Computational Complexity Theory for Advanced Function Spaces in Analysis

vom Fachbereich Mathematik
der Technischen Universität Darmstadt
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)
genehmigte Dissertation

Tag der Einreichung:	07.09.2016
Tag der mündlichen Prüfung:	02.11.2016
Referent:	Prof. Dr. Martin Ziegler
1. Korreferent:	Prof. Dr. Ulrich Kohlenbach
2. Korreferent:	Prof. Dr. Akitoshi Kawamura

von

Diplom-Mathematiker Florian Steinberg
aus Frankfurt a. M.

Darmstadt, D 17
2017

Acknowledgments

I would like to thank my supervisors Martin Ziegler, Akitoshi Kawamura and Ulrich Kohlenbach for making this dissertation possible. I would like to thank Martin Ziegler for insightful discussion of the contents of my research, for many suggestions of further research and for keeping me from straying to far. I would like to thank Akitoshi Kawamura for being an excellent host during my stays in Japan and for providing many useful tips and valuable ideas and inspirations regarding my research. I would like to thank Ulrich Kohlenbach for making me see the big picture and pointing out many connections to other fields of research as well as in depth discussion of my research. I would like to thank the IRTG 1529 and all the people involved with it for giving me the opportunity to pursue the research mostly without additional duties and while being able to gather new experiences abroad. In particular I would like to thank the IRTG 1529 for paying for my research stays in Japan and Korea and the travel expenses for many conferences. I would like to thank the community of my field of research for providing a healthy academic environment. I would like to thank Arno Pauly, Holger Thies and Matthias Schröder for the many hours of discussion. Some others whose input I deeply appreciate are: Hugo Férée, Eike Neumann, Norbert Müller, Franz Brause, Patrick Tolksdorf, Martin Rapp/Bolkart, and those I forgot. I would like to thank Christoph Dittman and Friederike Steglich for reading through parts of my thesis while not being from the field. I would like to thank the working group of logics at TU Darmstadt for the support. I appreciate that I got the opportunity to organize a conference toghether with the other PhD students: Angeliki Koutsoukou-Argyraki, Julian Bitterlich, Felix Canavoi and Daniel Körnlein. These people also patiently listened to my ramblings during the sessions of the "Doktorandentreff" which I deeply enjoyed. Thanks for that, and also for interesting discussion of other research topics. Special thanks go to Angeliki for starting the "Doktorandentreff". I would like to thank my friends both in Germany and Japan, and my family for giving me the support I needed to finish this project.

Darmstadt, February 1, 2017

Deutsche Zusammenfassung

Motivation für diese Doktorarbeit ist das Vorantreiben der Suche nach einem auf einem realistisches Maschinenmodell basierenden, mathematisch rigorosen Rahmen für Effizienzüberlegungen über die Numerik partieller Differenzialgleichungen. Während die Berechenbarkeitstheorie für kontinuierliche Strukturen auf die meisten Situationen anwendbar ist und beständig weiter entwickelt wird, ist es selbst in simplen Fällen oft noch unklar, welche Algorithmen als effizient zu gelten haben.

In dieser Arbeit wird das Problem im Rahmen der Darstellungen zweiter Ordnung angegangen. Diese beziehen sowohl ihren Namen, als auch ihren Begriff von Polynomialzeitberechenbarkeit aus der Komplexitätstheorie der Funktionale auf dem Baire Raum, auch Komplexitätstheorie zweiter Ordnung genannt. Als Maschinenmodell dienen Orakel-Turingmaschinen. Die Darstellungen zweiter Ordnung bieten viel Freiraum in der Gestaltung von Rechenmodellen und auch einen Grundstock an Konstruktuionen und behandelten Problemen. Insbesondere existiert eine etablierte Darstellung für das Rechnen mit stetigen Funktionen auf dem Einheitsintervall. Diese ist als die schwächste charakterisiert, die eine Auswertung in Polynomialzeit möglich macht.

Als ein erster Schritt wird die schwächste Darstellung zweiter Ordnung für die integrierbaren Funktionen angegeben, die es ermöglicht in Polynomialzeit Integrale auszuwerten. Unglücklicherweise erweist sich diese als unstetig und damit ungeeignet. Auf der Suche nach einer geeigneteren Darstellung werden die generellen Beschränkungen des Rechnens mit Darstellungen zweiter Ordnung auf metrischen Räumen untersucht. Dabei spielt die von Kolmogorov eingeführte metrische Entropie eines kompakten metrischen Raumes eine tragende Rolle. Es stellt sich heraus, dass es einen direkten Zusammenhang zwischen dieser und der Existenz 'kurzer' Darstellungen gibt, bezüglich derer sich die Metrik 'schnell' berechnen lässt. Um diese Resultate auf Räume integrierbarer Funktionen anwendbar zu machen, werden quantitative Versionen der Klassifikationsresultate kompakter Teilmengen, bekannt als die Sätze von Arzelà-Ascoli und Fréchet-Kolmogorov, untersucht.

Auf die gewonnen Erkenntnisse gestützt wird eine Familie von Darstellungen für L^p-Räume konstruiert. Es wird beweisen, dass diese berechenbar äquivalent zu den Darstellungen selbiger Räume als metrische Räume sind und gezeigt, dass sie die L^p-Norm in Exponentialzeit berechnenbar machen. Dies ist auch die Komplexität der Supremums Norm auf den stetigen Funktionen. Eine ähnliche Konstruktion führt zu Darstellungen von Sobolev-Räumen. Die Berechenbarkeit unterschiedlichster Operationen in Polynomialzeit wird gezeigt, aus technischen Gründen nur für den eindimensionalen Fall.

Abschließend wird ein Resultat präsentiert, das die Schwierigkeit des Berechnens des Lösungsoperators zum Dirichlet-Problem für Poisson's Gleichung auf der Einheitskugel gleichsetzt mit der Schwierigkeit eine stetigen Funktion zu integrieren. Zum Vergleichen von Schwierigkeiten von Aufgaben werden Polynomialzeit-Weihrauch-Reduktionen verwendet.

Translation of the German Abstract

This PhD thesis presents progress in the search for a mathematical rigorous framework for efficient numerics of partial differential equations based on a realistic machine model. While the computability theory of continuous structures is well developed and still an active field of research, in most settings it remains unclear what computations should be considered feasible.

This problem is tackled within the framework of second-order representations. The name as well as the notion of polynomial-time computability of the framework is inherited from the complexity theory of functionals on the Baire space, also called second-order complexity theory. As model of computation we use oracle Turing machines. Second-order representations offer a great deal of freedom in developing models of computation and a supply of well investigated structures. In particular, there exists an established second-order representation of the set of continuous functions on the unit interval. This representation has been classified as the weakest representation such that evaluation is possible in polynomial time.

As a first step we specify the weakest representation of the integrable functions such that integration is polynomial-time computable. This representation turns out to be discontinuous and therefore not suitable. We go on to explore the general restrictions of bounded-time computations on metric spaces within the framework of second-order representations. The notion of metric entropy, originally introduced by Kolmogorov, is used to classify those compact metric spaces that allow for a short representation such that the metric is computable efficiently. To be able to apply the results to spaces of integrable functions we investigate quantitative versions of classification results of their compact subsets called the Arzelà-Ascoli Theorem and the Fréchet-Kolmogorov Theorem.

We use the above to propose a family of representations for L^p-spaces. These are shown to be computably equivalent to the standard representations of the same spaces as metric spaces. Furthermore, we prove that the norm can be computed in exponential time. This is also the case for the supremum norm on the continuous functions on the unit interval. A similar family of representations is presented for Sobolev spaces. These representations are investigated in some detail. Several operators on Sobolev spaces are proven to be polynomial-time computable with respect to these representations. For technical reasons only the one-dimensional case is discussed.

