers, as these are countable and therefore have Lebesgue measure 0. Fortunately, we are able to show this through a different approach.

1.6 A Computable Element of $\mathcal{S}_p$

In this section we show the following result, which is the closest we got to finding a polynomial-time computable, badly approximable number.

Theorem 1.6.1. *There exists a computable, badly approximable number.*

To show this, we formulate an equivalent problem in terms of sequences of integers.

1.6.1 Badly approximable sequences

We change our view from a single badly approximable number, to a *badly approximable sequence.*

Definition 1.6.2. A badly approximable sequence $\{s_n\}_{n=1}^{\infty}$ is a sequence of non-negative integers with accompanying polynomial q such that all but finitely many n, for any $\beta \in \overline{\mathbb{Q}} \cap \left[s_n - \frac{1}{2}, s_n + \frac{1}{2}\right)$ we have $q(L(\beta)) \geq n$. A bounded badly approximable sequence satisfies the additional constraint that $s_n < 2^n$ for all n.

We will show that badly approximable numbers correspond to bounded badly approximable sequences. In particular, we show the following, which should be considered as the main result of this section.

Theorem 1.6.3. *The existence of a (polynomial-time) computable badly approximable number is equivalent to the existence of a (polynomial-time) computable bounded badly approximable sequence.*

In fact, there are polynomial-time reductions that can transform a badly approximable number into a bounded badly approximable sequence, and vice versa. This is an immediate consequence of the following two lemmata.

Lemma 1.6.4. *Let $\varepsilon \in [0,1)$ be badly approximable. Then the sequence defined by $s_n = \lfloor 2^n \varepsilon \rfloor$ is bounded badly approximable. That is, s_n is the first n bits of ε without the decimal point. Also, if ε is (polynomial-time) computable, so is $\{s_n\}_{n=1}^{\infty}$.*

Lemma 1.6.5. *Let $\{s_n\}_{n=1}^{\infty}$ be a bounded badly approximable sequence with accompanying polynomial of degree d. Define the index formula $\mathcal{I}(k) = 2^{(d+1)^k} - 1$ for all $k \in \mathbb{Z}_{\geq 0}$. As $\{s_n\}_{n=1}^{\infty}$ is bounded, we can define $\mathfrak{b}_i$ as the binary expansion of $s_{\mathcal{I}(i)}$, padded to $\mathcal{I}(i)$ bits by prepending zeroes. Then the real number with binary expansion*

$$\varepsilon = 0.00\mathfrak{b}_0 00\mathfrak{b}_1 00\mathfrak{b}_2 00\ldots$$

is badly approximable. Furthermore, if $\{s_n\}_{n=1}^{\infty}$ is (polynomial-time) computable, so is ε.

1.6.2 From number to sequence

We start by proving Lemma 1.6.4.

Lemma 1.6.4. *Let $\varepsilon \in [0,1)$ be badly approximable. Then the sequence defined by $s_n = \lfloor 2^n \varepsilon \rfloor$ is bounded badly approximable. That is, s_n is the first n bits of ε without the decimal point. Also, if ε is (polynomial-time) computable, so is $\{s_n\}_{n=1}^{\infty}$.*

Proof. As $\varepsilon \in [0,1)$, clearly we have $0 \leq s_n < 2^n$ for any $n \geq 1$. Furthermore, if ε is (polynomial-time) computable, then all of its bits are (polynomial-time) computable and so the sequence $\{s_n\}_{n=1}^{\infty}$ is also (polynomial-time) computable. It remains to show that $\{s_n\}_{n=1}^{\infty}$ is a badly approximable sequence.

Since ε is a badly approximable number, there is a polynomial p such that

$$|\varepsilon - \alpha| \geq 2^{-p(L(\alpha))}$$

for all but finitely many $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$. Let $n \geq 1$ be any positive integer and take any $\beta \in \overline{\mathbb{Q}} \cap \left[s_n - \frac{1}{2}, s_n + \frac{1}{2}\right)$. By definition of s_n, we have

$$|\varepsilon - 2^{-n}\beta| \leq |\varepsilon - 2^{-n} s_n| + 2^{-n}|s_n - \beta| < 2^{-n} + 2^{-n-1} < 2^{-n+1}.$$

Thus, as ε is badly approximable it follows that

$$2^{-n+1} > |\varepsilon - 2^{-n}\beta| \geq 2^{-p(L(2^{-n}\beta))},$$

implying that $n - 1 \leq p(L(2^{-n}\beta))$ for all but finitely many n, β. In particular, we exclude the finitely many n for which this does not hold for all β. Let ϕ_β be a formula of length $L(\beta)$ defining β through X. Through binary exponentiation we can construct a formula $\phi_{2^{-n}}$ of length $O(\log n)$ defining 2^{-n} through Y. Then the formula

$$\phi_{2^{-n}\beta} = (\phi_\beta) \wedge (\phi_{2^{-n}}) \wedge (Z = X * Y)$$

defines $2^{-n}\beta$ through Z. By re-encoding the variables so that the variables of ϕ_β keep the same length and the variables of $\phi_{2^{-n}}$ are encoded in $O(\log[L(\beta) + L(\phi_{2^{-n}})]) = O(\log[L(\beta) + \log(n)])$ bits, which is sufficient as there are at most $L(\beta) + L(\phi_{2^{-n}})$ variables in total, we see that

$$L(2^{-n}\beta) \leq L(\beta) + C(\log n) \cdot \log[L(\beta) + \log(n)]$$

for an absolute constant $C > 0$, independent of β or n. Say p is of degree d. Then, since $n-1 \leq p(L(2^{-n}\beta))$ there exists an absolute constant $D > 0$ such that for n sufficiently large,

$$L(\beta) + C(\log n) \cdot \log[L(\beta) + \log(n)] \geq L(2^{-n}\beta) \geq D n^{1/d}.$$

Now there exists an absolute constant $D' > 0$ such that $L(\beta) \geq D' n^{1/d}$ for all sufficiently large n. □

Remark 1.6.6. Note that the polynomial associated with the bounded badly approximable sequence $\{s_n\}_{n=1}^{\infty}$ is of at most the same degree as the associated polynomial of ε.

1.6.3 From sequence to number

We now prove the substantially more involved Lemma 1.6.5.

Lemma 1.6.5. *Let $\{s_n\}_{n=1}^{\infty}$ be a bounded badly approximable sequence with accompanying polynomial of degree d. Define the index formula $\mathcal{I}(k) = 2^{(d+1)^k} - 1$ for all $k \in \mathbb{Z}_{\geq 0}$. As $\{s_n\}_{n=1}^{\infty}$ is bounded, we can define $\mathfrak{b}_i$ as the binary expansion of $s_{\mathcal{I}(i)}$, padded to $\mathcal{I}(i)$ bits by prepending zeroes. Then the real number with binary expansion*

$$\varepsilon = 0.00\mathfrak{b}_0 00\mathfrak{b}_1 00\mathfrak{b}_2 00\ldots$$

is badly approximable. Furthermore, if $\{s_n\}_{n=1}^{\infty}$ is (polynomial-time) computable, so is ε.

Proof. Clearly the function $\mathcal{I}(k)$ is polynomial time computable, so that if $\{s_n\}_{n=1}^{\infty}$ is polynomial time computable, so is ε. Note that the definition of ε implies that

$$\varepsilon = \sum_{i=0}^{\infty} s_{\mathcal{I}(i)} 2^{-\sum_{j=0}^{i}(\mathcal{I}(j)+2)}.$$

To show that ε is badly approximable, we claim that

$$|\varepsilon - \alpha| \geq 2^{-L(\alpha)^{(d+1)^2}}$$

for all but finitely many $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$. We present a proof by contradiction and assume that there are infinitely many $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$ such that $|\varepsilon - \alpha| < 2^{-L(\alpha)^{(d+1)^2}}$. As for any $L \in \mathbb{N}$ there are only finitely many α with $L(\alpha) \leq L$, there must also be arbitrarily large $L \in \mathbb{N}$ such that there is an $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$ with $L(\alpha) = L$ and $|\varepsilon - \alpha| < 2^{-L^{(d+1)^2}}$.

For such an $L \in \mathbb{N}$, define $n \in \mathbb{N}$ such that $\sum_{j=0}^{n} (\mathcal{I}(j)+2) < L^{(d+1)^2} - 1 \leq \sum_{j=0}^{n+1} (\mathcal{I}(j)+2)$. As L is an integer we have $L^{(d+1)^2} \geq \left(\sum_{j=0}^{n} (\mathcal{I}(j)+2)\right) + 2$. Further, define $\varepsilon_n = \mathfrak{b}_n.00\mathfrak{b}_{n+1}00\mathfrak{b}_{n+2}00\ldots$ so that

$$\varepsilon_n = s_{\mathcal{I}(n)} + \sum_{i=n+1}^{\infty} s_{\mathcal{I}(i)} 2^{-\sum_{j=n+1}^{i}(\mathcal{I}(j)+2)} = \left(\varepsilon - \sum_{i=0}^{n-1} s_{\mathcal{I}(i)} 2^{-\sum_{j=0}^{i}(\mathcal{I}(j)+2)}\right) 2^{\sum_{j=0}^{n}(\mathcal{I}(j)+2)}$$

and analogously define

$$\alpha_n = \left(\alpha - \sum_{i=0}^{n-1} s_{\mathcal{I}(i)} 2^{-\sum_{j=0}^{i}(\mathcal{I}(j)+2)}\right) 2^{\sum_{j=0}^{n}(\mathcal{I}(j)+2)}.$$

As α is an approximation of ε, so is α_n an approximation of ε_n. In particular, we have

$$|\varepsilon_n - \alpha_n| = 2^{\sum_{j=0}^{n}(\mathcal{I}(j)+2)} |\varepsilon - \alpha| < 2^{\left(\sum_{j=0}^{n}(\mathcal{I}(j)+2)\right) - L^{(d+1)^2}} < 2^{-2} = \frac{1}{4}.$$

On the other hand, we also have

$$\left|\varepsilon_n - s_{\mathcal{I}(n)}\right| = \sum_{i=n+1}^{\infty} s_{\mathcal{I}(i)} 2^{-\sum_{j=n+1}^{i}(\mathcal{I}(j)+2)} < 2^{-2} = \frac{1}{4},$$

by noting that the binary expansion of $\sum_{i=n+1}^{\infty} s_{\mathcal{I}(i)} 2^{-\sum_{j=n+1}^{i}(\mathcal{I}(j)+2)}$ is $0.00b_{\mathcal{I}(n+1)}00b_{\mathcal{I}(n+2)}\ldots < 2^{-2}$. It follows that

$$\left|\alpha_n - s_{\mathcal{I}(n)}\right| \leq |\varepsilon_n - \alpha_n| + \left|\varepsilon_n - s_{\mathcal{I}(n)}\right| < \frac{1}{2}$$

and thus $\alpha_n \in \left[s_{\mathcal{I}(n)} - \frac{1}{2}, s_{\mathcal{I}(n)} + \frac{1}{2}\right)$. Note that as α is real algebraic, so is α_n, so that by definition of $s_{\mathcal{I}(n)}$ this implies that $q(L(\alpha_n)) \geq \mathcal{I}(n)$. As q is a polynomial of degree d, there exists an absolute constant D independent of α or n such that $L(\alpha_n) \geq D\mathcal{I}(n)^{1/d}$ for all but finitely many exceptions. To derive a contradiction we construct a defining ETR formula for α_n of length strictly less than $D\mathcal{I}(n)^{1/d}$.

Pick a defining ETR formula ϕ_0 of α with variables chosen such that if ϕ_0 is true then $X_0 = \alpha$. Then $L(\phi_0) = L$. Next, iteratively define ETR formulas ϕ_k defining X_k as

$$\phi_{i+1} = (\phi_i) \wedge \left(X_{i+1} = \left(2^{\mathcal{I}(i)+2} * X_i\right) - s_{\mathcal{I}(i)}\right).$$

Using induction on the definition of the ϕ_i, we see that for any $k \in \mathbb{N}$, ϕ_k defines the number

$$\left(\alpha - \sum_{i=0}^{k-1} s_{\mathcal{I}(i)} 2^{-\sum_{j=0}^{i}(\mathcal{I}(j)+2)}\right) 2^{\sum_{j=0}^{k-1}(\mathcal{I}(j)+2)}.$$

Hence, an ETR formula ϕ defining α_n is given by

$$\phi = (\phi_n) \wedge \left(Y = 2^{\mathcal{I}(n)+2} * X_n\right),$$

defining α_n through variable Y. Note that we can remove the variables X_k by substitution, as each of these variables is only used once. The formula then becomes

$$\phi = Y = \left(2^{\mathcal{I}(n)+2} * \left(2^{\mathcal{I}(n-1)+2} * \left(2^{\mathcal{I}(n-2)+2} * \left(\dots\left(2^{\mathcal{I}(0)+2} * X_0 - s_{\mathcal{I}(0)}\right)\dots\right) - s_{\mathcal{I}(n-2)}\right) - s_{\mathcal{I}(n-1)}\right)\right) \wedge (\phi_0).$$

Of course, we still need ETR formulas to define $2^{\mathcal{I}(i)+2}$ and $s_{\mathcal{I}(i)}$. The first can be defined by binary exponentiation in an ETR formula of at most $C' \log \mathcal{I}(i)$ symbols and at most as many variables for any $i \in \mathbb{Z}_{\geq 0}$ for some absolute constant C'. Thus, $2^{\mathcal{I}(0)+2}$ up to $2^{\mathcal{I}(n)+2}$ can be defined in one ETR formula of length

$$\left(C' \sum_{i=0}^{n} \log \mathcal{I}(i)\right) \log \left(L + C' \sum_{i=0}^{n} \log \mathcal{I}(i)\right),$$

where we encode the variables such as to not interfere with those in ϕ_0. To define $s_{\mathcal{I}(i)}$, note that $s_{\mathcal{I}(i)} < 2^{\mathcal{I}(i)}$ as $\{s_n\}_{n=1}^{\infty}$ is a bounded badly approximable sequence. Hence, Lemma 1.2.8 implies that $s_{\mathcal{I}(i)}$ can be expressed by a formula of at most $C''\mathcal{I}(i)$ symbols and no variables for any $i \in \mathbb{N}$ for some absolute constant C''. Substituting the expressions for $s_{\mathcal{I}(i)}$ to avoid the need for any additional variables, we see that

$$L(\alpha_n) \leq L(\phi) \leq L + Cn + \left(C' \sum_{i=0}^{n} \log \mathcal{I}(i)\right) \log \left(L + C' \sum_{i=0}^{n} \log \mathcal{I}(i)\right) + C'' \sum_{i=0}^{n-1} \mathcal{I}(i),$$

choosing the explicit absolute constant $C > 0$ to account for the constant number of additional brackets and other operators in ϕ. Note that the term $L = L(\alpha)$ comes from the defining formula ϕ_0 of α. In this final comparison we utilize the precise definition of $\mathcal{I}(j)$. In particular, by choice of n, we have $L^{(d+1)^2} - 1 \leq \sum_{j=0}^{n+1}(\mathcal{I}(j)+2) = \sum_{j=0}^{n+1}(2^{(d+1)^j}+1) \leq 2 \cdot 2^{(d+1)^{n+1}}$ so that $L \leq 2^{(d+1)^{n-1}+1}$ for L sufficiently large. Combining this with the bounds on $L(\alpha_n)$ derived above, we find that

$$D\mathcal{I}(n)^{1/d} \leq L(\alpha_n) \leq 2^{(d+1)^{n-1}+1} + Cn + \left(C' \sum_{i=0}^{n} \log \mathcal{I}(i)\right) \log \left(2^{(d+1)^{n-1}+1} + C' \sum_{i=0}^{n} \log \mathcal{I}(i)\right) + C'' \sum_{i=0}^{n-1} \mathcal{I}(i).$$

From the definition of $\mathcal{I}(k) = 2^{(d+1)^k} - 1$ we see that there is an absolute constant $D' > 0$ so that

$$2^{(d+1)^{n-1}+1} + Cn + \left(C' \sum_{i=0}^{n} \log \mathcal{I}(i)\right) \log \left(2^{(d+1)^{n-1}+1} + C' \sum_{i=0}^{n} \log \mathcal{I}(i)\right) + C'' \sum_{i=0}^{n-1} \mathcal{I}(i) \leq D' 2^{(d+1)^{n-1}}$$

for sufficiently large n. This implies that

$$D\left(2^{(d+1)^n} - 1\right)^{1/d} = D\mathcal{I}(n)^{1/d} \leq L(\alpha_n) \leq D' 2^{(d+1)^{n-1}}.$$

However, the left hand side grows asymptotically faster than the right hand side, and as we can choose L arbitrarily large, n can also become arbitrarily large. This is the desired contradiction. Thus, ε is badly approximable. □

Remark 1.6.7. While the proof of Lemma 1.6.4 yields an associated polynomial of the same degree as the original associated polynomial, in the above proof of Lemma 1.6.5 the degree is increased from d to $(d+1)^2$. Although this can be slightly improved by a tighter analysis, improving the degree to d is not possible with the same proof strategy. The length of the defining formulas of $s_{\mathcal{I}(0)}$ up to $s_{\mathcal{I}(n-1)}$ required to isolate $s_{\mathcal{I}(n)}$ would dominate the inequality so that the asymptotic argument fails.

1.6.4 A computable badly approximable sequence

Clearly, Theorem 1.6.3 on the equivalence between badly approximable sequences and badly approximable numbers follows from Lemma 1.6.4 and Lemma 1.6.5. Thus, to show $\mathcal{S}_p$ contains a computable number, it suffices to find a computable bounded badly approximable sequence.

Proof of Theorem 1.6.1. It suffices to find a computable bounded badly approximable sequence. We claim picking s_n as the smallest non-negative integer for which $L(\alpha) \geq n$ for all $\alpha \in \overline{\mathbb{Q}} \cap \left[s_n - \frac{1}{2}, s_n + \frac{1}{2}\right)$ works. Clearly this is a badly approximable sequence. To see that it is bounded, a simple counting argument suffices: the 2^n intervals $\left[k - \frac{1}{2}, k + \frac{1}{2}\right)$ for $0 \leq k < 2^n$ are disjoint, and there are at most $\sum_{i=0}^{n-1} 2^i < 2^n$ real algebraic numbers α with $L(\alpha) < n$. Thus at least one of the integers $0 \leq k < 2^n$ must have $L(\alpha) \geq n$ for all $\alpha \in \overline{\mathbb{Q}} \cap \left[k - \frac{1}{2}, k + \frac{1}{2}\right)$ and so $0 \leq s_n < 2^n$. Finally, computability follows from the fact that we can enumerate all formulas ϕ of length $L(\phi) < n$ with a variable X and check which of the formulas $\left(k - \frac{1}{2} \leq X < k + \frac{1}{2}\right) \wedge (\phi)$ is in ETR. □

Remark 1.6.8. While the proof of Theorem 1.6.1 yields a computable badly approximable number, this number is likely not polynomial time computable, as the procedure of finding these minimal s_n seems to be as hard as deciding ETR.

Now suppose that we know a certain number x to be polynomial-time computable and badly approximable. By Lemma 1.6.4, the truncations $\lfloor 2^n x \rfloor$ of the binary expansions of x form a bounded polynomial-time computable, badly approximable sequence. In particular, this also implies that the SLP length $\lfloor 2^n x \rfloor$ has some non-trivial lower bound. For instance, if we could show $x = \ln 2$ to be badly approximable, it would follow that $\lfloor 2^n \ln x \rfloor$ is not easy to compute, something that is still an open problem[15]. Certainly, as so little is known about non-easy to compute sequences in terms of SLP length, this once again illustrates the difficulty of determining a specific badly approximable number.

1.6.5 A different way to construct a computable element

In the previous section, we saw a construction of a computable, badly approximable number which was relatively easy to see by the equivalence with badly approximable sequences. We now present a different construction of a computable, badly approximable number x that follows more immediately from the definition of a badly approximable number.

The idea is to construct x iteratively, by first avoiding all $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$ with $L(\alpha) = 1$, then all with $L(\alpha) = 2$, and so forth. More specifically, we define a sequence of intervals $\{I_n\}_{n=1}^{\infty}$ iteratively as follows. Fix an encoding of formulas. To start we take $I_1 = [0, 1]$. To derive I_{n+1}, we begin by computing all of the $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$ with $L(\alpha) = n$ by enumerating all 2^n possible formulas with our fixed encoding. For simplicity of calculation we assume that certainly one of these formulas does not define an α, for instance because it contains malformed parentheses. There are therefore less than 2^n algebraic reals α with $L(\alpha) = n$.

We now split I_n into 2^n equally sized intervals, and pick the leftmost interval that does not contain any α with $L(\alpha) = n$. We then halve the size of this interval around its midpoint, and set that as I_{n+1}. As there are fewer than 2^n such α, such an interval exists. Now the sequence $\{I_n\}_{n=1}^{\infty}$ is computable, and if we define x_n as the midpoint of I_n, we can take $x = \lim_{n\to\infty} x_n$ which then also is computable. It remains to show that x is indeed badly approximable.

Inductively, we see that $|I_n| = \prod_{i=2}^{n} 2^{-i} = 2^{(2-n-n^2)/2}$. By the way I_{n+1} is constructed, any element of I_{n+1} has distance at least $\frac{1}{2}|I_{n+1}| = 2^{(-2-3n-n^2)/2}$ to any algebraic real α with $L(\alpha) = n$. As $x \in I_n$ for all n by construction, we have that

$$|x - \alpha| \geq 2^{(-2-3L(\alpha)-L(\alpha)^2)/2} \geq 2^{-3L(\alpha)^2}$$

for all algebraic reals α. Indeed, x is badly approximable.

Chapter 2

The Problem Of ε-Rudi

Consider drawing a number of disks of unit diameter in the plane, and connecting the midpoints of two intersecting disks by an edge. In this way, the midpoints induce what is called a *unit disk intersection graph*. Between two midpoints with distance d there will be an edge if $d \leq 1$, and no edge if $d > 1$. Given a set of midpoints, the induced unit disk intersection graph is straightforward to compute; loop over all pairs of midpoints and compare their distance with 1.

One might wonder whether all graphs can be realized as a unit disk intersection graph. It turns out that this is not the case. For instance, consider the star graph on 7 vertices as in Figure 2.1. No matter how the vertices are positioned in the plane, two of the disks corresponding to the leaves will intersect.

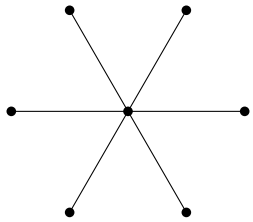


Figure 2.1: A graph without a unit disk intersection representation.

The fact that not every graph can be realized as a unit disk intersection graph gives rise to the inverse problem of Unit Disk Intersection Graph Recognition, or Udi for short; given any graph, determine whether it has a unit disk intersection representation. This problem, perhaps indicated by the fact that it has a proper name, turns out to be quite complex and difficult. However, from a complexity theoretic standpoint Udi has been largely figured out, in the sense that it is known to be $\exists\mathbb{R}$-complete[9][1][2]. This means that Udi is just as hard as the problem of recognizing ETR. To see where this relation with ETR comes from, note that for a given graph it is simple to derive a formula whose satisfiability is equivalent to the graph being realizable as a unit disk intersection graph. Namely, we introduce free variables for each of the midpoint coordinates and take the formula to be the conjunction of all the distance constraints imposed by the graph.

Instead of Udi, which has been studied in detail and whose complexity is known, we instead look at a robust version of the problem known as ε-Rudi, short for ε-Robust Unit Disk Intersection Graph Recognition. Instead of deciding whether a graph can be realized as a unit disk intersection graph, we ask whether the graph can be realized as an *ε-robust unit disk intersection graph*. Still we draw a number of disks of unit diameter in the plane, and draw edges between the midpoints of any two intersecting disks. This time however, to be considered ε-robust, we require that any two non-intersecting disks differ by a distance of at least ε. Thus, a graph can be realized as an ε-robust unit disk intersection graph if and only if there is a set of points corresponding to the vertices of the graph such that two points with distance d have $d \leq 1$ if they share an edge, and $d > 1 + \varepsilon$ if they do not. No two points are allowed to have a distance between 1 and $1 + \varepsilon$. Hereafter, being realizable as an ε-robust unit disk intersection graph will simply be denoted as having an *ε-robust representation*. Again, we can derive an ETR formula $\Psi_G(\varepsilon)$ whose satisfiability corresponds to whether G has an ε-robust representation. Encoding the n vertices of a graph G by coordinates $X_1, Y_1, \ldots, X_n, Y_n$, we can for instance take

$$\Psi_G(\varepsilon) = \exists X_1, Y_1, \ldots, X_n, Y_n(((X_1 - X_2) * (X_1 - X_2) + (Y_1 - Y_2) * (Y_1 - Y_2) > (1 + \varepsilon)^2) \wedge \ldots),$$

where we take the conjunction of all distance constraints imposed by G. Note that this does not necessarily imply $\exists\mathbb{R}$-membership of ε-Rudi, as ε need not be representable through the constants 0 and 1.

An intuitive idea would be to include ε in the input of ε-RUDI, together with the graph. However, in this manner we make the problem at least as difficult as UDI, as only inputting $\varepsilon = 0$ is equivalent to UDI. Instead, we include ε in the problem statement, so that we define a family of algorithmic problems such as $\frac{1}{3}$-RUDI or $(\sqrt{2}-1)$-RUDI, or even $(\pi - e)$-RUDI. In this way, we can better study how a specific choice of ε influences the complexity of ε-RUDI.

Indeed, we will see that the choice of ε greatly influences the complexity of ε-RUDI, exhibiting a family of different quirks in complexity theory. This chapter serves to explore the beautiful zoo of problems defined by ε-RUDI, and we will see examples of ε for which ε-RUDI is trivial, remains $\exists\mathbb{R}$-complete, or even becomes undecidable. We will also see that the conjectured existence of a polynomial-time computable, badly approximable ε^* would imply that ε^*-RUDI is a member of NP. This is fascinating, as it would imply that adding robustness to a problem can in some cases make the problem easier, while not making the problem trivial.

2.1 A Simple Range

The first set of problems in the family of ε-RUDI that we study are those in a so-called *simple range*. Observe that the set of graphs with an ε-robust representation becomes smaller when ε increases, as the distance constraints on the midpoints only become stricter. It turns out that for large enough ε, the set of graphs with an ε-robust representation becomes so simple that it can be recognized in polynomial time. That is, for large enough ε the problem of ε-RUDI becomes polynomial-time solvable. To illustrate this idea, we show the following lemma.

Lemma 2.1.1. *Let $\varepsilon > 1$ and let G be any graph. Then G has an ε-robust representation if and only if G consists of a set of disjoint cliques.*

Proof. If G consists of a set of disjoint cliques, an ε-robust representation is given by placing all vertices of a single clique in the same point, and positioning these points sufficiently far apart.

On the other hand, suppose that G has an ε-robust representation with $\varepsilon > 1$. In this representation, two vertices with distance d have $d \leq 1$ if they share an edge in G, and $d > 2$ if they do not. In particular, any two vertices that share a neighbour have distance $d \leq 2$ by the triangle inequality, and hence must themselves also share an edge. This implies that G consists of a set of disjoint cliques. □

In the proof of Lemma 2.1.1 we place all of the vertices corresponding to a single clique at the same point. An important observation is that we can always do this; if two incident vertices are *twins*, that is they share the same closed neighbourhood, then without loss of generality we can position the corresponding midpoints in the same position. We can therefore ignore twins in a graph without loss of generality. The statement that G consists of a set of disjoint cliques is equivalent to every pair of vertices that share an edge being twins.

As checking whether two vertices of G are twins can be done in polynomial time by checking whether every vertex is a neighbour of either both or none of the two vertices, we can also check whether G consists of a set of disjoint cliques in polynomial time. That is, if $\varepsilon > 1$, then ε-RUDI is solvable in polynomial time. In [5] the following, slightly stronger result is shown.

Theorem 2.1.2. *For each $\varepsilon > \frac{\sqrt{6}+\sqrt{2}}{2} - 1 \approx 0.932$, ε-RUDI is polynomial time solvable.*

The crux of the proof of Theorem 2.1.2 is that for such $\varepsilon > \frac{\sqrt{6}+\sqrt{2}}{2} - 1$, assuming twins are ignored, all vertices of an ε-robust graph have degree at most 2. Then the graph consists of a disjoint set of cycles and paths, which can again easily be verified in polynomial time.

2.2 Undecidability of ε-Rudi

In this section we will slightly abuse notation and denote a graph and a representation of that graph with the same symbol. We will also use either the midpoint or a disk to represent a vertex and freely intermingle those two representations.

We saw that ε-RUDI is solvable in polynomial time for large enough ε. This makes it relatively easy compared to the $\exists\mathbb{R}$-complete base problem. However, this is not the case for all ε. In fact, we show that for almost all small ε, the problem of ε-RUDI is undecidable.

Theorem 2.2.1. *ε-RUDI is decidable for only countably many $0 \leq \varepsilon < \frac{2}{\sqrt{3}} - 1 \approx 0.155$.*

The proof of Theorem 2.2.1 stems from the fact that there are uncountably many $\varepsilon \in [0, \frac{2}{\sqrt{3}} - 1)$, while there are only countably many distinct algorithms. The goal is to show that any such ε requires a unique algorithm, which follows from the following lemma.

Lemma 2.2.2. *For any two $0 \leq \varepsilon < \varepsilon' < \frac{2}{\sqrt{3}} - 1$, there exists a graph G with $G \in \varepsilon$-RUDI and $G \notin \varepsilon'$-RUDI.*

The main idea behind the graph constructed for Lemma 2.2.2 is to enclose a set of vertices that do not have any edges between them with a connected cycle of vertices. As vertices that do not share an edge should have a distance greater than $1+\varepsilon$, this corresponds to packing disks of diameter $1+\varepsilon$ in an area of bounded perimeter. The construction is chosen such that for the larger ε', these disjoint disks no longer fit inside the connected cycle of vertices.

A more intricate depiction of the construction is given in Figure 2.2. As depicted, the construction consists of several parts, labeled $L_{k,1}, L_{k,2}$, M and C. The part labeled C is the chain of vertices that surrounds the set of vertices without any edges between them, corresponding to the disjoint disks of diameter $1+\varepsilon$. The sets $L_{k,1}$ and $L_{k,2}$ contain these sets of vertices. The underlying graphs of $L_{k,1}$ and $L_{k,2}$ are exactly the same; we require two copies to ensure one of the two sets lies on the *interior* of the bounding cycle C, so that we can reason about the cycle being too short for the larger disks of diameter $1+\varepsilon'$. Besides having two copies of the set of disjoint vertices, the part labeled M is required to connect the sets $L_{k,1}$ and $L_{k,2}$ and the cycle C in such a way that ensures one of the $L_{k,1}$ and $L_{k,2}$ is indeed on the interior of C.

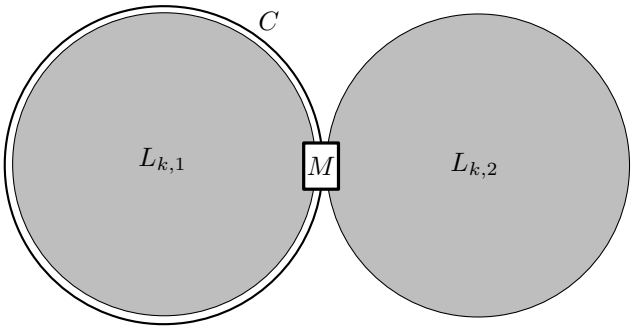


Figure 2.2: General overview of the construction for G_k.

We start with the part that is labeled M in Figure 2.2. We call this graph a *cloverleaf gadget*, as depicted in Figure 2.3. A cloverleaf gadget is a graph that forces the order of the four edges e_1, e_2, e_3 and e_4 connected to the center vertex up to mirror symmetry in any disk intersection graph representation. In other words, in any ε-robust representation of the cloverleaf gadget, e_1 is opposite e_3, and e_2 is opposite e_4.

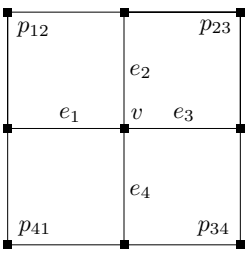


Figure 2.3: A cloverleaf gadget.

The cloverleaf gadget works by connecting each of the four pairs of adjacent edges by completing the cycle with a distinct chain of connected vertices. In this case we create a cloverleaf gadget with cycles of length four. While in this case, the cloverleaf gadget is simply a 3×3 grid, when these connecting cycles are made longer, the resulting graph looks like a cloverleaf, hence the gadget's name. Formally, we describe the functionality of a cloverleaf gadget as follows.

Lemma 2.2.3. *If a graph G contains four edges e_1, e_2, e_3 and e_4 are connected to a common vertex v, and there are paths p_{12}, p_{23}, p_{34} and p_{41} of length at least 2 connecting the other endpoints of e_1, e_2, e_3 and e_4 as in Figure 2.3, then in any representation of G the edges e_1, e_2, e_3 and e_4 are drawn either in clockwise or counterclockwise order around v.*

To show Lemma 2.2.3, the following lemma is useful.

Lemma 2.2.4. *If a convex 4-gon P has two opposite sides of length $\geq s$, then at least one diagonal also has length $\geq s$.*

Proof. Suppose to the contrary that both diagonals have length $< s$. Label the vertices of P as A, B, C and D in order, so that $|AB| \geq s$ and $|CD| \geq s$, as depicted in Figure 2.4. As P is convex, the diagonals AC and BD intersect each other, call this intersection S. By the triangle inequality we have

$$2s > |AC| + |BD| = |AS| + |BS| + |CS| + |DS| \geq |AB| + |CD| \geq 2s,$$

which is a clear contradiction. □

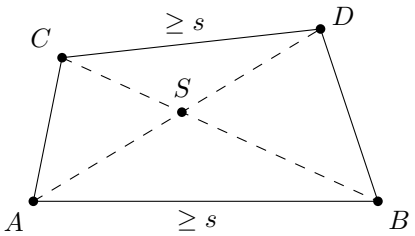


Figure 2.4: A graphical depiction of the proof of Lemma 2.2.4.

We now show Lemma 2.2.3.

Proof of Lemma 2.2.3. Suppose there is a geometric representation of the cloverleaf gadget in which the four central edges are not in their desired order, that is, e_1 is not opposite e_3 or e_2 is not opposite e_4. Without loss of generality we may assume that e_1 is opposite e_2 and e_3 is opposite e_4. Now, no matter how the four paths p_{12}, p_{23}, p_{34} and p_{41} that complete the cycles of the cloverleaf gadget are positioned, two of these paths have to intersect. Hence there are two graph edges of the cloverleaf that intersect in this representation. However, besides these two intersecting edges, there are no other edges between the four vertices involved. This is impossible by Lemma 2.2.4, as the four vertices would form a convex 4-gon with all sides at least $1 + \varepsilon$ long but both diagonals of length at most 1. This shows that the cloverleaf gadget indeed fixes the order of the edges connected to the center vertex up to mirror symmetry. □

We now explain how the graphs are constructed for $L_{k,1}$ and $L_{k,2}$ as in Figure 2.2. Consider the hexagonal lattice $\Lambda \subseteq \mathbb{R}^2$ generated by the vectors $u = (1 + \varepsilon, 0)$ and $v = (-\frac{1}{2}(1 + \varepsilon), \frac{1}{2}\sqrt{3}(1 + \varepsilon))$, that is,

$$\Lambda = \{au + bv : a, b \in \mathbb{Z}\}.$$

Note that $|u| = |v| = 1 + \varepsilon$. Furthermore, we define Λ_k as all points in Λ with distance at most $k(1 + \varepsilon)$ from the origin. In our construction, as mentioned before, these points will correspond to mutually disjoint disks of diameter $1 + \varepsilon$. To ensure these disjoint disks have to all lie on the interior of C, we have to join these disjoint vertices by auxiliary vertices. For this, we add midpoints between all pairs of points in Λ_k with distance $1 + \varepsilon$. This is illustrated in Figure 2.5. We denote this new point set by L_k, consisting of the points of Λ_k together with these midpoints. As L_k is a subset of the hexagonal lattice generated by the vectors $u/2$ and $v/2$, the pairwise distances are either at least $1 + \varepsilon$, or at most $\frac{1}{2}\sqrt{3}(1 + \varepsilon) \leq 1$, since $\varepsilon \in [0, \frac{2}{\sqrt{3}} - 1]$. This is where the bound on ε in Theorem 2.2.1 comes from. Hence, L_k induces a corresponding graph $L_k \in \varepsilon$-RUDI. Both $L_{k,1}$ and $L_{k,2}$ are copies of this induced graph.

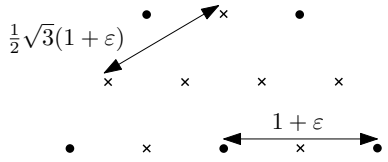


Figure 2.5: A section of L_k, dots are elements of Λ_k.

The part labeled C in Figure 2.2 is a cycle whose length will be chosen specifically so that the resulting graph is representable for ε, but not for ε'. For this, we need to bound the minimal length of a cycle that contains a set of disjoint disks. The following lemma serves this purpose.

Lemma 2.2.5. *Suppose $n \geq 2$ disks of diameter d are pairwise disjoint in the plane. Then any simple closed curve C enclosing all n disks has length at least $d\sqrt{n\pi\sqrt{12}}$.*

Proof. Let A denote the area enclosed by C. Without loss of generality, we may assume that A is convex, as the convex hull of any connected domain has a perimeter bounded above by the original perimeter. Due to Fejes Tóth [31], if a convex domain contains $n \geq 2$ unit circles, it has area at least $n\sqrt{12}$. Scaling by $\frac{d}{2}$, we see that A has area at least $\frac{1}{4}d^2 n\sqrt{12}$. The result follows from the isoperimetric inequality, which states that for any domain of perimeter L and area F we have $L^2 \geq 4\pi F$ [31]. □

With Lemma 2.2.5, we can now prove Lemma 2.2.2. This proof will also clarify how C is specifically chosen and how the different parts $L_{k,1}, L_{k,2}$, M and C in Figure 2.2 are connected.

Proof of Lemma 2.2.2. We construct a sequence of graphs G_k, parameterized by an integer $k \geq 2$, one of which we will later choose as G. Suppose $k \geq 2$ is fixed. To construct G_k, we start with an ε-robust representation, from which we infer the graph. Thus, all the constructions we give will be in terms of points in the plane, and not vertices. The graph G_k is inferred from this representation. As mentioned before, to ensure a representation of L_k is contained in the interior of C and not the exterior, we have to add two copies of L_k so that at least one will be in the interior of C. We connect C and the two copies $L_{k,1}$ and $L_{k,2}$ by a cloverleaf gadget M. An overview of this construction was already given in Figure 2.2.