Finally, we present a result that classifies the computational complexity of the solution operator of the Dirichlet Problem for Poisson's Equation on the unit disk as that of integrating a continuous function. As a tool for the comparison, polynomial-time Weihrauch reductions are used.

Contents

1 Introduction

1.1 Background

Computability theory provides a mathematical framework for algorithmic considerations about discrete structures. The merits are well known and computability theory has many applications. The most important applications are proofs of non-existence of algorithms (or real-world processes, assuming the Church-Turing thesis) to solve certain problems. The most famous such result is the undecidability of the halting problem. It has concrete implications for programming practice: Automatic checks for correctness are impossible in general due to the halting problem. This has seeded whole fields of research like model checking. In its traditional form computability theory is only applicable to discrete structures. Many applications, however, attempt to model computations on continuous structures like the real numbers, where classical computability theory cannot be used.

Complexity theory is the resource sensitive refinement of computability theory. For a decidable problem it asks about the time or memory space needed to decide the problem. Here, the time or space granted to a machine depends of the size on the input. The most important class of problems from complexity theory is the class of polynomial-time decidable problems. The polynomial-time decidable problems are in good accordance with those problems that are feasible in practice, this is know as the Cobham-Edmonds thesis. However, over the years many other important complexity classes have been considered. Complexity theory classifies problems into these classes using reductions. For many of these classes, the question whether or not they can be separated are longstanding problems that are considered difficult. A common kind of result from complexity theory is to prove that a problem is one of the hardest problems in a complexity class in the sense that every other problem in this class is reducible to this problem. Similar to the situation for computability theory, the main field of application of complexity theory are discrete structures.

Before we get into the technical details of how to do computability and complexity theory on the real numbers, let us sketch why the elaborate models used are necessary. Numerical analysis is well known for its huge success in solving problems relevant to applied sciences like engineering, so why bother to deal with more complicated mathematical descriptions than the ones that are currently in use? When numerical analysis solves a problem an engineer poses, this task consists of two parts: One is to implement an algorithm that produces a solution of the problem within a reasonable amount of time. The other is to give a mathematical description of both the problem and an algorithm solving it, prove convergence and provide bounds on the speed of convergence. Of course these tasks are heavily interdependent, from case to case the mathematical algorithm might exists first and be followed by an implementation or the algorithm may exist first and its mathematical justification only follow at a later point in time.

Computable analysis and its resource sensitive refinement real complexity theory provide a mathematically rigorous way to translate between the two tasks of implementation and mathematical justification. Some may claim that a proper solution of the latter task can be transformed into one of the former already without computable analysis. To see that using the traditional practices of numerical analysis the relation of these two tasks is not as tight as some may believe it to be, consider the following: The most common way to implement an algorithm handling real numbers on a computer is to use floating point arithmetic. Floating point arithmetic, however, defies a description accessible to mathematical reasoning: For instance it is problematic due to its lack of associativity, distributivity and its 'uncontrolled' error propagation. The mathematical description of algorithms does usually not consider these deficiencies. In extreme cases this can lead to a complete discrepancy of the outcome of an implementation and the mathematical prediction. Consider for instance the iterations of the logistic map, which turn up as description of biological systems. A straightforward implementation of the iteration

$$x_0 := 0.5 \quad \text{and} \quad x_{n+1} := 3.75 x_n (1 - x_n)$$

in floating point arithmetic does not deliver any significant digits for x_{100} (compare for instance the introduction of [Sch02]). The results obtained change under changing the order in which the multiplications are carried out. This goes unnoticed for an uneducated user: He might rely on the outcome of his first implementation.

Computer Scientists are well aware of these issues. Attempts to solve them led the use of multiple precision arithmetic and correct rounding (for instance MPFR), interval arithmetic etc. **Computable analysis** offers a full solution by specifying a mathematically strict model of computation on continuous structures by means of Turing machines. As such the model is in principle implementable (implementations are available in projects as iRRAM and AERN). One of the big merits of such a model is that it is possible to give rigorous proofs of computability and incomputability. Computable analysis is very successful in proving results that match up with the expectations of researchers from the field of numerics: One of the first things that students are told in a numerics lecture is not to test for equality of real numbers, as this often leads to errors, but to use an epsilon test instead. In computable analysis it is possible to prove undecidability of equality of real numbers but computability of epsilon tests. Another result illustrating the alignment is that in computable analysis from a symmetric matrix the eigenvalues can be computed, the eigenvectors cannot in general, but can be obtained if the cardinality of the spectrum (i.e. the number of distinct eigenvalues) is known [ZB04]

Computable analysis is not restricted to real numbers and matrices: It can be applied to arbitrary separable metric spaces with a specified dense subsequence and a general function space construction is available. This makes it applicable to almost all spaces of practical interest. How to compute in more elaborate spaces (like spaces of distributions) and computability of operators relevant to applications are active fields of research [WZ07; BY06; ZW03; KS05]. Computable analysis sits in between computability Theory, numerical analysis and Functional analysis and many important lessons can be learned. For instance proofs of incomputability of operators are often due discontinuity and can be traced back to use of the wrong function spaces. After consulting with analysts and reevaluating, they can usually be turned into computability results (for discussion of an example see [WZ02]). Furthermore, numerical analysis provides a lot of algorithms and knowhow that one can attempt to translate. Sometimes a translation is possible, in other cases no or no straightforward translation is possible. In any case usually additional insight is the outcome.

While computable analysis is a well-accepted foundation for computability on continuous structures it is also desirable to be able to talk about the speed of algorithms, i.e. their computational complexity. A framework general enough to have the potential of being able to treat complexity theoretical considerations about a significant number of problems that are relevant in practice has only recently been devised by Kawamura and Cook. The delayed development is partially due to a mismatch of the function space construction with computational complexity theory. This mismatch makes it necessary to use second-order complexity theory, that is complexity theory for operators on string functions, to be able to handle operators on real functions. While higher-order complexity theory delves of into the more theoretical regions of computer science, second-order complexity theory still has a concrete meaning: It talks about running times of oracle Turing machines that can be interpreted as programs with function calls. Here, the use of oracles reflects that the time needed to produce the answer of the function call is not counted towards the time consumption of the run of the program. Furthermore, the program is granted more time depending on how big the return value of the function call is.

This theory, we call it **real complexity theory** from now on, is rather well accepted as foundation whenever it is applicable. However, the construction for separable metric spaces from computable analysis only generalizes in very special cases and, as mentioned before, the function space construction does also not generalize. Thus, the number of situations it is applicable to is drastically reduced as compared to computable analysis. While recovering the function space construction is impossible, the lack of applicability on spaces similar to the continuous functions on the unit interval is most probably due to the lack of recipes for constructions and examples for how to treat these spaces within the framework.

An important part of any computational complexity theory is the comparison of algorithms, the classification of the difficulty of tasks and proofs of hardness of certain problems. The tool for comparison of tasks in computable analysis is the Weihrauch reduction. Many classification results have been produced using Weihrauch reductions and also the structure of the lattice of Weihrauch degrees is well investigated. Due to the recent development and acceptance of a general framework for real complexity theory, the polynomial-time counterpart has not caught much attention yet.

Motivation for the investigations

While computable analysis is accepted as a foundation of algorithmic considerations, for real complexity theory there is some convincing left to do. This problem is tackled in this thesis. One of the reasons for the lower acceptance of real complexity theory compared to computable analysis is that early results were in conflict with the expectations of many researchers: Maximization was declared to be a difficult problem, integration even harder. In numerics it is widely regarded that efficient integration is possible or at least that the problem of integration is less difficult than maximization. However, there are several ways to resolve these discrepancies without discarding the model of computation: One of the most frequent arguments why the results are not reflected in applications is that only very special functions show up in the instances considered in practice. Some efforts have been made to address this. A function space that is sufficiently rich for applications and sufficiently restrictive to allow polynomial-time integration, however, has, up to the knowledge of the author, not been discovered yet.