A more detailed depiction of how $L_{k,1}, L_{k,2}, C$ and M are interconnected can be seen in Figure 2.7. The points with which $L_{k,1}$ and $L_{k,2}$ are connected to M are $(k(1+\varepsilon), 0)$ and $(-k(1+\varepsilon), 0)$ in their two original lattices, that is their right- and leftmost point, respectively. As $\Lambda_k \subseteq D(k(1+\varepsilon))$, all other points of Λ_k will have an x-coordinate less than $k(1+\varepsilon)$, or more than $-k(1+\varepsilon)$, respectively. Due to the structure of Λ_k this in fact implies that all other points differ by at least $\frac{1}{2}(1+\varepsilon)$ in the x-coordinate. Therefore so do all points of L_k, except for two midpoints induced by the vertices connected to M.

As we do not rotate $L_{k,1}$ or $L_{k,2}$ in this representation, it is now relatively straightforward to verify that all distances between points of $L_{k,1}, L_{k,2}$ and M are either at most 1 or at least $1+\varepsilon$. Furthermore, from the above we can deduce that if we choose points of C above or below M on the middle vertical, as depicted in Figure 2.7, then the distance from points of C to points of $L_{k,1}$ and $L_{k,2}$ is at least $1 + \frac{1}{2}(1+\varepsilon) \geq 1+\varepsilon$.

What remains is how we specifically choose the points of C. For this, take the circle Γ of radius $(k+1)(1+\varepsilon)$ around $L_{k,1}$. As $L_{k,1} \subseteq \text{Disk}(0, k(1+\varepsilon))$, any point on Γ has distance at least $1+\varepsilon$ to any point of $L_{k,1}$. To guarantee that the points we choose for C also have distance at least $1+\varepsilon$ to $L_{k,2}$, we slightly modify the circle by intersecting it with the middle vertical through M. An exaggerated version of the resulting curve Γ' can be seen in Figure 2.6, and has length at most $2\pi(k+1)(1+\varepsilon)$.

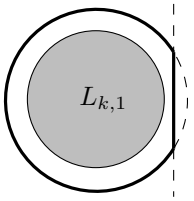


Figure 2.6: The curve from which we choose points for C.

Thus, we can pick $\lceil 2\pi(k+1)(1+\varepsilon)\rceil + 1$ points as the cycle C, with three points belonging to M as in Figure 2.7, and the rest equally spaced on Γ', so that the distance between two consecutive points is at most 1. Further, two nonconsecutive points of C will have distance at least $1+\varepsilon$. We may finally conclude that our construction induces a unit disk intersection graph $G_k \in \varepsilon$-RUDI.

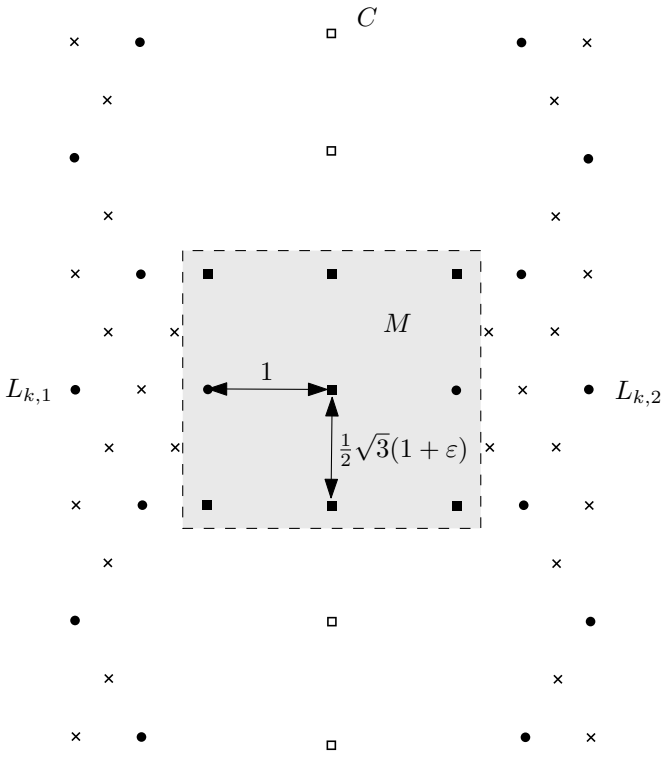


Figure 2.7: Connecting the different parts of G_k.

We conclude by showing that $G_k \notin \varepsilon'$-RUDI for some large enough $k \geq 2$. Suppose that $G_k \in \varepsilon'$-RUDI, and consider any ε'-robust representation. As the only edges in C are between consecutive points, the points of C together with these edges in the representation form a closed curve that must be simple by Lemma 2.2.4. Denote by v_1 and v_2 the points of $L_{k,1}$ and $L_{k,2}$ that are a part of the cloverleaf gadget M. Due to the order-preservation of the cloverleaf gadget, one of these two points lies in the interior of the curve formed by C. Suppose without loss of generality this is point v_1. Now define the simple closed curve C' from C, by taking a slightly different path through M, as depicted in Figure 2.8. Due to the cloverleaf gadget, the vertex v_1 also lies on the interior of C'. Furthermore, as none of the vertices in $L_{k,1}$ have any edges with points of C', and as the graph $L_{k,1}$ is connected, Lemma 2.2.4 implies that the vertices of $L_{k,1}$ must all lie on the interior of C'.

We will finally make use of Lemma 2.2.5. Centered at each point of $\Lambda_{k,1}$ we draw a disk of diameter $1+\varepsilon'$. As there are no edges between the vertices of $\Lambda_{k,1}$, these disks are all disjoint. Further, these disks are all fully contained in the interior of C'. Suppose to the contrary that there is some disk centered at $w \in \Lambda_{k,1}$ that intersects C' in a point s, then there is a vertex v on C that has distance at most $\frac{1}{2}$ to s, but the triangle inequality then implies that $|v-w| \leq |v-s| + |w-s| \leq \frac{1}{2} + \frac{1}{2}(1+\varepsilon') < 1+\varepsilon'$ which is impossible due to the non-edge between v and w. Therefore we may apply Lemma 2.2.5. As each edge segment has length at most 1, we can bound the length of C' by $\lceil 2\pi(k+1)(1+\varepsilon)\rceil + 3$ which, combined with Lemma 2.2.5, yields

$$2\pi(k+1)(1+\varepsilon) + 4 \geq \lceil 2\pi(k+1)(1+\varepsilon)\rceil + 3 \geq (1+\varepsilon')\sqrt{|\Lambda_{k,1}|\pi\sqrt{12}}.$$

Reordering terms yields the following relation between $1+\varepsilon'$ and $1+\varepsilon$.

$$\frac{1+\varepsilon}{1+\varepsilon'} \geq \frac{\sqrt{|\Lambda_{k,1}|\sqrt{3}}}{\sqrt{2\pi}(k+1)} - \frac{4}{2(1+\varepsilon')\pi(k+1)}. \tag{2.2.1}$$

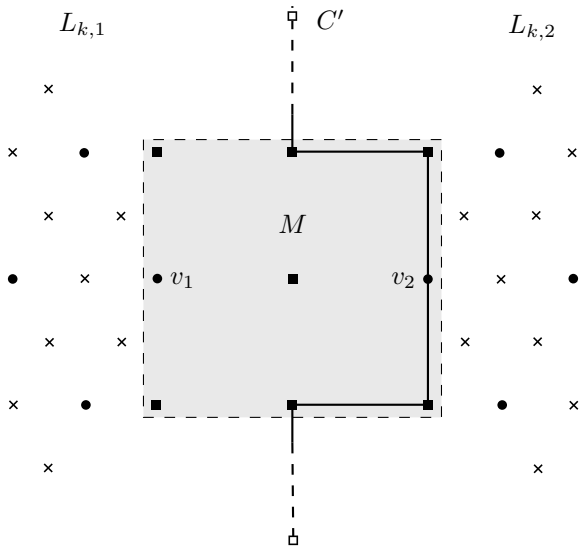


Figure 2.8: The modified curve C'.

We are interested in the limiting behaviour of the right-hand side. As k grows, the latter term becomes negligibly small, so that we only need to consider the former term. For this we require an expression for $|\Lambda_{k,1}|$. By transforming the lattice to a square lattice and counting the number of solutions to the diophantine equation $x^2 + xy + y^2 \leq k$, Taylor[32] has shown that

$$|\Lambda_{k,1}| = 1 + 6\sum_{n=0}^{\infty}\left(\left\lfloor\frac{k^2}{3n+1}\right\rfloor - \left\lfloor\frac{k^2}{3n+2}\right\rfloor\right).$$

We provide an asymptotic lower bound for $|\Lambda_{k,1}|$. Clearly, each term is non-negative so that

$$|\Lambda_{k,1}| \geq 6\sum_{n=0}^{k-1}\left(\left\lfloor\frac{k^2}{3n+1}\right\rfloor - \left\lfloor\frac{k^2}{3n+2}\right\rfloor\right).$$

Now, as we only sum k terms and $x - 1 < \lfloor x \rfloor \leq x$ it follows that we have a lower bound of the form

$$|\Lambda_{k,1}| \geq 6k^2\sum_{n=0}^{k-1}\left(\frac{1}{3n+1} - \frac{1}{3n+2}\right) - O(k) = 6k^2\sum_{n=0}^{\infty}\left(\frac{1}{3n+1} - \frac{1}{3n+2}\right) - O(k).$$

The latter equality follows from the fact that we can roughly bound

$$\sum_{n=k}^{\infty}\left(\frac{1}{3n+1} - \frac{1}{3n+2}\right) = \sum_{n=k}^{\infty}\frac{1}{(3n+1)(3n+2)} \leq \sum_{n=k}^{\infty}\frac{1}{n(n+1)} = \frac{1}{k},$$

where the latter equality follows from the fact that the sum telescopes. We calculate the infinite sum in the lower bound for $|\Lambda_{k,1}|$ substituting in an integral.

$$\sum_{n=0}^{\infty}\left(\frac{1}{3n+1} - \frac{1}{3n+2}\right) = \sum_{n=0}^{\infty}\int_0^1 x^{3n} - x^{3n+1}\mathrm{d}x = \sum_{n=0}^{\infty}\int_0^1 (1-x)x^{3n}\mathrm{d}x.$$

Clearly $(1-x)x^{3n}$ is positive for all $x \in (0,1)$ so that by Tonelli's theorem we may interchange integration and summation to obtain

$$\sum_{n=0}^{\infty}\left(\frac{1}{3n+1} - \frac{1}{3n+2}\right) = \int_0^1\sum_{n=0}^{\infty}(1-x)x^{3n}\mathrm{d}x = \int_0^1\frac{1}{1+x+x^2}\mathrm{d}x.$$

By substituting $y = (2x+1)/\sqrt{3}$ it follows that

$$\sum_{n=0}^{\infty}\left(\frac{1}{3n+1} - \frac{1}{3n+2}\right) = \frac{2}{\sqrt{3}}\int_{1/\sqrt{3}}^{\sqrt{3}}\frac{1}{1+y^2}\mathrm{d}y = \frac{2}{\sqrt{3}}\left[\tan^{-1}(\sqrt{3}) - \tan^{-1}(1/\sqrt{3})\right] = \frac{\pi}{3\sqrt{3}}.$$

Thus, as a lower bound for $|\Lambda_{k,1}|$ we obtain

$$|\Lambda_{k,1}| \geq \frac{2\pi}{\sqrt{3}}k^2 - O(k),$$

from which it follows that

$$\lim_{k\to\infty} \frac{\sqrt{|\Lambda_{k,1}|\sqrt{3}}}{\sqrt{2\pi}(k+1)} \geq 1.$$

As a consequence, we see that

$$\lim_{k\to\infty} \frac{\sqrt{|\Lambda_{k,1}|\sqrt{3}}}{\sqrt{2\pi}(k+1)} - \frac{4}{2(1+\varepsilon')\pi(k+1)} \geq 1,$$

since the second term becomes arbitrarily small as k tends to infinity. As $(1+\varepsilon)/(1+\varepsilon')$ is a constant less than 1, for k large enough Equation (2.2.1) cannot hold. For such k we have $G_k \in \varepsilon$-RUDI and $G_k \notin \varepsilon'$-RUDI, as desired. □

2.3 A Gap Theorem for ε-Rudi

Suppose we want to determine whether a specific graph G has an ε-robust representation as a unit disk intersection graph. If in addition we know that G has an r-robust representation with $r > \varepsilon$, then clearly the answer to this question is positive. However, we can say more about such an ε-robust representation. Namely, if we start with an r-robust representation intuitively we can perturb the points of the representation somewhat, while still remaining ε-robust. We can use this perturbation freedom to force an ε-robust representation in which all of the midpoints lie on a predefined grid. This is formalized in the following theorem.

Theorem 2.3.1. *Let G be a graph with an r-robust unit disk representation, then there is an ε-robust unit disk representation on a d-grid, as long as*

$$d \leq \sqrt{1/2} \approx 0.707 \text{ and } \varepsilon + 3\sqrt{2}d \leq r.$$

We call Theorem 2.3.1 a *gap theorem* as the grid resolution d is mainly bounded by the gap $r-\varepsilon$ between the robustness of a starting representation and the required robustness in our final representation. The larger this gap, the coarser the grid of the ε-robust representation can be.

To prove Theorem 2.3.1, we introduce the useful notion of über robustness. The main idea behind the proof of Theorem 2.3.1 is that any graph G is δ-über robust as long as G has some r-robust representation with r sufficiently larger than ε. For fixed ε, we say that a graph G has a *δ-über robust* representation if we can move every point of such a representation of G by at most δ and still retain an ε-robust representation. We also say G is δ-über robust, if such a representation exists. Note that, unless G has no edges, any δ-über robust graph G has $\delta \leq \frac{1}{2}$, as otherwise we can move two connected vertices more than 1 apart. The additional freedom of a δ-über robust graph G allows us to fix the representation to a grid of limited resolution.

Lemma 2.3.2. *If G is δ-über robust and Λ is a d-grid with $d \leq \delta\sqrt{2}$ then there is an ε-robust unit disk representation of G on Λ.*

For clarification, we define the d-grid Λ as the lattice $\{au+bv : a, b \in \mathbb{Z}\}$ spanned by the vectors $u = (d, 0)$ and $v = (0, d)$.

Proof. Let P be an δ-über robust representation of G and consider any point $p \in P$. Breaking ties arbitrarily, let $q \in \Lambda$ be the closest grid point to p. Note that in the Voronoi diagram of Λ, each cell is a square of side length d. It follows that p has distance at most $d/\sqrt{2}$ from q. Now, since $d/\sqrt{2} \leq \delta$ and G is δ-über robust, we can move every point of P to the nearest grid point to obtain the desired representation. □

In order to infer Theorem 2.3.1 from Lemma 2.3.2, it remains to show that any graph G with a sufficiently large gap $r - \varepsilon$ is δ-über robust for some suitable δ. This is precisely what the following lemma shows.

Lemma 2.3.3. *Let G be a graph with an r-robust unit disk representation and suppose $0 < \delta \leq \frac{1}{2}$ is fixed. Then G is δ-über robust as long as $\varepsilon + 6\delta \leq r$.*

Proof. Note that if $r > 1$, then G can only consist of disjoint cliques by Lemma 2.1.1. In this case we can position the cliques arbitrarily far apart and the result follows. Thus, we may assume that $r \leq 1$.
Let P be an r-robust representation of G. Scale P by a factor $(1 - 2\delta) < 1$. Then the distance of any adjacent points becomes at most $1 - 2\delta$. Similarly, as $r \leq 1$, the distance between any non-adjacent points becomes at least $(1 - 2\delta)(1 + r) = 1 + r - 2\delta - 2r\delta \geq 1 + r - 4\delta \geq 1 + \varepsilon + 2\delta$.
This implies that we can move all points by at most δ and still have a valid ε-robust representation. □

Theorem 2.3.1 now follows as an immediate corollary of Lemma 2.3.2 and Lemma 2.3.3, with $\delta = d/\sqrt{2}$.

2.4 Witnesses of Polynomial Size and NP-Membership

In this section we employ the gap theorem derived in the previous section to show that a graph G with a sufficiently large gap $r - \varepsilon$ has a polynomially sized binary witness. This section is also where we tie in to Chapter 1, as the main result we show is the following theorem.

Theorem 2.4.1. *Let ε be badly approximable. Then ε-RUDI admits a binary witness.*

We say that ε-RUDI admits a binary witness if the problem admits a binary witness for every ε-robust graph G. For all other input graphs no witness exists, as these are not ε-robust. Here a *binary witness* for an input graph G is a polynomially sized binary witness, paired with a polynomial verification algorithm on a realRAM machine with access to the additional constant ε. Note that this differs from the definition of an NP-witness, as there the verification algorithm should be a wordRAM algorithm with no access to additional constants.

To see that ε-RUDI admits such a verification algorithm, note that it suffices to compare the midpoint distances to 1 and $1 + \varepsilon$, and check whether these comparisons agree with the constraints imposed by the input graph G. Hence, to show that Theorem 2.4.1 holds, we only need to show that each input graph G has a polynomially sized representation.

To maximize the result of the gap theorem, we would like to choose r as large as possible. This leads us to the definition of the *ridigity value* r of a graph G as the supremum of all ε such that G has an ε-robust representation. In terms of description length, we derive the following result for r.

Lemma 2.4.2. *If r is the rigidity value of a graph G with n vertices, then $r \in \overline{\mathbb{Q}} \cap \mathbb{R}$ and there is an absolute constant $c > 0$ with $L(r) \leq O(n^c)$.*

Proof. Suppose we have any graph G with an ε-robust representation. As mentioned before, for any ε we can construct an ETR formula $\Psi_G(x)$ whose satisfiability corresponds to whether G has an x-robust representation, simply by taking the conjunction of all the distance constraints imposed by G. If we define the quantifier $(\mathrm{H}\delta)$ as $(\exists \delta' > 0)(\forall \delta \in (0, \delta'))$, which can be interpreted as "for all sufficiently small δ", first defined by Bürgisser and Cucker[33], then the formula

$$(\mathrm{H}\delta)(\Psi_G(r - \delta) \wedge \neg\Psi_G(r + \delta))$$

is a defining formula for the rigidity value r of G. This already implies that $r \in \overline{\mathbb{Q}} \cap \mathbb{R}$. Also, as was shown by Bürgisser and Cucker, this formula can be converted to an equivalent ETR formula $\Phi(r)$ in polynomial time[33]. As $\Psi_G(x)$ is of polynomial size in n, then so is $\Phi(r)$. Certainly the minimum defining formula of r will then also have size $O(n^c)$. □

Using Lemma 2.4.2, we can now show Theorem 2.4.1.

Proof of Theorem 2.4.1. As ε is badly approximable there is an absolute constant $C > 0$ with

$$|\varepsilon - \alpha| \geq 2^{-L(\alpha)^C}$$

for all but finitely many $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$. By Corollary 1.5.4, ε is not algebraic and so we can increase C to remove all exceptions. By Lemma 2.4.2 we have $r \in \overline{\mathbb{Q}} \cap \mathbb{R}$ and

$$|\varepsilon - r| \geq 2^{-L(r)^C} \geq 2^{-C'^C n^{cC}},$$

where C' is the implicit constant in $L(r) = O(n^c)$ in Lemma 2.4.2. As G has an ε-robust representation and ε is not algebraic while r is, we see that $r > \varepsilon$. We now make use of the gap theorem. Choosing $d = 2^{-C'^C n^{cC}-3}$, the requirements of Theorem 2.3.1 are satisfied, and so we can find an ε-robust representation of G on a $2^{-C'^C n^{cC}-3}$-grid. While this resolution is exponential in n, note that we can represent an integer of this size in only $O(n^{cC})$ bits, which is polynomial in n.