This thesis takes another stance: While the mode of computation on the continuous functions has been fixed and does not allow polynomial-time integration, it is debatable whether people really want to compute on the continuous functions, or to compute on the continuous functions in this way. The framework of second-order representations provides good reasons for the representation of the continuous functions to be a canonical choice, but in no way does it say that it is the only correct choice. On the contrary: It leaves enough spaces for modifications and [Kaw11] for instance encourages the use of different representations on the same space to consider relative or absolute errors etc. This thesis constructs and provides methods to construct representations of spaces that turn up in numerics, a particular source of inspiration being the field of partial differential equations.

Another valuable tool are polynomial-time Weihrauch reductions: They offer translations between algorithms while preserving efficiency. For this it is not necessary that the input algorithm is considered efficient from the point of view of real complexity theory. For instance an implementation of the integration operator that is efficient for a class of functions is translated to a solution of another problem provided that the polynomial-time Weihrauch reduction preserves the (yet unknown) class of functions the integration operator is efficient on. In this case an efficient algorithm is produced. If the algorithm that is produced turns out to lack efficiency, we have learned a valuable lesson about the class of functions integration is efficiently solvable on.

1.1.1 Historical digest and references

Computable analysis is said to originate from the very paper that is also one of the foundations of discrete computability Theory: The famous 1936 work by Turing [Tur36]. This paper, or to be more specific its errata [Tur37], where Turing accredits Brouwer with the right idea, introduced a notion of how to compute a real number and carried out first, very restricted, investigations of computability of functions on the real numbers. A fully grown computability theory for functions on the reals was later devised by Grzegorczyk [Grz55]. Hauck, Kreitz and Weihrauch introduced and investigated representations (these correspond to what in this thesis is called Cantor space representation), thereby generalizing applicability from the real numbers to more general classes of topological spaces. A detailed description of this framework can be found in [Wei00]. The underlying ideas were previously already present in constructive mathematics: For instance in the works of Troelstra [Tro69] and Troelstra and Dalen [TD88a; TD88b] which are based on ideas by Brouwer. Also see [Bee85]. A comparison of the settings can be found in [Tro92]. With some care these representations also made it possible to talk about complexity of elements of and functions between some represented spaces.

Further complexity theoretical considerations were carried out by Friedman [Fri84] and by Friedman and Ko [KF82] and are gathered in the book [Ko91]. Friedman and Ko in particular provided a connection between complexity theoretical properties of the maximization and integration operators on the continuous functions and the famous $\mathcal{P}$ vs. $\mathcal{NP}$ resp. the lesser known but stronger $\mathcal{FP}$ vs. $\#\mathcal{P}$ problem. While being strictly pointwise considerations, these results made it seem improbable that a polynomial-time algorithm for integration could be found without resolving one of the millennium problems. At this point in time, however, it was not even clear what a polynomial-time algorithm transforming functions to functions is: While the theory of representations provides a reasonable complexity theory for real numbers, and a computability-wise well behaved function space construction is available, the complexity theory of this function space representation was soon realized to be ill-behaved. Indeed: There is no representation (in the sense of Weihrauch) of the space of continuous functions on the unit interval such that the evaluation is polynomial-time computable.

Only recently were these problems overcome by an extended framework introduced by Kawamura and Cook [KC10; Kaw11]. This framework replaces the infinite strings that Weihrauchs representations use to encode elements of continuous structures by string functions and replaces the TTE-Machines used to realize functions between represented spaces by oracle Turing machines. To define running times, second-order complexity theory is applied. While higher-order complexity is still critically discussed, an accepted complexity theory for functionals on Baire space has been established by Mehlhorn [Meh76] and a characterization by resource bounded oracle Turing machines was provided by Kapron and Cook [KC96].

Many of the non-uniform results Friedman and Ko proved carry over to this uniform setting. Whereby the dependency on the millennium problem is removed: It can be proven that any algorithm for integration takes at least exponential time. This does not solve the millennium problem as exponential-time operators may still carry all polynomial-time instances to polynomial-time instances.

1.1.2 Organization of the thesis

This chapter gives an overview over the content of the thesis and specifies what background knowledge is necessary for the understanding of each part.

The first chapter contains introductory materials. A description of the backgrounds and a brief historical digest are followed by an overview over the contents and results of the thesis and some basic notational conventions. Finally, it contains a comprehensive introduction of the framework of second-order representations which are needed throughout all chapters but Chapter 4 and parts of Chapter 6. It assumes the reader to be familiar with classical (that is discrete) computability and complexity theory and to have a general background in mathematics.

The second chapter investigates representations that provide minimal information needed to make an operation polynomial-time computable. The first part recalls the construction of the standard representation of the continuous functions on the unit interval, that it can be characterized as minimal such that evaluation is polynomial-time computable and finally repeats some of the results regarding the complexity of integration for motivational purposes and later reference. The second part constructs the minimal representation of the integrable functions such that integration is polynomial-time computable and proves it to be discontinuous and therefore inappropriate. This chapter requires some additional knowledge of analysis. However, Riemann integration should suffice for the understanding of most parts.

The third chapter recalls the metric entropy of a compact subset of a metric space and investigates relations between this concept and the existence of metrics of low complexity. While the first part only requires familiarity with the theory of metric spaces, the later chapters are quite heavy on computability and complexity theory.

The fourth chapter turns to L^p-spaces. It recollects a quantitative version of the Arzelà-Ascoli Theorem from approximation theory. It then introduces the L^p-modulus and some tools to finally provide an analogous quantitative version of the Fréchet-Kolmogorov Theorem. While this chapter is a lot heavier on analysis (basic knowledge about L^p-spaces is needed, and of Sobolev spaces and convolution is beneficial for the understanding) it is mostly constructive analysis and cuts back on computer science: It does not use any computability and complexity theory outside of examples and applications based on the previous chapters.

The fifth chapter defines representations of L^p and Sobolev spaces and investigates their properties. It shows that the representation of L^p is computably equivalent to the standard representation of the space as metric space and provides several polynomial-time computability results for the representations of the Sobolev-spaces. This chapter heavily relies on all of the previous chapters and uses all of the available tools.

The sixth chapter compares operators. For this purpose polynomial-time Weihrauch reductions are recollected. It classifies the integration operator to be polynomial-time Weihrauch equivalent to a canonical counting operator on Baire space. This extends the list of uniformizations of the well-known results of Friedman and Ko by the result relating the integration operator to the $\mathcal{FP}$ vs. $\#\mathcal{P}$ problem. It also investigates some multiplication operators on the spaces introduced in the previous chapter and finally proves that the solution operator of the Dirichlet Problem for Poisson's equation on a ball of arbitrary dimension is polynomial-time Weihrauch equivalent to the integration operator. It uses all tools used in the previous chapters. Additional knowledge about the basic solution theory of partial differential equations like Green's functions and in particular the solution theory of Poisson's equation, like the principle of image charges, is beneficial but not necessary for the understanding.