To ensure that this implies that there is some polynomially sized binary witness, we require a bound on the absolute size of the coordinates, to bound their numerator. One way to see this is by noting that if we sort any representation of G by x-coordinate, we may assume the difference in x-coordinate at each step is at most $1 + \varepsilon$, as otherwise this distance can be contracted. Repeating the argument for the y-coordinate, there exists an ε-robust grid representation of G where each coordinate is absolutely bounded by $(1+\varepsilon)n$. As per instance of ε-RUDI, this ε can be considered as a constant, this implies that we can indeed find a polynomially sized binary witness for every ε-robust graph G. □

Theorem 1.5.8, our measure-theoretic result that all non-badly approximable numbers have measure 0, implies the following corollary of Theorem 2.3.1.

Corollary 2.4.3. *Let $\varepsilon \geq 0$ be chosen uniformly at random. Then ε-*RUDI *admits a binary witness almost surely.*

More interesting is what happens when ε is badly approximable and polynomial-time computable, which is the type of special number that we looked for in Chapter 1. While we have a difference in verification algorithm between our binary witness and an NP-witness, for polynomial-time computable ε, the realRAM verification algorithm for a binary witness can be transformed into a wordRAM verification algorithm without the need for ε as a machine constant. This leads to the following corollary of Theorem 2.4.1.

Corollary 2.4.4. *Let ε be badly approximable and polynomial-time computable. Then ε-*RUDI *is in* NP.

Proof. Theorem 2.4.1 implies that any graph G has an ε-robust representation of polynomial size in the size of G. Thus, to show NP-membership, it suffices to construct a polynomial-time binary verification algorithm. The idea is that this can be inferred from the ETR formula $\Psi_G(\varepsilon)$ whose satisfiability corresponds to the representability of G. The verification algorithm only has to verify whether the given representation corresponds to a satisfying assignment of $\Psi_G(\varepsilon)$. As a representation of the instance is already given, all the required variables for $\Psi_G(\varepsilon)$ can be computed in polynomial time.

The only potential issue is that while ε is polynomial-time computable, we might need to compute more than polynomially many digits of ε to verify $\Psi_G(\varepsilon)$. Luckily, ε is badly approximable. The only difficulty in verifying $\Psi_G(\varepsilon)$, which is quantifier-free, is deciding the comparisons. To each such comparison in terms of $0, 1$ and ε, and the given variable values, we can assign a bound r such that the constraint holds for $\varepsilon < r$ and does not hold for $\varepsilon > r$, or some other comparison between ε and r. For instance, $\varepsilon^2 \geq \varepsilon + 1$ if $\varepsilon \geq \frac{1}{2}(\sqrt{5}+1)$ or $\varepsilon \leq \frac{1}{2}(-\sqrt{5}+1)$. Hence, it suffices to compare ε to these induced bounds r for each comparison.

Each of these r is an algebraic number defined by the specific comparison and the values of the variables. The comparison can be transformed into a polynomially sized defining formula for r in terms of the variable values. As we assume the variables to be of polynomial description size, this implies that each r can be described by an ETR formula of polynomial length. Now, as ε is badly approximable, it lies exponentially far from r so that we only need to calculate polynomially many digits of ε to compare ε with r, as desired. □

It should be clear to the reader why we were so interested in finding a polynomial-time computable, badly approximable number in Chapter 1 as the existence of such a number would lead to the quite counter-intuitive result that for some ε, the problem of ε-RUDI is in NP.

2.5 $\exists\mathbb{R}$-Completeness

We have seen that, depending on ε, the problem ε-RUDI can be of varying complexity; from polynomial-time solvable, to having a binary witness, or even being undecidable. We even conjecture that there are ε for which ε-RUDI is in NP. Are there also ε for which ε-RUDI is precisely as difficult as the non-robust problem of UDI? That is, are there ε for which ε-RUDI is $\exists\mathbb{R}$-complete? Of course, for $\varepsilon = 0$ we already know that this is the case as the resulting problem is equivalent to UDI. What can we say about $\varepsilon > 0$?

In terms of $\exists\mathbb{R}$-membership, we know the following.

Theorem 2.5.1. *Let $\varepsilon \in \overline{\mathbb{Q}} \cap \mathbb{R}_{\geq 0}$ be a real algebraic number. Then ε-*Rudi *is in $\exists\mathbb{R}$.*

Proof. Similarly to Udi, for any graph G we can construct an ETR formula with constants $0, 1$ and ε whose satisfiability is equivalent to G having an ε-robust representation. We encode every midpoint position by two real variables and define an ETR formula as the conjunction of all distance constraints imposed by G. Furthermore, for real algebraic ε, we already saw that we can define ε by an ETR formula with only constants 0 and 1. Combining the ETR formula $\Psi_G(\varepsilon)$ derived from G with a defining formula Φ_ε defining ε through the variable X, we derive an ETR formula with constants 0 and 1 whose satisfiability is equivalent to G having an ε-robust representation, namely $\Psi_G(X) \wedge \Phi_\varepsilon$, where we replace every occurrence of ε in $\Psi_G(\varepsilon)$ by X. It is straightforward to check that this ETR formula is of polynomial size, so that indeed ε-Rudi is in $\exists\mathbb{R}$. □

Hence, for real algebraic ε, it suffices to show $\exists\mathbb{R}$-hardness. For specific ε, this can be done through a very problem-specific reduction from Etr-Inv, a known $\exists\mathbb{R}$-complete problem where every clause is either an addition $x + y = z$, an inversion $xy = 1$, or an initialization $x = 1$ [3]. This reduction requires rigid building blocks to build up addition and inversion gadgets, for the corresponding clauses in a given Etr-Inv instance.

Unfortunately, to reason about these rigid building blocks, we have to adhere to a slightly modified version of ε-Rudi. Instead of requiring two non-neighbouring vertices to have a distance $d > 1 + \varepsilon$, we also allow a distance of $1 + \varepsilon$, so that the distance should be $d \geq 1 + \varepsilon$. We denote this problem by ε-Rudi*. Allowing equality ensures that for any rigidity value r of a graph G, we have an actual r-robust representation for G, which leads to rigid structures that can be used in the reduction from Etr-Inv. It should be noted that $\varepsilon = 0$ no longer leads to a problem equivalent to Udi. For ε-Rudi*, the following is shown in [5].

Theorem 2.5.2. *The problem $\left(\sqrt{1+\frac{1}{4}} - 1\right)$-*Rudi* *is $\exists\mathbb{R}$-complete.*

While this result may be somewhat underwhelming, as it only exposes a single ε for which ε-Rudi* is $\exists\mathbb{R}$-hard, it gives an indication of how involved such $\exists\mathbb{R}$-hardness proofs are. While we will not expose the full proof of Theorem 2.5.2 given in [5], we give an alternative proof of the rigidity of the main building block used in the reduction. We define a *globally rigid* graph as a graph for which there is a unique ε-robust representation, up to rotation and mirroring.

Lemma 2.5.3. *The needle in Figure 2.9 is globally rigid for $\varepsilon = \sqrt{1+\frac{1}{4}} - 1$, with the exception of the two* helper points *to the top and bottom of the needle.*

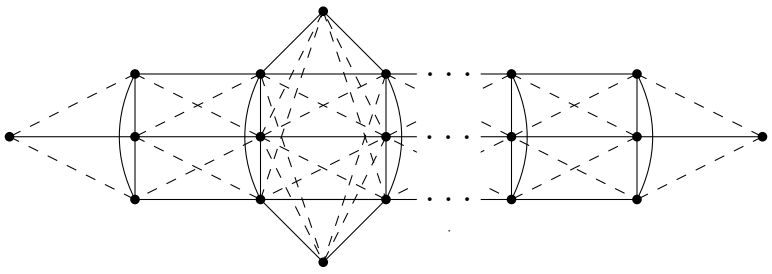


Figure 2.9: A globally rigid needle.

The needle is constructed in such a way that most of the dashed non-edges are forced to have length *exactly* $1 + \varepsilon = \sqrt{1+\frac{1}{4}}$. This in turn leads to rigidity of the entire needle. The two helper points ensure that the *points* of the needle are on the middle row, as otherwise the three rows could still be arbitrarily permuted. The key behind the rigidity of the needle is the following lemma on the rigidity of the matching of two K_3.

Lemma 2.5.4. *Up to a permutation of rows, the representation in Figure 2.10 is globally rigid for $\varepsilon = \sqrt{1+\frac{1}{4}} - 1$.*

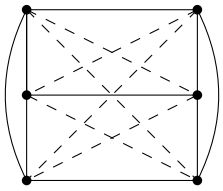


Figure 2.10: A representation for the matching of two K_3.

Lemma 2.5.4 is the main result of this section. Before we can prove it, we show a number of auxiliary lemmas. A useful extension of Lemma 2.2.4 is the following.

Lemma 2.5.5. *Any quadrilateral with sides of length at most* 1 *and diagonals of length at least* $1+\varepsilon$ *is simple and convex.*

Proof. The quadrilateral must be simple as no edges can intersect by Lemma 2.2.4. Suppose it is not convex and label the quadrilateral $ABCD$ so that A lies in $\triangle BCD$, as in Figure 2.11. The result follows from $|AC| \geq 1+\varepsilon > 1 \geq |BC|$ and similarly $|AC| > |DC|$, which is a contradiction with the fact that A lies in $\triangle BCD$. □

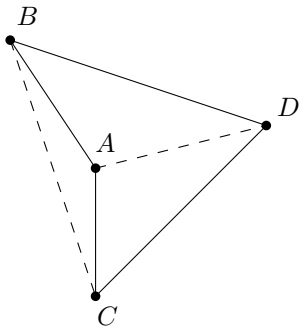


Figure 2.11: A non-convex quadrilateral.

To aid in the proof of Lemma 2.5.4, we show the following helper lemma.

Lemma 2.5.6. *The area of a quadrilateral with sides of length at most* 1 *and diagonals of length at least* $1+\varepsilon = \sqrt{1+\frac{1}{k^2}}$ *is at least* $\frac{1}{k}$*, with equality if and only if the quadrilateral is a rectangle of* 1 *by* $\frac{1}{k}$*.*

Proof. By Lemma 2.5.5, any such quadrilateral is convex. Denote by p, q and $\theta \geq \frac{\pi}{2}$ the lengths of the two diagonals and the angle between them, as depicted in Figure 2.12. Then the area of the quadrilateral is $\frac{1}{2}pq\sin\theta \geq \frac{1}{2}\left(1+\frac{1}{k^2}\right)\sin\theta$. We will bound $\sin\theta$ from below. As $\theta \in [\frac{\pi}{2}, \pi]$, this is equivalent to bounding θ from above.

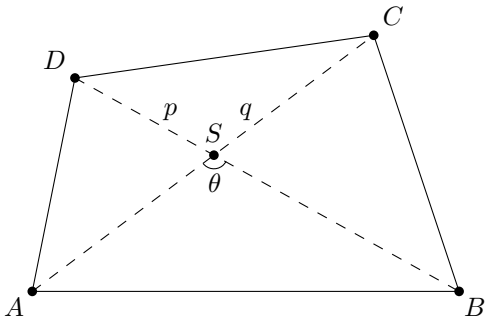


Figure 2.12: An arbitrary quadrilateral.

Denote the intersection of the two diagonals by S. As the sum of the diagonal lengths is at least $2(1+\varepsilon)$, we can label the quadrilateral vertices such that $\triangle ASB$ has $\angle ASB = \theta$, $|AB| \leq 1$ and $|AS| + |SB| \geq 1 + \varepsilon$. Thus, θ can be bounded above by the maximal angle $\angle ASB$ in a triangle with $|AB| \leq 1$ and $|AS| + |SB| \geq 1 + \varepsilon$. Lengthening the segment AB or moving S closer to AB increases $\angle ASB$, so that any optimal value is attained when $|AB| = 1$ and $|AS| + |SB| = 1 + \varepsilon$. This fixes an ellipse on which S must lie. As the set of points with constant $\angle ASB$ is a circular arc, in the optimum this circular arc and the ellipse touch, which implies that $|AS| = |SB| = \frac{1}{2}(1+\varepsilon)$. Applying the law of cosines yields

$$\cos \angle ASB = \frac{\frac{1}{4}(1+\varepsilon)^2 + \frac{1}{4}(1+\varepsilon)^2 - 1}{\frac{1}{2}(1+\varepsilon)^2} = \frac{1-k^2}{1+k^2}.$$

It follows that the area of the quadrilateral is at most

$$\frac{1}{2}\left(1+\frac{1}{k^2}\right)\sin \angle ASB = \frac{1}{2}\left(1+\frac{1}{k^2}\right)\sqrt{1-\cos^2 \angle ASB} = \frac{1}{2}\left(1+\frac{1}{k^2}\right)\frac{2k}{k^2+1} = \frac{1}{k}.$$

Equality is achieved in the case of a 1 by $\frac{1}{k}$ rectangle. To see this is the only case, note that for equality we must have $|AS| + |BS| = 1 + \varepsilon$, so that also $|CS| + |DS| \geq 1 + \varepsilon$. Thus $\triangle CSD$ is also isosceles and it follows that $ABCD$ is a 1 by $\frac{1}{k}$ rectangle. □

We use Lemma 2.5.6 to show the following.

Lemma 2.5.7. *If $ABCD$ is a quadrilateral with sides of length at most 1 and diagonals of length at least $1+\varepsilon = \sqrt{1+\frac{1}{k^2}}$ then $d(C, AB) + d(D, AB) \geq \frac{2}{k}$, with equality if and only if $ABCD$ is a 1 by $\frac{1}{k}$ rectangle with $|AB| = 1$.*

Proof. By Lemma 2.5.5, any such quadrilateral is convex. Assume AB is the x-axis with A left of B and write $y_C = d(C, AB)$ and $y_D = d(D, AB)$. As $ABCD$ is convex, C and D lie on the same side of AB. We assume that $ABCD$ minimizes $y_C + y_D$, and that this minimum is at most $\frac{2}{k}$. If we assume without loss of generality that $y_D \geq y_C$ then $y_C \leq \frac{1}{k}$. As AC is a non-edge, by Pythagoras the difference in x-coordinate of A and C is at least 1. Thus, if $|AB| < 1$, moving B to the right decreases $|BC|$. On the other hand, as $|AD| \leq 1 < |BD|$, D lies on the same side of the perpendicular bisector of AB as A, and so moving B to the right only increases $|BD|$. Thus moving B to the right does not violate any constraints until $|AB| > 1$ so that we may assume that $|AB| = 1$. Note that as long as $|AC| \geq 1 + \varepsilon$, C still lies to the right of B and D, so that moving C to the left only decreases $|BC|$ and $|CD|$. Thus moving C to the left does not violate any constraints until $|AC| < 1 + \varepsilon$ and we may also assume that $|BC| = 1 + \varepsilon$. In this case, y_D is minimal if $|BD| = 1 + \varepsilon$ and $|CD| = 1$. As $|AC| = |BD| = 1 + \varepsilon$ and $|AB| = |CD| = 1$, the triangles $\triangle ABD$ and $\triangle DCA$ are congruent so that the area of $ABCD$ is $\frac{1}{2}(y_C + y_D)$. By Lemma 2.5.6 we now have $\frac{1}{2}(y_C + y_D) \geq \frac{1}{k}$ with equality if and only if $ABCD$ is a 1 by $\frac{1}{k}$ rectangle, from which the result follows. □

We are now able to show Lemma 2.5.4.

Proof of Lemma 2.5.4. Suppose a representation of the graph is $v_1, v_2, v_3, w_1, w_2, w_3$, with edges v_iv_j, w_iw_j and v_iw_i for all $i, j \in \{1, 2, 3\}$. By Lemma 2.5.5, any two segments v_iw_i and v_jw_j are comparable. In particular, we can order the segments v_1w_1, v_2w_2 and v_3w_3. Without loss of generality, assume v_2w_2 is second in the ordering, and that v_1w_1 lies above v_2w_2 and v_3w_3 lies below. Then by Lemma 2.5.7 for $k = 2$, we have

$$d(v_1, v_3) + d(w_1, w_3) \geq d(v_1, v_2w_2) + d(v_3, v_2w_2) + d(w_1, v_2w_2) + d(w_3, v_2w_2) \geq 2.$$

On the other hand, as v_1v_3 and w_1w_3 are edges, we have $2 \geq d(v_1, v_3) + d(w_1, w_3)$, implying that equality must hold and $v_2w_2w_1v_1$ and $v_2w_2w_3v_3$ must both be 1 by $\frac{1}{2}$ rectangles. The result is immediate. □

With Lemma 2.5.4, we are now equipped to prove Lemma 2.5.3.

Proof of Lemma 2.5.3. By Lemma 2.5.4 each subgraph as in Figure 2.10 of the needle is globally rigid, up to permutation of rows. As these are all connected by more than 1 point, the entire main body of the needle, without the two endpoints and the two helper points, is rigid up to permutation of rows. Furthermore, the permutation per subgraph must be the same. Next note that while we can attach the helper vertices to the outer two rows, Figure 2.13 shows that there is no possible location to attach a helper point to the middle row.

Any point with distance at most 1 to the two center vertices, indicated by the shaded region, has distance less than $1+\varepsilon$ from at least one of the other vertices. Hence, the two helper vertices force the row without a helper vertex to be the middle row. In particular, the two endpoint vertices are also connected to the middle row. Figure 2.14 shows that there are only two possible locations for the endpoints, but one of these locations is taken by another vertex of the needle, with which the endpoint does not have an edge. Hence, the position of the endpoints are also fixed. □

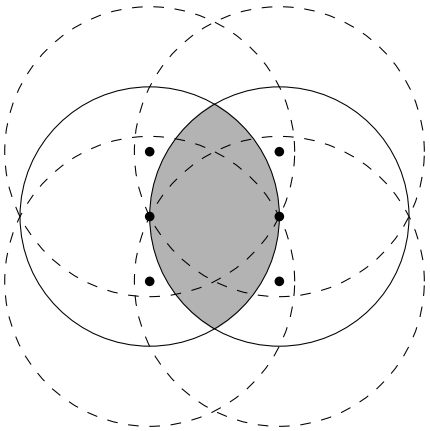


Figure 2.13: Connecting helper points to the middle.

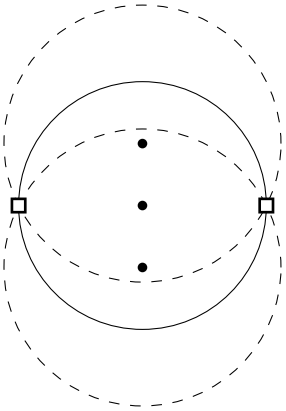


Figure 2.14: Locations for the endpoints.

The proof of Theorem 2.5.2 in [5] goes on to construct addition and inversion gadgets from the needle in Figure 2.9 to complete the reduction. While the fact that Theorem 2.5.2 only shows $\exists\mathbb{R}$-completeness for a single ε is relatively underwhelming, there is hope for a more general result. The fact that both Lemma 2.5.6 and Lemma 2.5.7 have a parameter k, which is set to $k = 2$ in the proof of Lemma 2.5.4, hint to this desired generalization.

Namely, if we can show an analog to Lemma 2.5.4 stating that the matching of two K_{k+1} is globally rigid for $\varepsilon = \sqrt{1 + \frac{1}{k^2}} - 1$, for even k the needle in Figure 2.15 must be globally rigid for this ε, again except for the helper points. This would imply that ε-RUDI* is in $\exists\mathbb{R}$ for all these choices of ε.