Achievements

The parts that the author considers to be of particular interest and original to this thesis are the following:

1. Section 2.2: The discovery and investigation of the singular representation (see Definitions 2.2.8 and 2.2.11).
    a) The proof that the singular representation is the weakest representation of L^1 such that the integration operator is polynomial-time computable (Theorem 2.2.12).
    b) The proof that the singular representation is discontinuous in both the norm topology (Proposition 2.2.13, Theorem 2.2.14) as well as the weak topology (Theorem 2.2.15).
    c) A concrete description of the weakest representation computably equivalent to a given representation such that a given mapping is polynomial time computable (Proposition 2.3.5). The specification of the weakest representation of L^1 computably equivalent to the standard representation such that integration is possible in polynomial time (Theorem 2.3.8).
2. Chapter 3: The investigations of the tight connection between the metric entropy of a compact metric space and existence of short representations such that the metric has low space or time complexity (Theorem 3.2.2, Theorem 3.3.7).
    a) Specifying the exact relation between a modulus of continuity of a function between metric spaces and the increase in entropy when going from a set to its image under the function (Theorem 3.1.15).
    b) The construction of short communication functions from Proposition 3.2.6 and its application to prove that a running time bound of a metric restricts the size of the sets of elements with short names (Theorem 3.2.7). Also the repercussions for compact metric spaces (Corollary 3.2.9) and in particular the derived lower bound on the length of names of an open representation on compact sets of a specified size (Theorem 3.2.11).
    c) The construction of a family of representations of a specified length such that the metric is computable in a time only depending on the metric entropy of the space (Theorem 3.2.13), its improvement to space bounded computation (Theorem 3.3.12).
    d) The characterization of compact metric spaces with polynomial metric entropy as exactly those that allow a Cantor space representation such that the metric is polynomial-time computable with respect to an oracle (Theorem 3.3.4).
3. Chapter 4: A full quantitative version of the Fréchet-Kolmogorov Theorem (Theorem 4.3.2).

a) The relations between L^p-moduli and the singularity modulus and moduli of continuity (Lemma 4.2.4, Lemma 4.2.5, Lemma 4.2.12). The extraction of a bound of the L^p-norm from an L^p-modulus (Lemma 4.2.7).

b) The extraction of a bound of the size of a compact subset of L^p from a common L^p-modulus of its elements by refining the standard proof of the Fréchet-Kolmogorov Theorem (Lemma 4.3.9, Theorem 4.3.3).

c) The generalization of the lower bound on the size of sets of all functions that have a function as modulus given by Lorentz to arbitrary moduli (Proposition 3.1.4).

4. Chapter 5 and Section 6.1.4: The specification and investigation of well-behaved representations for L^p and Sobolev spaces (Definition 5.1.1, Definition 5.2.2).

   a) The proof that the representations of the L^p-spaces feature polynomial-time integration (Theorem 5.1.2), are computably equivalent to the standard representations of L^p as metric space (Theorem 5.1.3) and that the norm is exponential time computable (Theorem 5.1.8).

   b) The proofs that many of the continuous operations on Sobolev spaces are indeed polynomial-time computable (Proposition 5.2.8, Theorem 5.2.10, Theorem 5.2.11, Corollary 5.2.12).

   c) The proof that the operation of multiplying an L^p-function with a linear function is not polynomial-time computable but exactly as difficult as integrating a continuous function (Corollary 6.1.22), and that the same holds for L^1 with respect to the singular representation (Theorem 6.1.20). The proof that the pairing of L^p and L^q is at least as difficult as integrating a continuous function (Theorem 6.1.21).

5. Chapter 6: Establishing polynomial-time Weihrauch equivalence of the integration operator and the solution operator of the Dirichlet Problem for Poisson's Equation on the closed unit ball of arbitrary dimension (Theorem 6.3.1).

   a) The construction of polynomial-time Weihrauch reductions by uniformizing the proof of Friedman and Ko that the integration preserves polynomial time computability if and only if $\mathcal{FP} = \#\mathcal{P}$ (Theorem 6.1.17). The consequence that the uncurried version of the integration operator is polynomial time equivalent to the curried version (Corollary 6.1.19).

   b) The proof that solving the Dirichlet Problem for Laplace's Equation and Poisson's Equation can be reduced to integration (Theorem 6.3.2) by reevaluating classical solution formulas (Lemma 6.2.5, Proposition 6.2.7), truncating and transforming into spherical coordinates (All results from Proposition 6.3.4 to Proposition 6.3.8).

   c) The proof that integration can be reduced to solving the Dirichlet Problem for Poisson's Equation on a ball (Proposition 6.3.9).

Publications

During the course of writing, several papers containing some of the ideas presented in this thesis were published: [KSZ16b; PS16; KSZ16c; SSZ16; KSZ16a]. However, no essential parts written by others were transferred to this thesis without reformulation. Single sentences might have slipped through. We give a list of the abstracts of the papers together with clarifications about the overlaps of the contents.

[KSZ16b], abstract:

> The last years have seen an increasing interest in classifying (existence claims in) classical mathematical theorems according to their strength. We pursue this goal from the refined perspective of computational complexity. Specifically, we establish that rigorously solving the Dirichlet Problem for Poisson's Equation is in a precise sense 'complete' for the complexity class #P and thus as hard or easy as parametric Riemann integration (Friedman 1984; Ko 1991. Complexity Theory of Real Functions).

The findings of this paper are fully contained in Chapter 6, but the results have meanwhile been improved by the author to be uniform.

[KSZ16a], abstract:

> We promote the theory of computational complexity on metric spaces: as natural common generalization of (i) the classical discrete setting of integers, binary strings, graphs etc. as well as of (ii) the bit-complexity theory on real numbers and functions according to Friedman, Ko (1982ff), Cook, Braverman et al.; as (iii) resource-bounded refinement of the theories of computability on, and representations of, continuous universes by Pour-El&Richards (1989) and Weihrauch (1993ff); and as (iv) computational perspective on quantitative concepts from classical Analysis: Our main results relate (i.e. upper and lower bound) Kolmogorov's entropy of a compact metric space X polynomially to the uniform relativized complexity of approximating various families of continuous functions on X. The upper bounds are attained by carefully crafted oracles and bit-cost analyses of algorithms perusing them. They all employ the same representation (i.e. encoding, as infinite binary sequences, of the elements) of such spaces, which thus may be of own interest. The lower bounds adapt adversary arguments from unitcost Information-Based Complexity to the bit model. They extend to, and indicate perhaps surprising limitations even of, encodings via binary string functions (rather than sequences) as introduced by Kawamura&Cook (SToC'2010,§3.4). These insights offer some guidance towards suitable notions of complexity for higher types.

This paper has overlap with Chapter 3. However, the language used to describe the results is very different: The publication attempts to avoid the technical complications of second-order complexity theory as far as possible and cuts back on generality for this purpose. Conversely not everything discussed in the paper made it into the thesis. For instance the detour into higher-order complexity theory, the search for a notion of degree of a second-order polynomial and the endeavor to find a good definition of polynomial-time admissibility.

[KSZ16c], abstract:

> Pour-El and Richards [PER89], Weihrauch [Wei00], and others have extended Recursive Analysis from real numbers and continuous functions to rather general topological spaces. This has enabled and spurred a series of rigorous investigations on the computability of partial differential equations in appropriate advanced spaces of functions. In order to quantitatively refine such qualitative results with respect to computational efficiency we devise, explore, and compare natural encodings (representations) of compact metric spaces: both as infinite binary sequences (TTE) and more generally as families of Boolean functions via oracle access as introduced by Kawamura and Cook ([KC10], Sect. 3.4). Our guide is relativization: Permitting arbitrary oracles on continuous universes reduces computability to topology and computational complexity to metric entropy in the sense of Kolmogorov. This yields a criterion and generic construction of optimal representations in particular of (subsets of) Lp and Sobolev spaces that solutions of partial differential equations naturally live in.

This paper has overlap with both Chapter 3 and Chapter 5. Like in [KSZ16a], second-order complexity theory is avoided in [KSZ16c] as far as possible. No representation of length bigger than one is mentioned. Thus, the results presented in this thesis are more general.

[PS16], abstract:

> This paper considers several representations of the analytic functions on the unit disk and their mutual translations. All translations that are not already computable are shown to be Weihrauch equivalent to closed choice on the natural numbers. Subsequently some similar considerations are carried out for representations of polynomials. In this case in addition to closed choice the Weihrauch degree LPO^* shows up as the difficulty of finding the degree or the zeros.

This paper is purely computability theoretical and was therefore not included in this thesis. There is some overlap in the introduction of the Weihrauch reductions. The main achievements of the paper are briefly mentioned as an example in Section 6.1.1 (Theorem 6.1.3).

[SSZ16], abstract:

> We introduce, and initiate the study of, average-case bit-complexity theory over the reals: Like in the discrete case a first, naïve notion of

polynomial average runtime turns out to lack robustness and is thus refined. Standard examples of explicit continuous functions with increasingly high worst-case complexity are shown to be in fact easy in the mean; while a further example is constructed with both worst and average complexity exponential: for topological/metric reasons, i.e., oracles do not help. The notions are then generalized from the reals to represented spaces; and, in the real case, related to randomized computation.

The only overlap in the content of this paper and the thesis is Example 4.2.10 that turns up in both in slightly different forms.

1.1.3 Basic notational conventions

This chapter lists basic notations that are used throughout the thesis.

Fix the finite alphabet $\Sigma := \{\mathtt{0}, \mathtt{1}\}$. The following subsets of the set Σ^* of finite strings of zeros and ones are of relevance:

$\mathbb{N} := \{\varepsilon, \mathtt{1}, \mathtt{10}, \mathtt{11}, \ldots\}$ the set of non-negative **integers in binary** representation, where ε denotes the empty string and is interpreted as zero.

$\omega := \{\varepsilon, \mathtt{1}, \mathtt{11}, \mathtt{111}, \ldots\}$ the set of non-negative **integers in unary** representation.

We denote elements of Σ^* by $\mathbf{a}, \mathbf{b}, \ldots$ and elements of $\mathbb{N}$ and ω by $n, m, \ldots$. If this leads to ambiguity we use $\mathtt{1}^n$ with $n \in \mathbb{N}$ for the elements of ω. Let $|\cdot| : \Sigma^* \to \omega$ denote the **length function** replacing all 0s by 1s. To compute on $\mathbb{N}$ and ω we use the following **encodings** (i.e. **notations** in the sense of Weihrauch [Wei00]): For $\mathbb{N}$ the function $\nu_{\mathbb{N}} : \Sigma^* \to \mathbb{N}$ that eliminates leading zeros. For ω the function $\nu_\omega(\mathbf{a}) := |\nu_{\mathbb{N}}(\mathbf{a})|$.

Computations on products are handled via **tuple functions**. In Section 3.2 we need the tuple functions to have some very specific properties. Up until then any standard tuple function (i.e. some bijective, polynomial-time computable function with polynomial-time computable projections) may be used. For any given dimension d define an injection $(\Sigma^*)^d \to \Sigma^*$ as follows: Given strings $\mathbf{a}_1, \ldots, \mathbf{a}_d$ denote by $\mathbf{c}_i = c_{i,1} \ldots c_{i,\max\{|\mathbf{a}_i|\}+1}$ the padding of $\mathbf{a}_i$ to length $\max\{|\mathbf{a}_i| \mid i = 1, \ldots, d\} + 1$ by appending a 1 and then an appropriate number of 0s. Set

$$\langle \mathbf{a}_1, \ldots, \mathbf{a}_d \rangle := c_{1,1} \ldots c_{d,1} c_{1,2} \ldots c_{d,2} \ldots \ldots c_{1,\max\{|\mathbf{a}_i|\}+1} \cdots c_{d,\max\{|\mathbf{a}_i|\}+1}.$$

These tuple functions are not surjective thus we have to take extra care in some definitions. However, for a fixed d each of the tuple functions and its projections

$$\pi_i(\mathbf{b}) := \begin{cases} \mathtt{0}a_i & \text{if } \mathbf{b} = \langle \mathbf{a}_1, \ldots, \mathbf{a}_d \rangle \\ \varepsilon & \text{if } \mathbf{b} \notin \operatorname{img}(\langle \cdot, \ldots, \cdot \rangle) \end{cases} \tag{π}$$

are computable in linear time. The encoding of finite sequences of arbitrary length as $\langle d, \langle \mathbf{a}_1, \ldots, \mathbf{a}_d \rangle \rangle$ is explicitly written out when used to avoid confusion. The corresponding **pairing function for string functions** is defined as follows:

$$\langle \cdot, \cdot \rangle : \Sigma^{*\Sigma^*} \times \Sigma^{*\Sigma^*}, \quad \langle \varphi, \psi \rangle(\mathbf{a}) := \langle \varphi(\mathbf{a}), \psi(\mathbf{a}) \rangle.$$

The paring function for string functions is injective (and bijective if a bijective pairing function is used). For functions $l, k : \omega \to \omega$ we write $l = O(k)$ or $l \in O(k)$ if there exists some $C \in \omega$ such that for all n it holds that $l(n) \leq Ck(n) + C$.

The set of **real numbers** is denoted by $\mathbb{R}$. For $x \in \mathbb{R}$ let $\lfloor x \rfloor$ resp. $\lceil x \rceil$ denote the largest integer smaller or equal resp. the least integer larger or equal to x. The following subsets of the real numbers are of importance to this work:

$\mathbb{Z}$ the set of **integers**.

$\mathbb{D}$ the set of numbers that can be written as $\frac{r}{2^n}$ with $r \in \mathbb{Z}$ and $n \in \mathbb{N}$ called **dyadic numbers**.

$[0, 1]$ the unit interval.

The sets of integers and dyadic numbers are countable and can be handled by discrete computability and complexity theory via encodings. For the set $\mathbb{Z}$ we use the encoding defined on strings starting on 0 by $\nu_{\mathbb{Z}}(\mathtt{0}\mathbf{a}) := \nu_{\mathbb{N}}(\mathbf{a})$, on strings starting with $\mathtt{1}$ by $\nu_{\mathbb{Z}}(\mathtt{1}\mathbf{b}) := -\nu_{\mathbb{N}}(\mathbf{b})$ and on the empty string by 0. Due to the frequent use of the encoding of dyadic numbers $\mathbb{D}$ we choose special notation and set $[\![\langle \mathbf{a}, \mathbf{b} \rangle]\!] := \frac{\nu_{\mathbb{Z}}(\mathbf{a})}{2^{\nu_\omega(\mathbf{b})}}$. **Dyadic intervals** are encoded as pairs of dyadic numbers. We often pad strings. This is done using the following property of the encodings that follows straightforwardly from the definitions:

Lemma 1.1.1 (Padding). *Whenever $\{\mathbf{a}_1, \ldots, \mathbf{a}_N\}$ is a finite set of strings and $n \in \omega$ then there exists a set $\{\mathbf{b}_1, \ldots, \mathbf{b}_N\}$ of strings of that all have the same length, such that $|\mathbf{b}_i| \geq \max\{|\mathbf{a}_1|, \ldots, |\mathbf{a}_N|, n\}$ and $[\![\mathbf{a}_i]\!] = [\![\mathbf{b}_i]\!]$.*

Similar results hold for the other encodings.

For $d \in \mathbb{N}$ define an encoding of $\mathbb{D}^d$ by $[\![\langle \mathbf{a}_1, \ldots, \mathbf{a}_d \rangle]\!]_d := ([\![\mathbf{a}_1]\!], \ldots, [\![\mathbf{a}_d]\!])$. Since the dimension d is usually fixed, it is often omitted. **Dyadic boxes**, i.e. boxes whose corners have dyadic coordinates and whose edges are parallel to the axes, are denoted as

$$[\mathbf{a}, \mathbf{b}] := [[\![\mathbf{a}]\!]_d, [\![\mathbf{b}]\!]_d] = \{x \in \mathbb{R}^d \mid [\![\mathbf{a}]\!]_d \leq x \leq [\![\mathbf{b}]\!]_d\},$$

where the inequalities have to be understood component-wise.