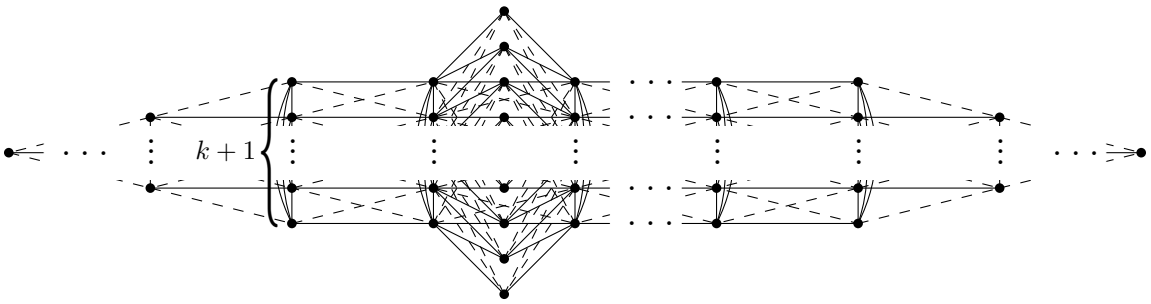


Figure 2.15: A more general needle.

Unfortunately, the proof of Lemma 2.5.4 seems difficult to generalize to all k. In particular, an important part of the proof of Lemma 2.5.4 is that there is a middle edge, with only a single edge above and below. In this case, we can calculate the distance between the top and bottom edge by adding their distance to the middle edge. When $k > 2$ there are more than 3 edges and this additive logic fails, as Lemma 2.5.7 can only really be used to bound the distance between adjacent edges, and the edges could perhaps not be parallel. Hence, we expect ε-RUDI* to be in $\exists\mathbb{R}$ for all $\varepsilon = \sqrt{1 + \frac{1}{k^2}} - 1$ with k a positive even integer, but it seems that we require a strategy that is even more involved than the proof for $k = 2$.

Chapter 3

A General Framework

In the previous chapter we derived results on the complexity of ε-Rudi, depending on what ε is chosen. Namely, we saw that the problem could become trivial, remain $\exists\mathbb{R}$-complete, admit a binary witness, or even become undecidable. Furthermore, the conjectured existence of a badly approximable and polynomial-time computable ε^* would imply ε^*-Rudi lies in NP. In this chapter, we focus on generalizing these results to *any* ε-robust problem.

From the proofs of these results on complexity for ε-Rudi, it is apparent that some results will generalize easier than others. In particular, we will see that while showing $\exists\mathbb{R}$-hardness is highly problem specific, results on the existence of a binary witness, conjectured NP-membership, the simple range, and undecidability all carry over to varying degrees in the general setting. This chapter serves to describe in detail how these results carry over by describing a general framework for ε-robust problems, followed by applying this framework to the problems of Packing and Art Gallery Problem.

3.1 General Properties of an ε-Robust Problem

In this section, we describe the framework that was derived for general ε-robust problems, generalizing the results we saw for ε-Rudi. As mentioned, some results generalize more easily than others, which can be very problem-specific. However, before generalizing the results on ε-Rudi to general ε-robust problems, we first have to describe what we mean by general ε-robustness.

3.1.1 A general definition of robustness

Let Problem be an arbitrary algorithmic decision problem in $\exists\mathbb{R}$, and let ε-Robust Problem denote some robust version of this problem. Let L be the set of positive instances of Problem, so that Problem is equivalent to recognizing L. Similarly, let $L(\varepsilon)$ be the set of positive instances of ε-Robust Problem for any $\varepsilon \geq 0$. Then ε-Robust Problem fits our framework if the following requirements hold.

- $L(0) = L$. That is, the version with zero robustness is precisely the original Problem.
- $\varepsilon < \varepsilon' \iff L(\varepsilon) \supseteq L(\varepsilon')$. In other words, any instance that is ε'-robust is also ε-robust for $\varepsilon < \varepsilon'$.
- For any $\varepsilon \geq 0$, ε-Robust Problem has a *real witness*. That is, a polynomial size real witness together with a polynomial time realRAM verification algorithm, which additionally has access to the constant ε.

Note that none of these requirements are very restrictive, in the sense that most robust versions of problems in $\exists\mathbb{R}$ will fit in the framework. The first requirement simply enforces that ε-Robust Problem is in fact a robust version of Problem, and not some other problem, as the version of ε-Robust Problem in which we require no additional robustness should correspond precisely to the original Problem.

The second requirement is also very intuitive for any definition of robustness, as if there is some ε'-robust representation of an instance, one would assume from a reasonable definition of robustness that this exact representation is also ε-robust for $\varepsilon < \varepsilon'$. This notion of a reasonable definition is precisely what the second requirement formalizes. This requirement also allows us to define the *rigidity* of an instance I as the supremum of all ε such that $I \in L(\varepsilon)$, analogous to what we did for ε-Rudi.

Finally, the third requirement is also not as restrictive as it may seem. As Problem is assumed to be an element of $\exists\mathbb{R}$, such a real witness already exists for Problem. Most of the time, this witness is readily modified to suit ε-Robust Problem by including ε in the verification algorithm.

Comparing the notion of a real witness with that of the binary witness, defined earlier, the only difference is that for a binary witness the witness should be encoded in a polynomial set of bits, whereas for a real witness the witness should be encoded in a polynomial set of reals. Besides a different input encoding, there does not need to be any algorithmic difference between the verification algorithm for a binary witness and a real witness.

Having formalized which ε-Robust Problem we consider in our framework, we can generalize the results that were obtained on the complexity of ε-Rudi.

3.1.2 Simple range or equivalence under rational scaling

We first study for which ε the problem ε-Robust Problem is polynomial time solvable. In general, we expect one of two distinct properties. On one hand, it could happen that for large enough ε the set $L(\varepsilon)$ becomes very simple and thus can be recognized in polynomial time, as is the case with ε-Rudi. On the other hand, it could also be that the notion of robustness allows for rational scaling of the input such that for any positive rational q the problems ε-RP and $q\varepsilon$-RP are computationally equivalent under polynomial time reductions. In this case there is no simple range, unless we are in the very specific case where ε-Robust Problem is polynomial time solvable for all $\varepsilon > 0$.

In case $L(\varepsilon)$ becomes simple for sufficiently ε, this threshold is problem specific and we are not aware of any general statements. It should be noted that not all ε for which ε-Robust Problem is polynomial time solvable have to fall in this simple range. For a quite artificial example, we can define a robust problem which recognizes different classes of graphs, with smaller classes recognized by picking a larger robustness ε. In particular, we look at the following five classes.

- L_0 contains precisely all graphs.
- L_1 contains precisely all string graphs.
- L_2 contains precisely all unit disk graphs.
- L_3 contains precisely all unit interval graphs.
- L_4 is empty.

It is known that $L_0 \supsetneq L_1 \supsetneq L_2 \supsetneq L_3 \supsetneq L_4$. Hence, under some abstract definition of ε-Robust Problem, these sets can be seen as $L(0), L(1), L(2), L(3)$ and $L(4)$ respectively. However, if we look at the algorithmic difficulty to recognize these sets, we get in this order: constant time, NP-complete, $\exists\mathbb{R}$-complete, polynomial time, constant time.

On the other hand, it could be that there is no simple range for ε-Robust Problem at all. As mentioned above, some notions of robustness allow us to relatively decrease the influence of ε by scaling the entire configuration up. An example of this, discussed in detail in Section 3.2, is a robust version of Packing, where a set of polygons is to be packed in a container, while keeping pairwise distance at least 2ε. By scaling up all the polygonal pieces and the container with a rational factor q, we achieve a situation equivalent to packing the original pieces with robustness $q^{-1}\varepsilon$. Thus, if the definition of robustness allows for such scaling invariance, for any positive rational q the problems ε-RP and $q\varepsilon$-RP are computationally equivalent under polynomial time reductions. In this case, there clearly cannot be a simple range unless ε-Robust Problem is polynomial time solvable for all $\varepsilon > 0$.

3.1.3 Undecidability

We now turn to the results on undecidability. For ε-Rudi, undecidability followed from the fact that we defined an uncountable number of distinct algorithmic problems. Similarly, if ε-Robust Problem defines an uncountable number of distinct problems, a similar result follows. We say ε-Robust Problem *defines an uncountable number of distinct problems* in the range $[a, b]$, if for any two $a \leq \varepsilon < \varepsilon' \leq b$, we have that $L(\varepsilon) \supsetneq L(\varepsilon')$. In other words, there is some instance that does have an ε-robust representation, but not an ε'-robust one. It is relatively elementary to see that this condition is equivalent to the rigidity values of all instances lying dense.

Lemma 3.1.1. *ε-Robust Problem defines an uncountable number of distinct problems in the range $[a, b]$, if and only if the rigidity values of all instances of ε-Robust Problem lie dense in $[a, b]$.*

Proof. If the rigidity values of all instances lie dense in $[a, b]$, there must be an instance I with rigidity value r with $\varepsilon < r < \varepsilon'$ for any two $a \leq \varepsilon < \varepsilon' \leq b$. But then $I \in L(\varepsilon)$ and $I \notin L(\varepsilon')$ so that $L(\varepsilon) \supsetneq L(\varepsilon')$. Hence ε-Robust Problem defines an uncountable number of distinct problems in the range $[a, b]$.

On the other hand, assume that ε-Robust Problem defines an uncountable number of distinct problems in the range $[a, b]$. We have to show that there is a rigidity value between any two $a \leq \varepsilon < \varepsilon' \leq b$. As $L(\varepsilon) \supsetneq L(\varepsilon')$, there is some instance I with $I \in L(\varepsilon)$ and $I \notin L(\varepsilon')$. But then by definition the rigidity value of I must lie between ε and ε'. □

An immediate consequence of Lemma 3.1.1, together with the fact that there are only countably many algorithms, is the following general result on undecidability for ε-Robust Problem.

Theorem 3.1.2. *If the rigidity values of* ε-Robust Problem *lie dense in the range* $[a, b]$ *then* ε-Robust Problem *is decidable for only countably many* $\varepsilon \in [a, b]$.

We leave the proof to the reader.

Thus, to derive results on undecidability for ε-Robust Problem, it is equivalent to study the structure of the rigidity values of ε-Robust Problem. While the rigidity values of ε-Robust Problem are strongly problem-dependent, it is sometimes relatively easy to construct a subset of these rigidity values that is already dense by considering relatively simple instances. Examples of such constructions will be given by the framework applications later in this chapter, such as instances of ε-Robust Packing where the only polygon to pack is a single square.

3.1.4 The gap property and NP-membership

For ε-Rudi, we saw several results on the existence of a binary witness and conjectured NP-membership. To reason about NP-membership for ε-Robust Problem, we generalize the gap theorem that led to these results for ε-Rudi.

Definition 3.1.3. We say that ε-Robust Problem has the *gap property*, if the following holds for all $r > \varepsilon$. Pick any $I \in L(r)$ and suppose it has size n. Then there exists a binary witness for I of size polynomial in $n + \log 1/\delta$, where $\delta = r - \varepsilon$. The specific polynomial is allowed to depend on ε.

Recall that a binary witness is a polynomially sized binary witness together with a realRAM verification algorithm. As our general definition of robustness requires the existence of a realRAM verification algorithm for a polynomially sized *real* witness, the main requirement of the gap property is that a representation exists that only requires a polynomial number of bits to encode. The gap property is a *gap* property in the sense that if the gap $\delta = r - \varepsilon$ is big, relatively few bits are required to represent a witness.

Note that Theorem 2.3.1 shows that ε-Rudi has the the gap property. To see this note that we could bound the coordinates of every midpoint by $(1 + \varepsilon)n$. If $2^{-k} \leq \delta < 2^{-k+1}$ for some integer k then Theorem 2.3.1 implies that we can position every midpoint on a grid of size 2^{-k-3}, thus requiring only $O((1 + \varepsilon)nk)$ bits to encode each midpoint, and $O((1 + \varepsilon)n^2k)$ bits for the entire witness. As $1 + \varepsilon$ is constant for a specific instance of ε-Rudi, this is indeed polynomial in $n + k$.

To accommodate a broader range of robust problems, the definition of the gap property is more abstract than Theorem 2.3.1 for ε-Rudi. In particular, we no longer use a grid, as it might not make sense to define such a grid in the context of ε-Robust Problem. Instead, we abstract to the notion of a binary witness, as this is all that is required for results on NP-membership. For ε-Rudi, the grid was simply used as an intermediate step to show the existence of such a binary witness.

NP-Membership. When studying ε-Rudi we discovered that we need ε to have two specific properties for ε-Rudi to be in NP. Analogous results hold for any ε-Robust Problem that has the gap property. The following theorem is the driving force behind these results, which is analogous to Theorem 2.4.1 for ε-Rudi.

Theorem 3.1.4. *Assume that* ε-Robust Problem *has the gap property and let* ε *be badly approximable. Then* ε-Robust Problem *admits a binary witness.*

Proof. This proof is almost completely analogous to that of Theorem 2.4.1 for ε-Rudi. Let $I \in L(\varepsilon)$ be any ε-robust instance and suppose that I has size n. By the third property of robustness, there is an ETR formula $\Psi_I(x)$ of size polynomial in n that is satisfiable if and only if $I \in L(x)$.

Recalling the definition of the quantifier $(\mathrm{H}\delta)$ as $(\exists\delta' > 0)(\forall\delta \in (0, \delta'))$, which can be interpreted as "for all sufficiently small δ", then the formula

$$(\mathrm{H}\delta)(\Psi_I(r-\delta) \wedge \neg\Psi_I(r+\delta))$$

is again a defining formula for the rigidity value r of I. Like for the rigidity values of ε-Rudi, this formula can be converted to an equivalent ETR formula $\Phi(r)$ in polynomial time[33]. As $\Psi_I(x)$ is of polynomial size in n, then so is $\Phi(r)$. In particular there is some absolute constant $c > 0$ with $|\Phi(r)| \leq n^c$ and it follows that the minimal description length $L(r)$ also satisfies $L(r) \leq n^c$.

Now, as ε is badly approximable there is another absolute constant $C > 0$ with

$$|\varepsilon - \alpha| \geq 2^{-L(\alpha)^C}$$

for all but finitely many $\alpha \in \overline{\mathbb{Q}} \cap \mathbb{R}$. By Corollary 1.5.4, ε is not algebraic and so we can increase C to remove all exceptions. As the rigidity r of I is defined by $\Phi(r)$, in particular $r \in \overline{\mathbb{Q}} \cap \mathbb{R}$ and we have

$$|\varepsilon - r| \geq 2^{-L(r)^C} \geq 2^{-n^{cC}}.$$

Now, as ε-Robust Problem has the gap property, there exists a binary witness for I of size polynomial in $n + n^{cC}$. Hence, there exists a binary witness for I of size polynomial in n, with this polynomial only depending on ε. As this holds for every satisfying instance I, indeed ε-Robust Problem admits a binary witness. □

As the set of non-badly approximable numbers has measure 0 by Theorem 1.5.8, we have the following corollary.

Corollary 3.1.5. *Assume that* ε-Robust Problem *has the gap property and let* $\varepsilon \geq 0$ *be chosen uniformly at random. Then* ε-Robust Problem *admits a binary witness almost surely.*

As with ε-Rudi, we can also show the following result on NP-membership of ε-Robust Problem.

Corollary 3.1.6. *Assume that* ε-Robust Problem *has the gap property and let* ε *be badly approximable and polynomial-time computable. Then* ε-Robust Problem *is in* NP.

Like with ε-Rudi, the idea is to translate the equivalent ETR formula $\Psi_I(\varepsilon)$ to a verification algorithm. Again, the only potential issue is that we might need to compute superpolynomially many digits of ε, but completely analogous to the proof of Corollary 2.4.4 this is never necessary as ε is badly approximable.

3.1.5 $\exists\mathbb{R}$-completeness

From the results on $\exists\mathbb{R}$-completeness for ε-Rudi, and also the results that we will derive for ε-Robust Packing, showing $\exists\mathbb{R}$-hardness for ε-Robust Problem seems very problem-specific. However, we do expect that robust versions of many $\exists\mathbb{R}$-complete problems can be shown to be $\exists\mathbb{R}$-hard as well for at least some $\varepsilon > 0$, albeit through very problem-specific methods. Specifically, we expect that a high density of rigidity values is a good indication that an $\exists\mathbb{R}$-completeness proof might be feasible. Based on the results for ε-Rudi, but especially those for ε-Robust Packing that we will see in the following section, we do conjecture the following, which would be a very generic result.

Conjecture 3.1.7. *Let* $r \geq 0$ *be the rigidity value of some instance* I *of* ε-Robust Problem *such that* $I \in L(r)$ *and suppose that* Problem *is* $\exists\mathbb{R}$*-complete. Then* r-Robust Problem *is* $\exists\mathbb{R}$*-complete as well.*

This conjecture is more "in spirit" than an actual conjecture, as there are relatively trivial counterexamples. For instance, for ε-Rudi the rigidity value of a path graph is $r = 1$, but this lies in the simple range. Still, it seems useful to remark this observation, as it could serve as a meaningful generalization of the results on $\exists\mathbb{R}$-membership.

It should be noted that with the current methods, we have little idea how to prove or disprove the above conjecture, especially considering the fact that the $\exists\mathbb{R}$-hardness proofs for ε-Rudi and ε-Robust Packing are quite involved and very problem-specific. The idea behind the conjecture is that one could perhaps make use of the rigid structure of an r-robust representation of I to construct general gadgets.

While Conjecture 3.1.7 would be very influential for the study of $\exists\mathbb{R}$-hardness for robust problems, we do note that this would likely not make $\exists\mathbb{R}$-hardness proofs less problem-specific, due to the fact that the structure of the rigidity values itself is very problem-dependent.

On the $\exists\mathbb{R}$-membership side, it is not so hard to show the following theorem.

Theorem 3.1.8. *Let $\varepsilon \in \overline{\mathbb{Q}} \cap \mathbb{R}_{\geq 0}$ be a real algebraic number. Then ε-*Robust Problem *is in $\exists\mathbb{R}$.*

Proof. For algebraic ε, there is a defining formula Φ_ε defining ε through X. The third property of robustness implies that there is a polynomially sized ETR formula $\Psi_I(\varepsilon)$ with constants $0, 1$ and ε whose satisfiability is equivalent to whether the instance I has an ε-robust representation. Similar to the proof of Theorem 2.5.1 for ε-Rudi, the formula $\Psi_I(X) \wedge \Phi_\varepsilon$ no longer requires the explicit constant ε, and its satisfiability is equivalent to whether the instance I has an ε-robust representation. This proves that ε-Robust Problem is in $\exists\mathbb{R}$. □

Because the proofs of $\exists\mathbb{R}$-hardness are so problem-specific and highly involved, in the example applications that follow we only give a proof of $\exists\mathbb{R}$-hardness for ε-Robust Packing.

3.2 Framework Application: ε-Robust Packing

To illustrate how our framework could be applied to other $\exists\mathbb{R}$-complete problems, we study a robust version of Packing, which asks whether a set of simple polygons with rational vertices fit inside a square of given rational size without overlap. The pieces can be translated and rotated arbitrarily. This problem was shown to be $\exists\mathbb{R}$-complete, even if only convex pieces are allowed, by Abrahamsen, Miltzow and Seiferth[4], who studied a wider range of packing problems. Our instance of Packing is denoted Pack($\llcorner \to \square, \circlearrowleft +$) by Abrahamsen, Miltzow and Seiferth. The ε-robust version of this problem, denoted by ε-Robust Packing requires that a configuration exists where the polygons can be arbitrarily translated by at most a distance ε, still without overlap. This is equivalent to fitting slightly larger polygons with rounded corners of radius ε into the outer square. See Figure 3.1 for an example of such a pseudo-polygon, which we call the *ε-expansion* of the original polygon. No explicit restrictions are made on rotation, but in Section 3.2.3 we show that the translational robustness implies some form of rotational robustness as well. We are again interested in how the computational complexity of ε-Robust Packing depends on ε.