For some $\Omega \subseteq \mathbb{R}^d$ denote the set of continuous functions from Ω to the real numbers by $\mathcal{C}(\Omega)$. We also handle multivalued functions: That is functions denoted as $f : X \rightrightarrows Y$ where the outcome is not a single value but a set of values $f(x) \subseteq Y$, interpreted as the acceptable return values. The domain of a multivalued functions is the set of elements such that the set of acceptable return values is not empty. If this set coincides with X, the function is called total, if it does not we indicate this by $f :\subseteq X \rightrightarrows Y$ or $f :\subseteq X \to Y$ if f is singlevalued. For $K \subseteq X$ denote the image of K under f by $f(K) := \bigcup_{x \in K} f(x)$. f is called surjective if $f(X) = Y$. Note that in contrast to functions the inverse multivalued partial function f^{-1} does always exist and that $f^{-1}(y)$ is the usual pre-image of the element y.

1.2 The model of computation

1.2.1 Representations

Encodings allow computations on countable structures using discrete computability theory. Most metric spaces appearing in practice, however, are not countable. For instance the real numbers, or, to mention a compact one, the unit interval are uncountable. Computable analysis removes the necessity of countability by encoding elements by infinite objects (infinite binary strings or string functions) instead of strings.

In this thesis we mean by the **Baire space** the space of all string functions $\mathcal{B} := (\Sigma^*)^{\Sigma^*}$ equipped with the product topology.

Definition 1.2.1. A **representation** of a set X is a partial surjective mapping $\xi :\subseteq \mathcal{B} \to X$. The elements of $\xi^{-1}(x)$ are called the **names** (or ξ-names) of x.

A space with a fixed representation is called a **represented space**. We denote represented spaces by $\mathbf{X}$, $\mathbf{Y}, \ldots$, their underlying sets by X, $Y, \ldots$ and their representations by $\xi_\mathbf{X}$, $\xi_\mathbf{Y}, \ldots$ Like the topology of a topological space the representation of a represented space is only mentioned if necessary to avoid ambiguities. An element of a represented space is called **computable** if it has a computable name. It is said to lie within a complexity class if it has a name from that complexity class.

On one hand, any represented space carries a natural topology: The final topology of the representation. On the other hand, one often looks for a representation suitable for a topological space. It is reasonable to require such a representation to induce the topology the space is equipped with. For this, continuity is necessary but not sufficient. Continuity together with openness is sufficient but not necessary.

The following is considered the standard representation of a separable metric space with distinguished dense sequence.

Definition 1.2.2. Let $\mathcal{M} := (M, d, (x_m)_{m\in\mathbb{N}})$ be a triple such that (M, d) is a complete separable metric space and $(x_m)_{m\in\mathbb{N}}$ is a dense sequence in M. Define the **Cauchy representation** $\xi_\mathcal{M}$ of M with respect to (x_m): A string function $\varphi \in \mathcal{B}$ is a $\xi_\mathcal{M}$-name of $x \in M$ if and only if

$$\forall n \in \mathbb{N} : d(x, x_{\varphi(n)}) < 2^{-n}.$$

Any standard representation is continuous and open with respect to the topology induced by the metric.

Recall the pairing function $\langle\cdot,\cdot\rangle : \mathcal{B}\times\mathcal{B} \to \mathcal{B}$ on string functions from the introduction.

Definition 1.2.3. Let $\mathbf{X}$ and $\mathbf{Y}$ be represented spaces. Define their **product** $\mathbf{X} \times \mathbf{Y}$ by equipping $X \times Y$ with the representation $\xi_{\mathbf{X}\times\mathbf{Y}}$ defined by

$$\xi_{\mathbf{X}\times\mathbf{Y}}(\varphi) = (x, y) \Leftrightarrow \exists\psi, \psi' \in \mathcal{B} : x = \xi_\mathbf{X}(\psi) \wedge y = \xi_\mathbf{Y}(\psi') \wedge \langle\psi, \psi'\rangle = \varphi.$$

That is: a string function is a name of the pair (x, y) if and only if it is the pairing of a name of x and a name of y.

Throughout the thesis, this construction is used self-evidently it.

1.2.2 Second order complexity theory

Computations of functions between represented spaces are carried out by operating on names and computing functions on Baire space:

Definition 1.2.4. Let $f : \mathbf{X} \rightrightarrows \mathbf{Y}$ be a (multivalued) function between represented spaces. A partial function $F :\subseteq \mathcal{B} \to \mathcal{B}$ is called a **realizer** of f, if

$$\varphi \in \xi_{\mathbf{X}}^{-1}(x) \quad \Rightarrow \quad F(\varphi) \in \xi_{\mathbf{Y}}^{-1}(f(x)).$$

That is: F translates names of x into names of acceptable return values, or of $f(x)$. No assumptions about the behavior on elements that are not names are made. F being a realizer of a function f can be visualized by the diagram in Figure 1.1. However, the domain of F is allowed to be bigger than that of ξ_X. Therefore, F being a realizer of f does not translate to the diagram being commutative in a rigorous way.

Recall that on the Baire space there exists a well-established computability theory originating from [Kle52], see [Lon05] for an overview. A functional $F :\subseteq \mathcal{B} \to \mathcal{B}$ is called computable if there is an oracle Turing machine $M^?$ such that $M^\varphi(\mathbf{a}) = F(\varphi)(\mathbf{a})$ for all strings $\mathbf{a}$ and string functions φ from the domain of F. Or spelled out: The computation of $M^?$ with oracle φ and on input $\mathbf{a}$ halts with the string $F(\varphi)(\mathbf{a})$ written on the output tape. A function between represented spaces is called **computable** if it has a computable realizer.

$$\begin{tikzcd}
\mathcal{B} \arrow[r, "F"] \arrow[d, "\xi_{\mathbf{X}}"'] & \mathcal{B} \arrow[d, "\xi_{\mathbf{Y}}"] \\
X \arrow[r, "f"'] & Y,
\end{tikzcd}$$

Figure 1.1: A diagram

Complexity theory for functionals is called **second-order complexity theory**. It was originally introduced by Mehlhorn [Meh76]. This thesis uses as definition a characterization via resource bounded oracle Turing machines due to Kapron and Cook [KC96]. Such a machine is granted time depending on the size of the input. The string functions are considered part of the input.

Definition 1.2.5. The **size** or **length** $|\varphi| : \omega \to \omega$ of a string function $\varphi \in \mathcal{B}$ is defined by

$$|\varphi|(n) := \max\{|\varphi(m)| \mid |m| \leq n\}.$$

For instance: Each polynomial-time computable string function is of polynomial size. A polynomial as function $\mathbb{N} \to \mathbb{N}$ has linear length where the slope is the degree of the polynomial. Polynomials $\omega \to \omega$ have themselves as length. Also compare Figure 1.2.

A running time is a mapping that assigns to sizes of the inputs an allowed number of steps a machine is allowed to take. Thus it should be of type $\omega^\omega \times \omega \to \omega$. We use the following conventions to measure the time of oracle interactions in a run of an oracle Turing machine: Writing the query takes time. Asking a query takes one time step. Reading the answer tape takes time. That is: writing the answer to an oracle query to the answer tape is part of the oracles job and does not take time. It may very well happen that the content of the answer tape is not accessible to the machine as a whole due to running time restrictions.

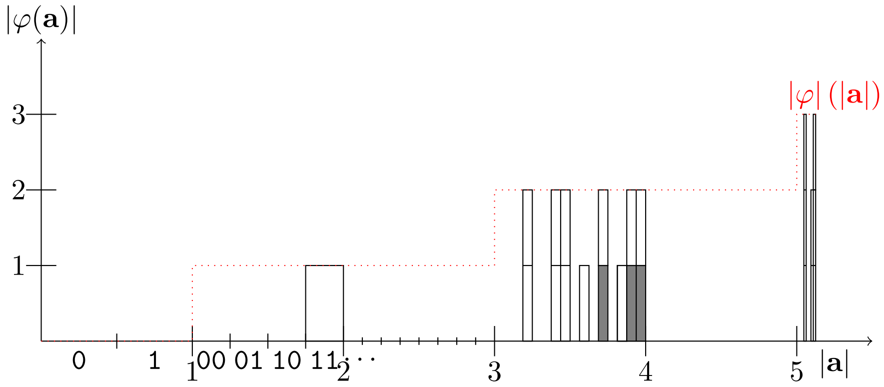


Figure 1.2: A graphical depiction of the first projection of the binary pairing from the introduction and its length.

Definition 1.2.6. An oracle Turing machine $M^?$ is said to **run in time** $T : \omega^\omega \times \omega \to \omega$ **on a set** $A \subseteq \mathcal{B}$ if the computation of $M^\varphi(\mathbf{a})$ on input $\mathbf{a} \in \Sigma^*$ with oracle $\varphi \in A$ terminates within $T(|\varphi|, |\mathbf{a}|)$ steps. It is said to **run in time** $O(T)$ **on** A if there is a $C \in \omega$ such that it runs in time $(l,n) \mapsto CT(l,n) + C$ on A.

If A is all of $\mathcal{B}$ we simply say that $M^?$ runs in time T resp. $O(T)$.

The subclass of running times that are considered polynomial, namely second-order polynomials, are recursively defined as follows:

- $(l,n) \mapsto p(n)$ for p a positive integer polynomial is a second-order polynomial.
- If P and Q are second-order polynomials then $P + Q$ and $P \cdot Q$ are second-order polynomials.
- If P is a second-order polynomial then $(l,n) \mapsto l(P(l,n))$ is a second-order polynomial.

Second-order polynomials turned up in other contexts before they were used as running times of oracle Turing machines (compare for instance [Koh96]). An example for a second-order polynomial is the mapping

$$(l,n) \mapsto l(l(n^2+5) + l(l(n)^2)).$$

Definition 1.2.7. Let $F :\subseteq \mathcal{B} \to \mathcal{B}$ be a functional and $T : \omega^\omega \times \omega \to \omega$ a running time. F is said to be **computable in time** T, if there is an oracle Turing machine $M^?$ computing F that runs in time T. F is said to be **computable in time** $O(T)$ if there exists a machine $M^?$ that runs in time $O(T)$. It is said to be **polynomial-time** computable if it runs in time P for some second-order polynomial P.

Example 1.2.8 (Projections). Consider the projections $\pi_i :\subseteq \mathcal{B} \to \mathcal{B}$ of the Baire space pairing function $\langle \cdot, \cdot \rangle : \mathcal{B} \times \mathcal{B} \to \mathcal{B}$ from the introduction. Note that we chose a pairing function that is not surjective. The projections are only defined on the image of the pairing function and therefore not total. Without loss of generality only consider the first projection. Recall that the pairing of string functions was defined by $\langle\varphi,\psi\rangle(\mathbf{a}) := \langle\varphi(\mathbf{a}),\psi(\mathbf{a})\rangle$ and the pairing of two strings was an intertwining of the strings padded to the same length by appending a `1` and then an appropriate number of `0`s. Let $M^?$ be the machine that copies its input $\mathbf{a}$ to the query tape, shifts the head on the answer tape to the very end of the return value, shifts forward until it finds a `1` written in an odd position and then copies all the odd positions in front of this symbol to the answer tape. This machine computes a total function on the Baire space and whenever the oracle is of the form $\langle\varphi,\psi\rangle$ it computes φ. $M^?$ runs in time $O(P)$ for $P(l,n) = l(n)$. Thus π_i is polynomial-time computable.

Definition 1.2.9. A function between represented spaces is called **polynomial-time computable** if it has a polynomial-time computable realizer (compare Definition 1.2.4).

An important special case is the following:

Definition 1.2.10. Let ξ and ξ' be representations of a space X. ξ **polynomial-time reduces** to ξ' if the identity from (X,ξ') to (X,ξ) is polynomial-time computable. The representations are **polynomial-time equivalent** if reductions in both directions exist.

If the identity is merely computable resp. has a continuous realizer, one speaks of computable resp. continuous reduction and equivalence. We chose a convention here: ξ is reducible to ξ' if there is a translation of ξ'-names into ξ-names. This follows the intuition that a ξ is reducible to ξ' if its names contain less information about the encoded element. Other authors want to call the translations reductions and choose to define reducibility the other way around to avoid confusion with English language (cf. [Wei00]).

An important concept from computable analysis is **admissibility** which for all representations we are concerned with is the same as continuous equivalence to standard representations [Wei00] (as the one for metric spaces from Definition 1.2.2). It implies that the topology of the represented space matches the final topology of the representation and therefore yields the continuity of the representation. It does not imply openness, for the connections between admissibility and openness see [BH02].

1.2.3 Second-order representations

The length

$$|\varphi|(\mathtt{1}^n) = \max\{|\varphi(m)| \mid |m| \leq \mathtt{1}^n\}$$

of a string function cannot be computed from an oracle of the string function in polynomial-time: To find the maximum in the definition the oracle φ has to be queried an exponential number of times. For many applications, polynomial-time computability of the length of names is desirable.

Definition 1.2.11. A string function φ is called **length-monotone** if for all strings $\mathbf{a}$ and $\mathbf{b}$

$$|\mathbf{a}| \leq |\mathbf{b}| \quad \Rightarrow \quad |\varphi(\mathbf{a})| \leq |\varphi(\mathbf{b})| \,.$$

The set of length-monotone string functions is denoted by Σ^{**}.

For a length-monotone string function $|\varphi|\,(|\mathbf{a}|) = |\varphi(\mathbf{a})|$ the length function restricted to Σ^{**} is polynomial-time computable. The function from Figure 1.2 is an example of a function that is not length-monotone. For a length monotone function the whole area below the dotted line would be filled.

Definition 1.2.12. A **second-order representation** is a representation whose domain is contained in Σ^{**}.

Equivalently: A second-order representation ξ of a space X is a partial surjective mapping $\xi : \Sigma^{**} \to X$ from the length-monotone string functions to the space. The prefix 'second-order' is for applicability of second-order complexity theory, and does not indicate the use of objects of higher order than for regular representations.

It is often the case that the restriction of a representation to the length-monotone functions is still surjective and thus a second-order representation. All representations this thesis is concerned with are second-order representations. For brevity 'second-order' is sometimes omitted.

Recall from the introduction that we fixed an encoding $[\![\cdot]\!]$ of the dyadic numbers.

Example 1.2.13 (Standard representation of reals). Make $\mathbb{R}$ a represented space by equipping it with the second-order representation $\xi_\mathbb{R}$ defined by

$$\xi_\mathbb{R}(\varphi) = x \quad \Leftrightarrow \quad \varphi \in \Sigma^{**} \wedge \forall 1^n \in \omega : \left|\frac{\varphi(1^n)}{2^n} - x\right| < 2^{-n} \wedge |\varphi|\,(n) \leq |\varphi|\,(0) + 4n.$$

The last condition is only necessary to avoid iteration of the length function in some cases. It may be left away if one is only interested in polynomial time computability. The use of unary coding for the error is more natural than it may seem at first: A polynomial-time equivalent representation can be defined by using codings of rational or dyadic numbers and requiring $|[\![\varphi(\mathbf{a})]\!] - x| < [\![\mathbf{a}]\!]$. Furthermore, this convention roughly means that for a polynomial-time computable real number the time of computation grows polynomial in the number of valid digits of the binary encoding that are produced. We equip subsets of the real numbers, in particular $[0, 1]$, with the range restriction of $\xi_\mathbb{R}$.

The second-order representation $\xi_\mathbb{R}$ induces the established complexity classes of real numbers. It is computably equivalent to the standard representation of the reals as metric space from Definition 1.2.2 if the standard enumeration of the dyadic numbers is chosen as dense sequence. Polynomial-time equivalence fails since the input is encoded in unary, not in binary. A proof that Definition 1.2.7 induces the usual notions of computability and polynomial-time computability of real functions (for the usual notions see [Ko91] or [Wei00]) can be found in [Lam06].