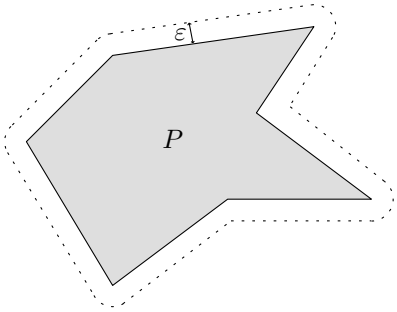


Figure 3.1: A polygon P with corresponding ε-expansion.

3.2.1 Equivalence under rational scaling

As the size of the polygonal pieces and the square container are independent of ε, if we scale up all of the pieces and the container by some constant $q \in \mathbb{Q}_{>0}$, the pieces are still only required to have a translational freedom of ε. This is equivalent to packing the original polygonal pieces with a translational freedom of $q^{-1}\varepsilon$. This implies the following theorem.

Theorem 3.2.1. *Let $\varepsilon \in \mathbb{R}_{\geq 0}$. For any $q \in \mathbb{Q}_{>0}$, the problems ε-*Robust Packing *and $q\varepsilon$-*Robust Packing *are computationally equivalent under polynomial time reductions.*

In particular, results on undecidability, NP-membership or $\exists\mathbb{R}$-completeness transfer from any specific ε to all rational multiples of ε as well. As we will see examples of nontrivial $\varepsilon > 0$, this implies ε-Robust Packing does not really have a simple range, as was the case for ε-Rudi.

3.2.2 Undecidability

As with ε-RUDI, the uncountably many choices for ε combined with the countable number of possible algorithms leads to the undecidability of many instances of ε-Robust Packing. In fact, due to the relative scaling invariance of ε-Robust Packing as shown in 3.2.1, we can derive a stronger result than for ε-Rudi.

Theorem 3.2.2. *The problem of ε-Robust Packing is decidable for only countably many $\varepsilon \in \mathbb{R}_{\geq 0}$.*

From Theorem 3.1.2 of the general framework, it suffices to show that the set of rigidity values of ε-Robust Packing lies dense in $\mathbb{R}_{\geq 0}$. Unlike ε-Rudi, showing this for ε-Robust Packing is straightforward.

Lemma 3.2.3. *The rigidity values of ε-Robust Packing lie dense in $\mathbb{R}_{\geq 0}$.*

Proof. We want to show that for any $0 \leq \varepsilon < \varepsilon'$ there is some instance I with rigidity value between ε and ε'. That is, we want $I \in \varepsilon$-Robust Packing and $I \notin \varepsilon'$-Robust Packing. Let $r \in \mathbb{Q}$ be a rational between ε and ε', so that $\varepsilon < r < \varepsilon'$. Let $t \in \mathbb{Q}_{>0}$ and choose a square of side length t as the only piece, and a square container of side length $t + 2r$. Clearly, $I \in \varepsilon$-Robust Packing, as we can place the $t \times t$ square exactly in the middle of the square container, as depicted in Figure 3.2. On the other hand, if $I \in \varepsilon'$-Robust Packing, then at the very least the area of the ε'-expansion of the $t \times t$ square has to be at most the area of the square container. In other words, we must have

$$t^2 + 4t\varepsilon' + \pi\varepsilon'^2 \leq (t + 2r)^2 = t^2 + 4tr + 4r^2.$$

As $r < \varepsilon'$, this cannot hold if $t \in \mathbb{Q}_{>0}$ is chosen large enough. □

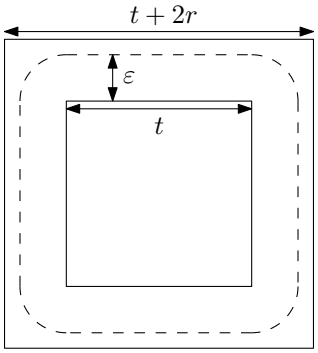


Figure 3.2: Packing a single square piece.

3.2.3 The gap property and NP-membership

As with ε-RUDI, to show the existence of a polynomially sized binary witness, we show that when the rigidity value of an input and ε differ by a significant amount, we can perturb the polygonal pieces in an r-robust configuration so that the resulting configuration has relatively simple rational vertex coordinates while still being ε-robust.

Theorem 3.2.4. *ε-Robust Packing has the gap property.*

Theorem 3.2.4 follows immediately from the following lemma.

Lemma 3.2.5. *Suppose I is an r-robust input of n bits with $r > \varepsilon$. Define $\delta = r - \varepsilon$ and pick $m \in \mathbb{Z}_{\geq 0}$ with $2^{-m} \leq \delta$. Then there exists an ε-robust configuration in which the vertex coordinates of the pieces relative to the square container are rational with numerator and denominator absolutely bounded by $2^{3m+5n+7}$.*

As by definition ε-Robust Packing requires translational freedom in the polygonal pieces, it is straightforward to see that when δ is large, we can translate the pieces of an r-robust configuration relatively far while still remaining ε-robust. However, when packing the polygonal pieces into the square container, pieces may also be rotated. To ensure all vertex coordinates are still relatively simple rationals after this rotation, we utilize Pythagorean triples. The following Lemma implies that we can find a unit vector consisting of relatively simple rationals that has an angle very similar to a target angle θ.

Lemma 3.2.6. *Fix $N \geq 1$. For any $\theta \in (-\pi, \pi]$, there is a rational unit vector $\left(\frac{a}{c}, \frac{b}{c}\right)$ with $0 < c < 8N^2$ such that $\left|\text{atan2}\left(\frac{b}{c}, \frac{a}{c}\right) - \theta\right| \leq \frac{1}{N}$. That is, the angle of the unit vector $\left(\frac{a}{c}, \frac{b}{c}\right)$ and θ differ by at most $1/N$.*

Proof. By flipping signs, it suffices to show the result for $\theta \in [0, \frac{\pi}{2}]$ and $a, b \geq 0$. Using the standard parametrization for Pythagorean triples, we write $a = m^2 - n^2$, $b = 2mn$ and $c = m^2 + n^2$ for some $n < m \leq 2N$. The latter inequality guarantees that $c < 8N^2$. If we write $\sigma = \text{atan2}\left(\frac{b}{c}, \frac{a}{c}\right) = \tan^{-1}\left(\frac{b}{a}\right) = \tan^{-1}\left(\frac{2mn}{m^2-n^2}\right)$, then we derive that $\tan\frac{\sigma}{2} = \frac{n}{m}$. Further, as $\theta \in [0, \frac{\pi}{2}]$ we have $0 \leq \tan\frac{\theta}{2} \leq 1$. Hence, picking $m = 2N$ we can find a $0 \leq n < m$ with $\left|\tan\frac{\sigma}{2} - \tan\frac{\theta}{2}\right| = \left|\frac{n}{m} - \tan\frac{\theta}{2}\right| \leq \frac{1}{2N}$. As the derivative of the arctangent is always in $(0, 1]$, taking the arctangent implies that $\left|\frac{\sigma}{2} - \frac{\theta}{2}\right| \leq \frac{1}{2N}$, from which the result follows. □

If we rotate a point (x, y) by a given unit vector (a, b), the resulting point is $(ax - by, bx + ay)$. Hence, if the unit vector (a, b) consists of simple rationals, the resulting point does not require substantially more bits to encode than the original point. Hence, Lemma 3.2.6 implies that we can rotate a polygon by an angle very close to a target angle θ, without affecting the size of the binary encoding too much. With this, we are now ready to show Lemma 3.2.5, which immediately implies that ε-Robust Packing has the gap property.

Proof of Lemma 3.2.5. As we have an r-robust input, there is some configuration of the pieces where all of the pieces have a pairwise distance of at least $2r$, and the pieces have a distance of at least r to the square container. By translating each piece by at most $\delta = r - \varepsilon$, the resulting configuration is still ε-robust. To see that we have some rotational freedom as well, we note that while translating a piece translates every point inside the piece by the same vector, we can also translate the points inside the piece by different vectors, as long as none of these exceed δ in length.

Rotating a polygon by an angle of $\Delta\theta$ around a point in the interior translates points at distance d by a vector of length $2d\sin(\Delta\theta/2)$, as seen in Figure 3.3. We can bound this length from above by $d\Delta\theta$, and as the input has size n, we can bound $d \leq 2^n$, so that rotating any polygonal piece around some interior point by an angle $\Delta\theta$ translates any point of the polygonal piece by at most $2^n\Delta\theta$. Thus, as long as we translate all pieces of the r-robust configuration by at most $\delta/2$ and rotate them by at most $\delta/2^{n+1}$, the total translation of any interior point does not exceed δ, so that the resulting configuration is still ε-robust.

View any polygonal piece in the r-robust configuration and suppose that the angle with which it is ro-

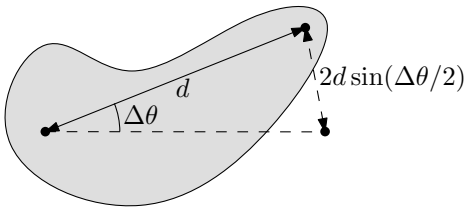


Figure 3.3: Rotating a polygon around a point in its interior.

tated with respect to the orientation in the input is θ. Recall that $m \in \mathbb{Z}_{>0}$ is chosen such that $2^{-m} \leq \delta$. By Lemma 3.2.6 with $N = 2^{m+n+1}$ we can find a rational unit vector $\left(\frac{a}{c}, \frac{b}{c}\right)$ with $|a|, |b|, c < 2^{2m+2n+5}$ such that $\left|\text{atan2}\left(\frac{b}{c}, \frac{a}{c}\right) - \theta\right| \leq 2^{-m-n-1} \leq \delta/2^{n+1}$. We will slightly rotate the polygonal piece so that the angle matches $\left(\frac{a}{c}, \frac{b}{c}\right)$, which is a rotation of at most $\delta/2^{n+1}$, as desired. To see how this affects the vertex coordinates of the polygon, pick one of the vertices as a reference. As the input is of size n, the vector between any vertex and the reference vertex is $\left(\frac{p}{q}, \frac{s}{t}\right)$ with $|p|, q, |s|, t < 2^n$. After rotating to $\left(\frac{a}{c}, \frac{b}{c}\right)$, the vector becomes $\left(\frac{apt-bqs}{cqt}, \frac{aqs+bpt}{cqt}\right)$, say $\left(\frac{x}{y}, \frac{z}{w}\right)$ with $|x|, y, |z|, w < 2^{2m+4n+6}$.

To ensure that the coordinates of all vertices are relatively simple rationals, it remains to translate the polygonal piece so that the reference point has both denominators equal to 2^{m+1}, which we can do by translating by a distance of at most $2^{-m-1} = \delta/2$. The numerators will be at most 2^{m+n+1}, as the square container has side length at most 2^n as it is also contained in the input. As a result, all polygonal vertices have a numerator and denominator of absolute value at most $2^{3m+5n+7}$. As we have rotated the pieces by at most $\delta/2^{n+1}$ and translated them by at most $\delta/2$, the resulting configuration is still ε-robust. □

As ε-Robust Packing has the gap property, from our framework we immediately obtain the following two corollaries, as with ε-Rudi.

Corollary 3.2.7. *If $\varepsilon \geq 0$ is selected uniformly at random, then ε-Robust Packing has a binary witness almost surely.*

Corollary 3.2.8. *If ε is badly approximable and polynomial-time computable, then ε-Robust Packing is in* NP.

3.2.4 $\exists\mathbb{R}$-completeness

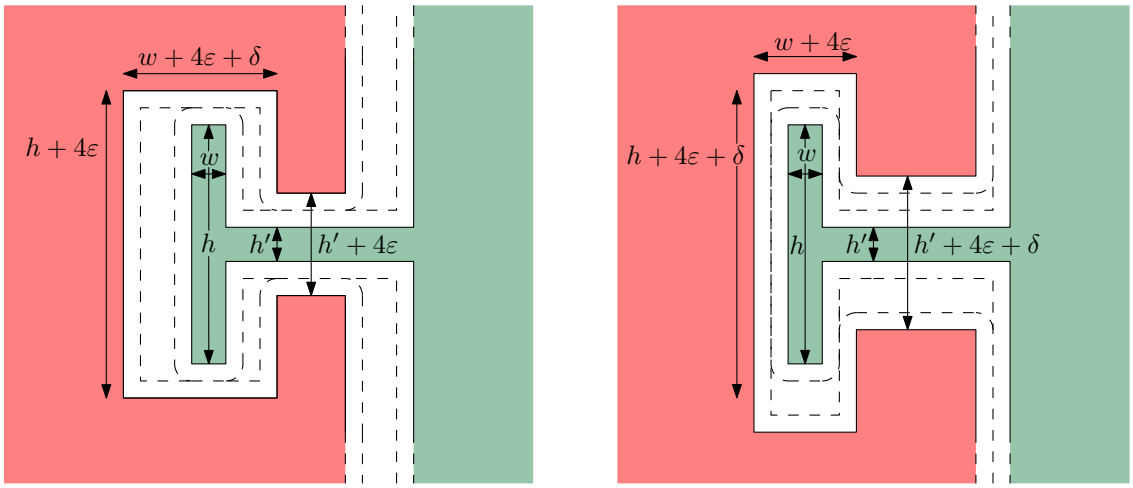
As for ε-Rudi, we show that ε-Robust Packing remains $\exists\mathbb{R}$-complete for some values of $\varepsilon > 0$. In fact, contrary to the perhaps somewhat underwhelming result for ε-Rudi, we show that ε-Robust Packing remains $\exists\mathbb{R}$-complete for all rational ε.

Theorem 3.2.9. *If ε is a real algebraic number then ε-Robust Packing is in $\exists\mathbb{R}$. If $\varepsilon \in \mathbb{Q}_{\geq 0}$, then ε-Robust Packing is $\exists\mathbb{R}$-complete.*

For algebraic ε, and thus rational ε in particular, by Theorem 3.1.8 we know that ε-Robust Packing is in $\exists\mathbb{R}$. It remains to show that ε-Robust Packing is $\exists\mathbb{R}$-hard for rational ε. We present a polynomial reduction from Range-Etr-Inv, a problem introduced and shown $\exists\mathbb{R}$-complete by Abrahamsen, Miltzow and Seiferth[4], to ε-Robust Packing. The idea is similar to that of Abrahamsen, Miltzow and Seiferth, but much simpler as we allow non-convex polygons. The main difficulty in [4] is the need for a very complex fingerprinting method to force a specific configuration. In our non-convex case, we can solve this problem by using a much simpler "puzzle-piece" method: to force two polygons to be adjacent, we can cut a unique notch in one of the polygons, with a corresponding addition to the other polygon. We will ensure there is very little slack in terms of unused area, so that all of these notches do in fact have to be filled.

Linking two polygons

As examples of such a puzzle-piece notch, see Figure 3.4. For rational ε, these piece dimensions are indeed attainable. While there is some intentional slack in the relative position of A and B, this slack is solely perpendicular or tangential. In the other direction the ε-expansions of A and B fit snugly, with several tangent edges, since the difference in dimensions in this direction is exactly 2ε. If the puzzle piece is not tangent, and thus rotated, its horizontal and vertical cross-section becomes even larger, which no longer fits as the original spacing was already minimal.



(a) A perpendicular link, denoted $\dashv\vdash$.

(b) A tangential link, denoted $|\,|$.

Figure 3.4: Linking two polygonal pieces with slack δ.

Besides fixing the relative position of A and B in the fixed direction, such puzzle-piece notches also fix their relative rotation. By varying the width w and height h of each notch, we can ensure that if every notch is filled, it has to be filled by the corresponding piece. The secondary height h' of the thin part of the puzzle pieces can also be varied, but this is not required in our construction, as varying w and h is enough to uniquely identify each notch. Finally, the parameter δ dictates how much slack there is in the puzzle piece. This δ will be the same over almost all puzzle pieces.

Disregarding ε

In what follows, we will oftentimes disregard ε. By increasing the scale of the entire configuration, the impact of ε becomes relatively small. In other words, the ε-expansion of any large enough polygon is very similar to the original polygon, and the rounded corners of the ε-expansion become almost negligible. Therefore, we will often draw the ε-expanded version of a polygonal piece in the figures that follow. The reader should keep in mind that in actuality, the polygonal pieces are ε smaller on all sides. The scaling factor will be denoted by s.

Encoding variables

Suppose we are given an Range-Etr-Inv instance on n variables $x_1, x_2, \ldots, x_n$ with $|x_i| \leq \delta$ for all $1 \leq i \leq n$, where δ is some rational, only depending on n in a polynomial manner, that will be defined later. The problem of Range-Etr-Inv was introduced by Abrahamsen, Miltzow and Seiferth for the $\exists\mathbb{R}$-hardness reduction for regular Packing[4]. As mentioned, in Range-Etr-Inv, the variables are absolutely bounded as $|x_i| \leq \delta$, and two sorts of constraints can be imposed. Namely, addition constraints of the form $x_i + x_j = x_k$, and inversion constraints of the form $(1 + x_i)(1 + x_j) = 1$. We wish to reduce the given instance of Range-Etr-Inv to a polynomially sized instance of ε-Robust Packing. The following sections detail the construction of this reduced instance.

Initially, we represent the variables $x_1, x_2, \ldots x_n$ by n horizontal bars that are connected with perpendicular slack $2\delta s$ to a rectangular container, as depicted in Figure 3.5. Recall that s is the scaling factor of the configuration. The perpendicular links ensure the n bars are all horizontal, and we can choose the height of the bars in such a way that there is exactly 2ε space between them, as ε is assumed to be rational. With this bar representation, we can represent the value of x_i in a solution by the horizontal offset of bar i; we will assign $x_i = -\delta$ for the leftmost position, $x_i = \delta$ for the rightmost position and interpolate linearly in between. While not pictured in Figure 3.5, if there are constraints that require multiple copies of the same variable, such as $x_1 + x_1 = x_2$, we include several bars corresponding to that same variable, rigidly linked together to ensure they all in fact represent the same variable. This rigid linking can be done by combining part of the three bars into the same polygon. Note that this requires at most 3 copies of each variable, as there can be 3 copies of the same variable in an addition constraint, and no constraint acts on more than 3 variables.

In order to represent the constraints in the original Range-Etr-Inv instance, we add several addition and inversion gadgets to the initial bar representation. Such gadgets are constructed at the top of the rectangular container, and will restrict the horizontal freedom of the bars at the top according to their functionality. To add constraints to a set of bars that were not initially at the top, we require a swapping gadget to swap the vertical ordering of two adjacent bars. Such a swapping gadget can be seen in Figure 3.6. By ensuring that only one swap happens at a time, lengthening the entire configuration if needed, the tangential links at the top and bottom of the swapping gadget ensure that the bars are vertically aligned. This ensures proper functionality of the swapping gadget, as in this case the horizontal offsets, and thus the encoded variable values, are equal for the matching pairs of bars.

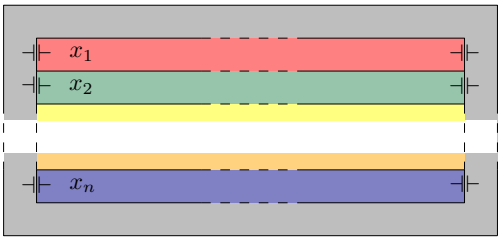


Figure 3.5: Variables encoded as horizontal bars.

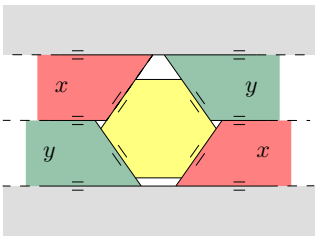


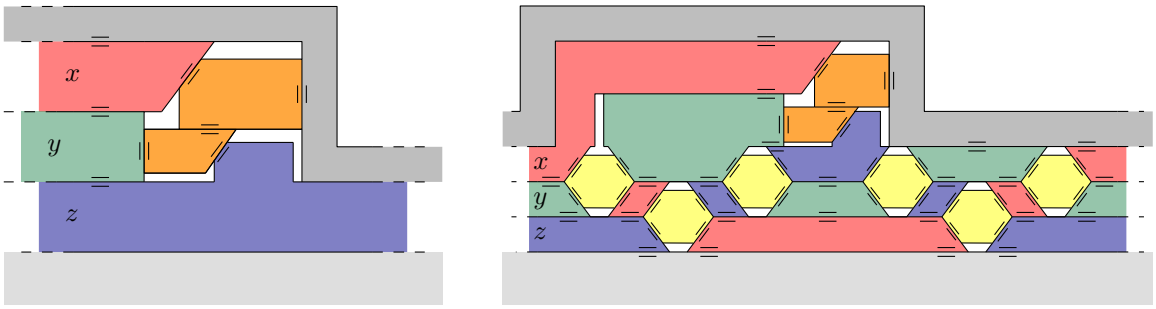
Figure 3.6: A swapping gadget.

One pitfall that should be pointed out here is that the slanted edges of the swapping gadget have to have specific slopes. For instance, if the edges are chosen to be at an angle of 45°, we cannot choose the rational coordinates of the swapping gadget in such a way that the distance between slanted edges also is exactly 2ε. In fact, if the slope is 1, distances between slanted edges can only be $q\sqrt{2}$ for some rational q. In particular, as we assume ε to be rational, the distance can indeed never be 2ε. This makes the construction in Figure 3.6 impossible, as the slanted tangential links are impossible to encode with rational coordinates. Instead, we choose all slanted edges to have a slope of $\pm\frac{4}{3}$. The fact that $(3, 4, 5)$ is a Pythagorean triple ensures that a distance of 2ε can in fact be achieved between corresponding slanted edges while using only rational coordinates, which in turn allows for slanted tangential links.

Using these swapping gadgets, we can move any set of bars to the top, where we use different gadgets to impose the addition and inversion constraints. As an example of how such swapping gadgets are used, see Figure 3.7b or Figure 3.8b.

Addition

To enforce addition constraints of the form $x + y = z$, we use the addition gadget shown in Figure 3.7. Figure 3.7a shows the main functional component of the addition gadget, while Figure 3.7b shows the full addition gadget, including all swapping gadgets required to connect the addition gadget to the entire configuration of bars. Note that, even though not all swapping gadgets have a bar both above and below them to guarantee vertical alignment as in Figure 3.6, all swapping gadgets work as desired as they have at least one of these two required bars which on its own is enough to guarantee vertical alignment. Again, all slopes are chosen to be $\pm\frac{4}{3}$ so that the slanted tangential links are feasible.



(a) Enforcing $x + y = z$.

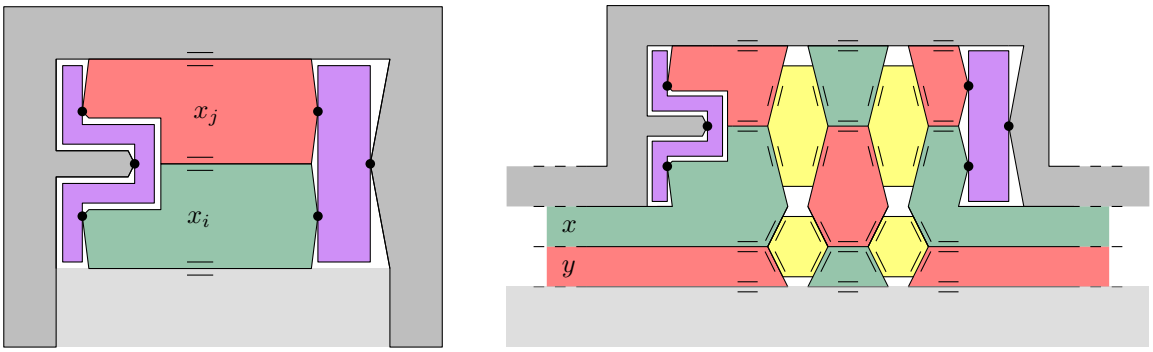
(b) The full gadget with required swaps.

Figure 3.7: An addition gadget.

Proper functionality of the addition gadget is relatively clear from Figure 3.7a. Because of the slanted tangential link with slope $\frac{4}{3}$, and since the top orange block is horizontally constrained, the vertical offset of both orange blocks is $\frac{4}{3}x$. Further, the horizontal offset of the bottom orange block is equal to y. Hence, as the bar corresponding to z is vertically constrained and connected to the bottom orange block by a tangential link with slope $\frac{4}{3}$, it indeed follows that $x + y = z$.

Inversion

To enforce inversion constraints of the form $(1 + x)(1 + y) = 1$, we make use of the inversion gadget shown in Figure 3.8. As with the addition gadget, Figure 3.8a shows the main functional component of the inversion gadget, while Figure 3.8b shows the full inversion gadget, including all swapping gadgets required to connect the inversion gadget to the entire configuration of bars. We require swapping gadgets to link the top bar of the inversion gadget to the variable bars below. Again note that the swapping gadgets function properly as they have at least one bar above or below to guarantee vertical alignment.

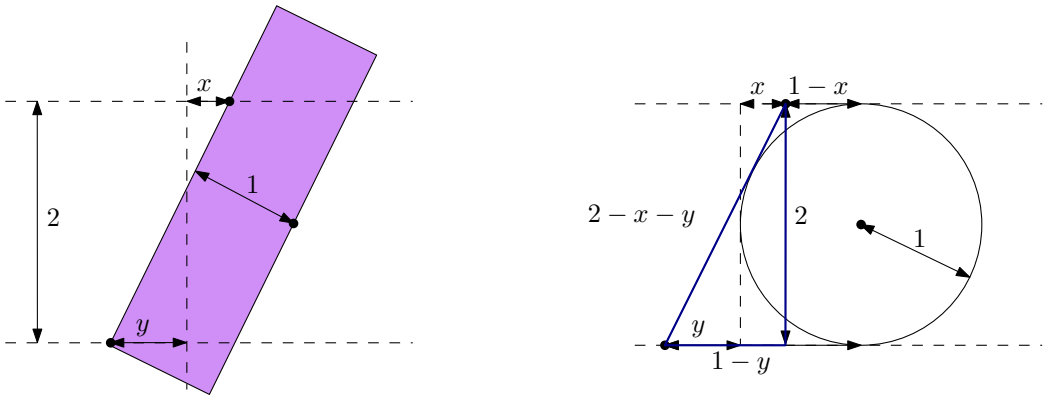


(a) Enforcing $(1+x)(1+y)=1$.

(b) The full gadget with required swaps.

Figure 3.8: An inversion gadget.

To see that the inversion gadget functions as desired, consider Figure 3.9. In Figure 3.9a, a schematic view of one of the two purple pieces in Figure 3.8a is given. The piece is constructed in such a way that the two tangent points on the left side have a relative vertical distance of 2, while the bar itself has a relative thickness that is almost 1. These distances are again relative to the scaling factor s, in the same way that the horizontal offset of the bars is s times the encoded variable value. In actuality these distances are $2s$ and s.



(a) The right piece of the inversion gadget.

(b) The piece as a circle's tangent line.

Figure 3.9: A schematic view of the inversion gadget.

It should be noted again that we consider the ε-expansions of the actual polygons as pieces; the actual bar has a thickness of $s-4\varepsilon$. The purple piece on the left of the inversion gadget in Figure 3.8a is constructed with similar dimensions. The vertical distance of 2 is guaranteed by the tangential links in Figure 3.8a. Note that the bar does not have a thickness of $s-2\varepsilon$, as might be expected, but is instead 2ε thinner. This is to ensure that the points of rotation function as proper rotation points, even though we take their ε-expansion. Namely, instead of taking the ε-expansion of both pieces, we can also not expand the pieces with the rotation points, and instead expand the bar by 2ε, resulting in an equivalent configuration. In this case, the 2ε-expansion of the bar properly rotates around the rotation points, which is required to show correctness. A graphical representation of this equivalent configuration can be seen in Figure 3.10.

Assuming that the two bars corresponding to x and y are tangent to one of the pieces of the inversion gadget, we can derive measurements as in Figure 3.9b. Here we have a right triangle with legs of length 2 and $|x-y|$ and hypotenuse $2-x-y$, highlighted in blue in Figure 3.9b. The hypotenuse length follows from the fact that the hypotenuse can be split into two tangent line segments to the unit circle, which have length $1-x$ and $1-y$ as from any given point both tangents to a circle have the same length. By the Pythagorean theorem we must have $4+x^2+y^2-2xy = 4+x^2+y^2-4x-4y+2xy$, from which it follows that indeed $(1+x)(1+y)=1$.

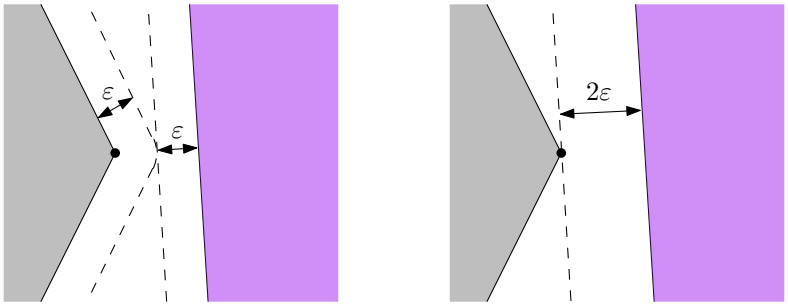


Figure 3.10: Taking two ε-expansions is equivalent to a single 2ε-expansion.

While we cannot necessarily guarantee that the bars corresponding to x and y are tangent to either one of the purple pieces of the inversion gadget, it turns out that both purple pieces together are enough to enforce equality. As the polygonal pieces in the inversion gadget must be disjoint, the purple piece on the left of Figure 3.8a enforces that $(1 + x)(1 + y) \geq 1$, while the purple piece on the right enforces $(1 + x)(1 + y) \leq 1$. Together, we see that the inversion gadget indeed enforces $(1 + x)(1 + y) = 1$.

Fitting the container

With the current construction, the container which should contain all the polygonal pieces is not square, while this is required in our formulation of ε-Robust Packing. Luckily, this is easy to remedy. Placing the container in the middle of a square of side length at least 4 times the width and height of the current container and cutting through the middle with a slope of $\frac{4}{3}$, as in Figure 3.11, we obtain two additional pieces. We can guarantee that these pieces are simple by cutting only until the outermost intersection points with the container. Up to rotation, there is only one unique way to pack these two polygons in the bounding square, shows in Figure 3.11, assuming the configuration is scaled such that ε is relatively small enough. The remaining polygons all have to be packed in the area that is left over, which is almost identical to the original container, besides two small gaps due to the rounded corners of the ε-expansions of the two large polygonal pieces. Again, if the configuration is scaled such that ε is relatively small these differences can be disregarded.

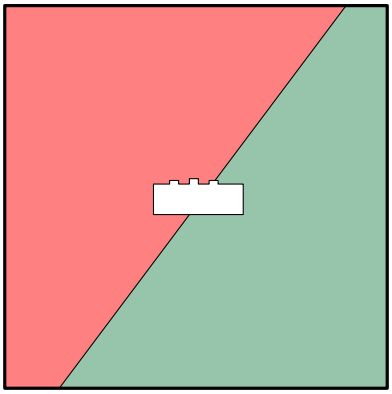


Figure 3.11: Two polygons to accomodate a square container.

Proof of Theorem 3.2.9

With the construction detailed above, we are ready to show $\exists\mathbb{R}$-completeness of ε-Robust Packing for rational ε.

Proof of Theorem 3.2.9. As mentioned at the beginning of this section, it suffices to show $\exists\mathbb{R}$-hardness. The construction above provides a reduction from any instance of Range-Etr-Inv, which is a known $\exists\mathbb{R}$-complete problem[4], to an instance of ε-Robust Packing. As the given Range-Etr-Inv instance has only $O(n^3)$ possible unique constraints, and as any constraint reduces to a single addition or inversion gadget, together with a possibly linear number of swapping gadgets to move the correct variables to the top, this reduction results in an instance of ε-Robust Packing with at most $O(n^4)$ polygons.

As long as our scaling factor s to disregard ε does not hyperexponentially depend on n, and δ depends polynomially on n, all of the polygon coordinates are polynomial in terms of n, which implies that the reduction is indeed of polynomial size. It remains to show that the reduction is in fact a proper reduction.

From the construction itself, it is clear that any satisfiable instance of Range-Etr-Inv reduces to a satisfiable instance of ε-Robust Packing. On the other hand, suppose we have a non-satisfiable instance of Range-Etr-Inv on n variables. Suppose for contradiction that there is some possible way to arrange the pieces generated by the reduction. If we can guarantee that all pieces have to be configured as in the construction, then the resulting ε-Robust Packing instance cannot be satisfied due to proper functionality of the addition and inversion gadgets. In order to show that all pieces have to be configured as in the construction, we first show that all perpendicular and tangential links have to be connected.

As the ε-expansions of the polygonal pieces have to be disjoint, we can precisely determine what area of the square will remain uncovered by all ε-expansions in any ε-robust configuration. If we disregard the rounded corners of the ε-expansions, all remaining uncovered area is $O(\delta s^2 n^4)$, as per required gadget the uncovered area is $O(\delta s^2)$ without the rounded corners. The rounded corners themselves add $O(\varepsilon^2 n^4)$ additional uncovered area, so that the total uncovered area is $O((\delta s^2 + \varepsilon^2)n^4)$. On the other hand, if a puzzle-piece notch is not filled, the uncovered area is $\Omega(wh) = \Omega(s^2)$. Hence, if we choose for instance $s = \Theta(2^n)$ and $\delta = \Theta(n^{-5})$ with suitable constants, this guarantees that every puzzle-piece notch has to be filled. Note that these choices of s and δ are small enough so that the reduction remains polynomial.

Now by choosing the dimensions of each puzzle-piece notch to be unique, as every notch has to be filled, each perpendicular and tangential link functions as desired, linking the two corresponding polygonal pieces. Through the functionality of these links, we see that almost all of the polygonal pieces have to match the configuration intended by the construction. Only the two pivoting pieces per inversion gadget can still be freely placed. In particular, we have only two types of polygons remaining. As the links fix the rotation of all of the other pieces, and their position is fixed up to a potential offset of $O(\delta s)$, the remaining spaces to place the remaining two types of polygons are all either on the left or right side of an inversion gadget. As the two types of polygon we still have to place are so different in their dimensions, with this difference in particular not depending on δ, the only possible places to put the remaining polygons is in the spots intended by the construction. Furthermore, each inversion gadget gets exactly one left pivoting piece and one right pivoting piece. Hence, all the inversion gadgets also have to function as desired. However, as we reduced from a non-satisfiable instance of Range-Etr-Inv, this is impossible. Hence, the reduction is indeed a proper polynomial reduction and ε-Robust Packing is $\exists\mathbb{R}$-complete for all $\varepsilon \in \mathbb{Q}_{\geq 0}$. □

3.2.5 Variants

While we chose a relatively intuitive manner to add robustness to the polygonal packing problem, with this notion the effect of ε is relatively negligible due to the rational scaling invariance. Several other ways of adding robustness exist that may seem just as reasonable, some intended to remedy this rational scaling invariance. We discuss two of these variants.

Fitting inside a unit square

One way of circumventing the scaling invariance is by restricting the bounding square to be of unit length. In this case, as the expansion of the input polygons increases their area by at least $\pi\varepsilon^2$ by the added circular arcs, we have a bound on the maximal number of input polygons as for n large enough we have $n\pi\varepsilon^2 > 1$ and the answer is trivially negative. However, we can still have a limited number of arbitrarily complex polygons. Another way to circumvent this restriction is by limiting the size of the bounding square, but by some factor that is dependent on n. For instance, packing n polygons into a container of side length n.

While this remedies the rational scaling invariance and introduces a simple range, the results on undecidability, the gap property and conjectured NP-membership carry over. The $\exists\mathbb{R}$-completeness result likely still holds, but requires a different proof as we can no longer scale the input to reduce the relative effect of ε.

Pieces of unit area

Another idea is to restrict the pieces themselves to have unit area. In this case, the problem does not become trivial. Instead, all of the same results hold.

For $\exists\mathbb{R}$-completeness, notches of width at most 2ε can be cut out of the polygonal pieces, without significantly impacting the expanded pseudo-polygons. If the notches are wedge-shaped, starting from a tiny opening on the boundary, the effect is negligible. The $\exists\mathbb{R}$-completeness result is modified by cutting out enough material from each polygon to achieve unit area.

The gap property and related results completely carry over, and so does the undecidability.

3.3 Framework Application: ε-Robust Art Gallery Problem

In this section, we study the problem of ε-Robust Art Gallery Problem; a robust variant of the Art Gallery Problem where perturbing all of the guards by a distance of at most ε does not change the validity of the solution. The Art Gallery Problem was shown to be $\exists\mathbb{R}$-complete by Abrahamsen, Adamaszek and Miltzow[3]. An example of ε-Robust Art Gallery Problem can be seen in Figure 3.12. The input consists of the number of guards k, encoded in unary, and a simple polygonal gallery with rational coordinates. As with ε-Robust Packing, we apply the main ideas of the framework to derive results on how the complexity of ε-Robust Art Gallery Problem depends on ε. Contrary to what we did for ε-Rudi and ε-Robust Packing, we will not show any results on $\exists\mathbb{R}$-completeness. The reason being that the $\exists\mathbb{R}$-completeness result of the art gallery problem is very involved and adapting it to the robust version would be outside the scope of this thesis. Instead, we focus on the less problem-specific results, as undecidability and NP-membership. The proofs in this section are very simple, but serve to show that application of the framework ideas can sometimes quickly lead to interesting results.

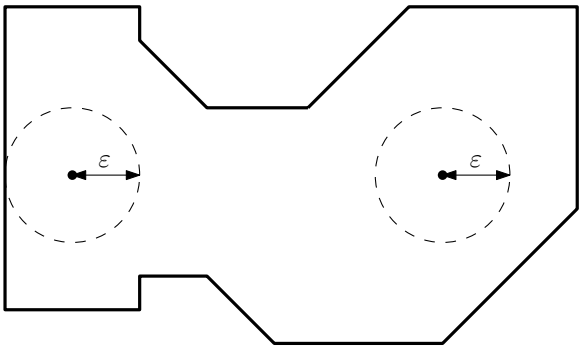


Figure 3.12: Robustly guarding a polygonal gallery.

3.3.1 Unary encoding of k

Famously, Chvátal [34] showed that the Art Gallery Problem can always be solved when $k \geq \lfloor n/3 \rfloor$, where n is the number of vertices of the gallery. Hence, without loss of generality we can assume that k is polynomially bounded by n, and hence also by the input size of the gallery. Whether we encode k in binary or unary therefore does not influence the asymptotics of the input size, and all complexity results hold in either case. Unfortunately, this is not as easily said for the ε-Robust Art Gallery Problem.

While it seems reasonable that any gallery that can be ε-robustly guarded, can also be ε-robustly guarded with a number of guards polynomial in the number of gallery vertices, we are unaware of any results to this end. Therefore, we cannot simply assume k to be polynomial in the input size of the gallery. If we allow k to be binary encoded in the input, k itself could be exponential in the input size, and as we will construct a binary witness that stores the position per guard, this witness could also be of exponential size. This is why we choose k to be unary encoded, so that the constructed binary witness will still be of polynomial size. Still, if one shows that any gallery that can be ε-robustly guarded, can also be ε-robustly guarded with a number of guards polynomial in the number of gallery vertices, then the same results would hold for binary encoded k.

3.3.2 Equivalence under rational scaling

Analogous to what we saw for ε-Robust Packing, as the size of the gallery is independent of ε, if we scale up the entire gallery by some constant $q \in \mathbb{Q}_{>0}$, the guards are still only required to have a positional freedom of ε. As with ε-Robust Packing, this is equivalent to guarding the original gallery with a robustness of $q^{-1}\varepsilon$. This implies the following theorem.

Theorem 3.3.1. *Let $\varepsilon \in \mathbb{R}_{\geq 0}$. For any $q \in \mathbb{Q}_{>0}$, the problems* ε-Robust Art Gallery Problem *and* $q\varepsilon$-Robust Art Gallery Problem *are computationally equivalent under polynomial time reductions.*

In particular, results on undecidability, NP-membership or $\exists\mathbb{R}$-completeness transfer from any specific ε to all rational multiples of ε as well. As we will see examples of nontrivial $\varepsilon > 0$, this implies ε-Robust Art Gallery Problem does not really have a simple range, as was the case for ε-Rudi.

It should be noted at this point that this computational equivalence holds in general for any problem in which the input can be scaled up by a factor q in such a way that the resulting instance is equivalent to the original instance, but with $q^{-1}\varepsilon$-robustness instead of ε-robustness.

3.3.3 Undecidability

As with ε-RUDI and ε-Robust Packing, while there are uncountably many choices for ε, there are only countably many possible algorithms. We again show that each ε requires a different algorithm by showing that the rigidity values for ε-Robust Art Gallery Problem lie dense in $\mathbb{R}_{\geq 0}$. This leads to the following result.

Theorem 3.3.2. *The problem of* ε-Robust Art Gallery Problem *is decidable for only countably many $\varepsilon \in \mathbb{R}_{\geq 0}$.*

As detailed in the general framework, this follows from the rigidity values for ε-Robust Art Gallery Problem lying dense in $\mathbb{R}_{\geq 0}$. As for ε-Robust Packing, this is relatively simple for ε-Robust Art Gallery Problem.

Lemma 3.3.3. *The rigidity values of* ε-Robust Art Gallery Problem *lie dense in $\mathbb{R}_{\geq 0}$.*

Proof. We construct galleries with rigidity value $r \in \mathbb{Q}_{\geq 0}$ for $k = 1$. A simple example is a $2r \times 2r$ square gallery. For $\varepsilon = r$, we can position the guard precisely in the middle of the gallery, but if $\varepsilon > r$, we can always perturb the guard to outside the gallery, from where the gallery is no longer guarded. As $\mathbb{Q}_{\geq 0}$ already lies dense in $\mathbb{R}_{\geq 0}$, certainly the full set of rigidity values of ε-Robust Art Gallery Problem also lies dense in $\mathbb{R}_{\geq 0}$.

□

3.3.4 The gap property and NP-membership

If an input I is r-rigid with $r > \varepsilon$, we can perturb the positions of all guards by at most $\delta = r - \varepsilon$ and still remain ε-robust, as the total perturbation of any guard will not exceed r by the triangle inequality. Similar to how we perturb the disks for ε-Rudi, this implies the following lemma.

Lemma 3.3.4. *Suppose I is an r-robust input of size n with $r > \varepsilon$. Define $\delta = r - \varepsilon$ and pick $m \in \mathbb{Z}_{\geq 0}$ with $2^{-m} \leq \delta$. Then there exists an ε-robust configuration in which all of the guards are positioned on a grid of size 2^{-m}.*

Proof. Completely analogous to the proof of Lemma 2.3.2 for ε-Rudi, in an r-robust configuration each guard has to move at most $2^{-m}/\sqrt{2} < \delta$ to the nearest grid point. After perturbing each guard to the grid the resulting configuration is still ε-robust by the triangle inequality. □

Lemma 3.3.4 immediately implies the following.

Theorem 3.3.5. ε-Robust Art Gallery Problem *has the gap property.*

Proof. Suppose the instance I has size n. Since the position of the guards can be bounded by the size of the gallery which is at most 2^n, Lemma 3.3.4 implies that each guard can be positioned in such a way that each coordinate is rational with numerator and denominator absolutely bounded by 2^{n+m} where $2^{-m} \leq \delta$. Hence, each guard can be encoded in a number of bits polynomial in $n + \log 1/\delta$. As we forced the encoding of k to be unary, we can also encode the position of every guard in a number of bits polynomial in $n + \log 1/\delta$. □

As ε-Robust Art Gallery Problem has the gap property, from our framework we again obtain the following two corollaries.

Corollary 3.3.6. *If $\varepsilon \geq 0$ is selected uniformly at random, then ε-Robust Art Gallery Problem has a binary witness almost surely.*

Corollary 3.3.7. *If ε is badly approximable and polynomial-time computable, then ε-Robust Art Gallery Problem is in* NP.

3.3.5 Variants

There are of course many different ways to add robustness to the Art Gallery Problem, some of which have been discussed in detail in prior literature. We discuss a variant related to our definition of ε-Robust Art Gallery Problem.

Perturbing guards inside the gallery

In our variant of ε-Robust Art Gallery Problem, guards are allowed to be perturbed outside of the gallery. It could however be intuitive to not allow such perturbations, instead restricting the guards to remain inside the gallery. For convex galleries, this implies infinite rigidity. However, infinite rigidity and convexity are equivalent and we can easily check for convexity in polynomial time. Thus, both problems are quite similar in terms of rigidity. It remains fairly easy to show that the rigidity values lie dense in $\mathbb{R}_{\geq 0}$. In this case, instead of a $2r \times 2r$ square gallery, we can construct a gallery as in Figure 3.13. For $\varepsilon = r$ the guard can still be placed perfectly in the middle, but for $\varepsilon > r$ we can always perturb the guard to one of the four outer lobes, from which the guard cannot guard the entire gallery.

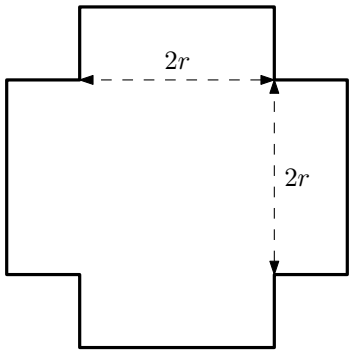


Figure 3.13: A gallery with rigidity $r \in \mathbb{Q}_{\geq 0}$ for $k = 1$.

To show the gap property, a similar result still holds, but is more involved. Namely, the closest grid point to a guard could lie outside of the gallery, preventing us from perturbing the guard to that grid point. However, depending on how "pointy" each vertex is, every guard is either close to a vertex, or has some half-disk with fixed radius centered at the guard contained entirely in the gallery. Note that we can bound the pointiness of a vertex based on the input size. Now, by choosing a sufficiently subdivided grid that contains all of the gallery vertices, we can guarantee that we can perturb every guard to a sufficiently close grid point: either the closest gallery vertex, or some grid point inside the half-disk centered at the guard that we know to be inside of the gallery. Said grid guarantees a polynomial representation due to the bound on vertex pointiness.

Discussion and Future Research

We have exposed an intriguing relation between the conjectured existence of a polynomial-time computable, badly approximable ε^* and the NP-membership of a whole class of ε^*-robust problems. Furthermore, the developed framework has opened a lot of new questions; we have seen applications of the framework on ε-Rudi, ε-Robust Packing and ε-Robust Art Gallery Problem, but there are certainly many other $\exists\mathbb{R}$-complete problems that admit a definition of robustness to which the framework can be applied.

While we have seen two examples of computable, badly approximable numbers, both are likely not computable in polynomial time. We have also seen an equivalence between badly approximable numbers and badly approximable sequences. While this equivalence has been useful to show the existence of a computable, badly approximable number, the same main difficulty that arises for showing that a specific number is badly approximable also is present when studying badly approximable sequences instead. Namely, the fact that we need to reason about the description length of *all* relevant sequences of algebraic numbers. In the case of a badly approximable sequence $\{s_n\}_{n=1}^{\infty}$, this difficulty arises from the fact that we not only want $L(s_n)$ to be large, we want $L(\beta)$ to be large for all algebraic $\beta \in [s_n - \frac{1}{2}, s_n + \frac{1}{2})$. In any equivalence, this problem will hold as there is no way to discretize these "continuous" requirements. Formulated differently, if there was an equivalent statement for which we only need to consider the description length of a single sequence of algebraic numbers, by the equivalence we also only have to check a single sequence of algebraic numbers to show a number is badly approximable, which is not true.

We also studied measures of complexity of a real number from two other fields of research, namely the minimal SLP length $\tau(n)$ and the measures of height $H(\alpha)$, degree $\deg(\alpha)$ and Mahler measure $M(\alpha)$ from diophantine approximation. Comparing these measures with the description length $L(\alpha)$, we derived relatively tight upper bounds on $L(\alpha)$ in terms of the other measures, but lower bounds on $L(\alpha)$ were exponentially worse. This is due to the fact that an ETR formula is more general than an SLP and a minimal polynomial, so that it is relatively simple to derive a corresponding ETR formula from either an SLP or a minimal polynomial. In the other direction, it is far more complicated to turn an arbitrary defining formula into an SLP or a minimal polynomial. In fact, we saw that in the case of height, degree and Mahler measure, we cannot do better than a lower bound on $L(\alpha)$ that is exponentially worse than the upper bound, as we can use the repeated squaring power of an ETR formula to construct an example where we cannot bound $L(\alpha)$ any better.

The absence of good lower bounds on $L(\alpha)$ in terms of the other measures of complexity implies that we cannot lift approximation results from SLPs and diophantine approximation to a result on badly approximable numbers. On the other hand, the upper bounds on $L(\alpha)$ imply that any results on badly approximable numbers does lift to results on SLPs and diophantine approximation. As such results have been sought for, this only further illustrates the difficulty of finding a polynomial-time computable, badly approximable number. Perhaps a topic of future research could be to try to lift the relatively good results on transcendence measures for π and e from diophantine approximation to badly approximable numbers, by modifying these proofs to suit description length instead. However, it is more likely that a completely novel approach is required.

While we were not able to show the existence of a polynomial-time computable, badly approximable number, the developed framework gives insight into many of the other aspects of complexity of an ε-robust problem. We have found relatively general results on undecidability and the existence of a binary witness and conjectured NP-membership. The existence of a binary witness depends on the very general gap property, which we saw holds for several ε-robust problems. While showing that an ε-robust problem has the gap property is problem-specific, we saw that it was relatively straightforward in our examples, with the exception that we had to force the number of guards k in the ε-Robust Art Gallery Problem to be unary encoded due to the absence of polynomial upper bound on the required number of guards in the ε-robust case.

Less straightforward are the framework results on $\exists\mathbb{R}$-completeness and the structure of the rigidity values required for the result on undecidability. Both of these results seem to be very problem-specific and sometimes very non-trivial. For ε-Rudi, we had to derive a cumbersome construction by enclosing a set of disjoint disks in a cycle of other disks to reason about rigidity values, and we were only able to show $\exists\mathbb{R}$-completeness for a single value of ε. While we expect these results to always be the most problem-specific of the framework, the fact that we can define a rigidity value through an ETR formula could indicate that more general results on rigidity are possible. For $\exists\mathbb{R}$-completeness we made the very broad conjecture that an ε-robust problem is $\exists\mathbb{R}$-complete whenever ε is a rigidity value, however this seems difficult to show. One possible idea is that an ε-robust representation of an instance with rigidity ε has to have some imposed rigidity, as with the needle that we used for ε-Rudi. Like with the needle, perhaps these rigid structures can be combined to construct gadgets for a reduction from an $\exists\mathbb{R}$-complete problem. There is the issue that not all ε-robust problems concern positioning objects in the plane, so that the required combination of these rigid structures itself is problem-specific. However, if for instance we restrict ourselves to a reduced set of ε-robust problems, such as those concerned with positioning points in the plane, there could be more merit in this idea.

In any case, the framework certainly serves as a useful tool in the study of the complexity of ε-robust problems, be it for specific problems, or for the general family of ε-robust problems.

Bibliography

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

[2] Jiří Matoušek. "Intersection graphs of segments and $\exists\mathbb{R}$". In: *arXiv* (2014). URL: http://arxiv.org/abs/1406.2636.

[3] Mikkel Abrahamsen, Anna Adamaszek, and Tillmann Miltzow. "The art gallery problem is $\exists\mathbb{R}$-complete". In: *J. ACM* 69.1 (2022), Art. 4, 70. ISSN: 0004-5411. DOI: 10.1145/3486220.

[4] Mikkel Abrahamsen, Tillmann Miltzow, and Nadja Seiferth. "Framework for ER-completeness of two-dimensional packing problems". In: *2020 IEEE 61st Annual Symposium on Foundations of Computer Science*. IEEE Computer Soc., Los Alamitos, CA, 2020, pp. 1014–1021. DOI: 10.1109/FOCS46700.2020.00098.

[5] Paul Jungeblut et al. "The Complexity of Robust $\exists\mathbb{R}$-Complete Problems".

[6] Jan Kratochvíl and Jiří Matoušek. "Intersection graphs of segments". In: *J. Combin. Theory Ser. B* 62.2 (1994), pp. 289–315. ISSN: 0095-8956. DOI: 10.1006/jctb.1994.1071.

[7] Marcus Schaefer. "Complexity of some geometric and topological problems". In: *Graph drawing*. Vol. 5849. Lecture Notes in Comput. Sci. Springer, Berlin, 2010, pp. 334–344. DOI: 10.1007/978-3-642-11805-0_32.

[8] Ross J. Kang and Tobias Müller. "Sphere and dot product representations of graphs". In: *Discrete Comput. Geom.* 47.3 (2012), pp. 548–568. ISSN: 0179-5376. DOI: 10.1007/s00454-012-9394-8.

[9] Colin McDiarmid and Tobias Müller. "Integer realizations of disk and segment graphs". In: *J. Combin. Theory Ser. B* 103.1 (2013), pp. 114–143. ISSN: 0095-8956. DOI: 10.1016/j.jctb.2012.09.004.

[10] William Evans et al. "Representing graphs and hypergraphs by touching polygons in 3D". In: *Graph drawing and network visualization*. Vol. 11904. Lecture Notes in Comput. Sci. Springer, Cham, 2019, pp. 18–32.

[11] Michael Hoffmann et al. "Recognition of Unit Segment and Polyline Graphs is $\exists\mathbb{R}$-Complete". In: *Arxiv* abs/2401.02172.2401.02172 (2024), pp. 1–18. DOI: 10.48550/arXiv.2401.02172.

[12] Marcus Schaefer, Jean Cardinal, and Tillmann Miltzow. "The Existential Theory of the Reals as a Complexity Class: A Compendium". In: *Arxiv* abs/2407.18006.2407.18006 (2024), pp. 1–126. DOI: 10.48550/arXiv.2407.18006.

[13] Carlos G. T. de A. Moreira. "On Asymptotic Estimates for Arithmetic Cost Functions". In: *Proceedings of the American Mathematical Society* 125.2 (1997), pp. 347–353. URL: http://www.jstor.org/stable/2161660.

[14] Welington de Melo and Benar Svaiter. "The cost of computing integers". In: *Proceedings of the American Mathematical Society* 124 (1996). DOI: 10.1090/S0002-9939-96-03173-5.

[15] Pascal Koiran. "Valiant's Model and the Cost of Computing Integers". In: *Computational Complexity* 13 (2004), pp. 131–146. DOI: 10.1007/s00037-004-0186-2.

[16] Michael Shub and Steven Smale. "On the intractability of Hilbert's Nullstellensatz and an algebraic version of "P=NP"". In: *Duke Mathematical Journal* 81 (1995), pp. 47–54. DOI: 10.1215/S0012-7094-95-08105-8.

[17] Peter Bürgisser and Gorav Jindal. "On the Hardness of PosSLP". In: *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1872–1886. DOI: 10.1137/1.9781611977912.75. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611977912.75.

[18] Pranjal Dutta, Nitin Saxena, and Amit Sinhababu. "Discovering the Roots: Uniform Closure Results for Algebraic Classes Under Factoring". In: *J. ACM* 69.3 (2022). DOI: 10.1145/3510359. URL: https://doi.org/10.1145/3510359.

[19] Harald A. Helfgott. "The ternary Goldbach conjecture is true". In: *Arxiv* abs/1312.7748.1312.7748 (2013), pp. 1–79. DOI: 10.48550/arXiv.1312.7748.

[20] Charles-Jean de la Vallée Poussin. "Sur la fonction $\zeta(s)$ de Riemann et le nombre des nombres premiers inférieurs à une limite donnée". In: *Mémoires couronnés de l'Académie de Belgique* 59 (1899), pp. 1–74.

[21] Paul Pollack. *Not Always Buried Deep*. American Mathematical Society, 2009.

[22] Klaus F. Roth. "Rational approximations to algebraic numbers". In: *Mathematika* 2.1 (1955), pp. 1–20. DOI: 10.1112/S0025579300000644.

[23] Clive S. Davis. "Rational approximations to e". In: *Journal of the Australian Mathematical Society* 25.4 (1978), pp. 497–502. DOI: 10.1017/S1446788700021480.

[24] Yann Bugeaud. *Approximation by Algebraic Numbers*. Cambridge Tracts in Mathematics. Cambridge University Press, 2004.

[25] Anne-Maria Ernvall-Hytönen, Tapani Matala-aho, and Louna Seppälä. "On Mahler's Transcendence Measure for e". In: *Constructive Approximation* 49.2 (2019), pp. 405–444. DOI: 10.1007/s00365-018-9429-3. URL: http://dx.doi.org/10.1007/s00365-018-9429-3.

[26] Kurt Mahler. "On two extremum properties of polynomials". In: *Illinois Journal of Mathematics* 7.4 (1963), pp. 681–701. DOI: 10.1215/ijm/1255645104. URL: https://doi.org/10.1215/ijm/1255645104.

[27] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra*. 3rd ed. Cambridge University Press, 2013.

[28] Kurt Mahler. "An Inequality for the Discriminant of a Polynomial". In: *Michigan Mathematical Journal* 11 (1964), pp. 257–262.

[29] Maurice Mignotte. "Some Useful Bounds". In: *Computing* 4 (1982), pp. 259–263.

[30] Kurt Mahler. "Arithmetische Eigenschaften einer Klasse transzendental-transzendenter Funktionen". In: *Mathematisches Zeitschrift* 32 (1930), pp. 545–585.

[31] László Fejes Tóth. *Lagerungen in Der Ebene Auf Der Kugel und Im Raum*. Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen mit besonderer Berücksichtigung der Anwendungsgebiete. Springer Verlag, 1953.

[32] Peter Taylor. "Triangular lattice points close to the origin". In: (2018). URL: http://cheddarmonk.org/papers/triangle-lattice.pdf.

[33] Peter Bürgisser and Felipe Cucker. "Exotic quantifiers, complexity classes, and complete problems". In: *Found. Comput. Math.* 9.2 (2009), pp. 135–170. ISSN: 1615-3375. DOI: 10.1007/s10208-007-9006-9.

[34] Václav Chvátal. "A combinatorial theorem in plane geometry". In: *Journal of Combinatorial Theory, Series B* 18.1 (1975), pp. 39–41. ISSN: 0095-8956. DOI: https://doi.org/10.1016/0095-8956(75)90061-1. URL: https://www.sciencedirect.com/science/article/pii/0095895675900611.