The pairing function on strings and the definition of the pairing of string functions from the introduction are carefully chosen such that the following holds:

Lemma 1.2.14. *The product of two second-order representations is a second-order representation.*

PROOF. Let $\mathbf{X}$ and $\mathbf{Y}$ be second-order represented spaces and let $\langle\varphi,\psi\rangle$ be a $\xi_{\mathbf{X}\times\mathbf{Y}}$-name of $(x,y)\in X\times Y$. By assumption φ and ψ are length-monotone functions. To see that also $\langle\varphi,\psi\rangle$ is length-monotone, note that the pairing function is increasing in both arguments, and therefore for strings $|\mathbf{a}|\leq|\mathbf{b}|$

$$|\langle\varphi,\psi\rangle(\mathbf{a})| = |\langle\varphi(\mathbf{a}),\psi(\mathbf{a})\rangle| \leq |\langle\varphi(\mathbf{b}),\psi(\mathbf{a})\rangle| \leq |\langle\varphi(\mathbf{b}),\psi(\mathbf{b})\rangle| = |\langle\varphi,\psi\rangle(\mathbf{b})|\,.$$

This proves the assertion. ∎

Example 1.2.15. The d-fold product representation of $\xi_{\mathbb{R}}$ from Example 1.2.13 turns $\mathbb{R}^d$, and with it all its subsets, into a second-order represented space.

All the encodings from the introduction assign codes of all sizes bigger than some minimal size to an element (for the dyadic numbers this is stated formally in Lemma 1.1.1). Since only representations whose names return codes from one of these are considered, an arbitrary name can always be padded to a length monotone one. Therefore, all representations introduced restrict to Σ^{**} and are introduced as second-order representations right away.

To be able to talk about clocked machines it is necessary to restrict to those running-times that are time-constructible.

Definition 1.2.16. A function $T:\omega^\omega\times\omega\to\omega$ is called **time-constructible**, if there is an oracle Turing machine $M^?$ that runs in time $O(T)$ and on input $\mathbf{a}$ with oracle $\varphi\in\Sigma^{**}$ halts with $T(|\varphi|,|\mathbf{a}|)$ written to the output tape.

Note, that we only require the machine to be correct on length-monotone string functions. In particular this renders all second-order polynomials time-constructible, which would not be the case if we replaced Σ^{**} by $\mathcal{B}$ in the above definition.

If $F:\subseteq\Sigma^{**}\to\mathcal{B}$ is computed by a machine that runs in time T on $\mathrm{dom}(F)$ and T is time-constructible, then this function is computable in time $O(T)$: Just replace the machine with a clocked machine that still computes F and runs in time $O(T)$.

1.2.4 Representations of compact spaces and length

Recall that a set is called relatively compact if its closure is compact. The compact subsets of the Baire space are easily classified. One possible classification is that a subset of the Baire space is relatively compact if and only if it is contained in a set from the following family:

Definition 1.2.17. Define a family $(K_l)_{l\in\omega^\omega}$ of compact subsets of $\mathcal{B}$ by

$$K_l := \{\varphi\in\mathcal{B}\mid|\varphi|\leq l\}.$$

The mentioned property of this family of sets resembles hemi-compactness: Whenever K is a compact subset of Baire space, there is some $l \in \omega^\omega$ such that $K \subseteq K_l$. The difference to hemi-compactness is that the index set ω^ω is not countable but has the cardinality of the continuum.

Definition 1.2.18. A function $l : \omega \to \omega$ is called a **length** of a representation ξ of a space X if

$$\xi(K_l) = X.$$

I.e. l is a length of ξ if every element has a ξ-name of length l. This does not imply that the domain of ξ is included in K_l, but is strictly weaker.

Example 1.2.19. Let Ω be a bounded subset of $\mathbb{R}^d$. And let $C \in \omega$ be such that each element $x \in \Omega$ has supremum norm $|x|_\infty$ less than 2^C. Let ξ_Ω denote the range restriction of the d-fold product of the standard representation $\xi_\mathbb{R}$ from Example 1.2.13. Then the function $l(n) := 2d(n + C + 4)$ is a length of ξ_Ω.

Not every representation has a length, but representations of compact spaces usually do for the following reason:

Proposition 1.2.20. *An open representation of a compact space has a length.*

Proof. Let ξ be an open representation of a compact space K. Let $\mathbf{a}_m$ denote the standard enumeration $\mathbb{N} \to \Sigma^*$ that removes the first digit. Recursively define a function $l : \omega \to \omega$ as follows:

For $m \in \mathbb{N}$ set

$$U_{0,m} := \{\varphi \mid \varphi(\varepsilon) = \mathbf{a}_m\} \subseteq \mathcal{B}.$$

The sequence $(U_{0,m})_{m\in\mathbb{N}}$ forms an open cover of the Baire space. Since ξ is open, the images of these sets compose an open cover of K. Since K is compact, there exists a finite subset I of $\mathbb{N}$ such that $(\xi(U_{0,n}))_{n\in I}$ covers K. Define

$$l(0) := \max\{|n| \mid n \in I\}$$

Note that this means that each element of K has a name φ such that $|\varphi|(0) = |\varphi(\varepsilon)| \leq l(0)$.

Now assume that l has been defined up to value n such that each element of K has a name φ such that for each $k \leq n$ it holds that $|\varphi|(k) \leq l(k)$. To construct $l(n+1)$ let J be the set of all strings of length exactly $n+1$. Define a sequence $(U_{n+1,m})_{m\in\mathbb{N}^J}$ as follows:

$$U_{n+1,(m_\mathbf{b})_{\mathbf{b}\in J}} := \{\varphi \mid \forall k \leq n : |\varphi|(k) \leq l(k) \text{ and } \forall \mathbf{b} \in J : \varphi(\mathbf{b}) = m_\mathbf{b}\}.$$

Note that each of these sets is an open subset of Baire space and that the assumptions of the recursion make sure that the union of their images is K. Thus, again from the compactness of K it follows that there is a finite subset I of $\mathbb{N}^J$ such that $(\xi(U_{n+1,m}))_{m\in I}$ covers K. Set

$$l(n+1) := \max\{|m| \mid \exists (n_\mathbf{a})_{\mathbf{a}\in J} \in I, \exists \mathbf{b} \in J : m = n_\mathbf{b}\}.$$

By choice of I it is again the case that each element of K has a name φ such that for each $k \leq n+1$ it holds that $|\varphi|(k) \leq l(k)$.

The constructed function is a length of the representation by its definition. ■

2 Minimal representations

The first part of this chapter recollects some facts about the standard representation of continuous function on the unit interval. In particular this representation is the weakest representation to allow evaluation of a function. Computability-wise this statement is well known, however, we need the polynomial-time version that originates from [KC10].

The second part of this chapter introduces a representation of the space of all integrable functions on the unit interval with similar properties, where the evaluation operator is replaced with an integration operator. This representation is generalized to the spaces $\mathrm{L}^1(\Omega)$ for an arbitrary bounded measurable set $\Omega \subseteq \mathbb{R}^d$. This Thesis provides a proof that these representations are minimal with the property that integration is possible in polynomial time and that they are discontinuous with respect to the two most common topologies on $\mathrm{L}^1(\Omega)$.

2.1 Representing continuous functions

The standard representation of the continuous functions on the unit interval has been used for a long time, it is computably equivalent to the standard representation of $\mathcal{C}([0,1])$ as metric space and is for instance discussed in [PER89] and [Wei00]. Its second-order version was introduced and investigated by Kawamura and Cook in [KC10]. Here we recall the definition and the properties that are relevant for this thesis.

2.1.1 Moduli of continuity

A central notion for the standard representation is the modulus of continuity. For later use we introduce it a little more general than needed in this chapter. Let $\Omega \subseteq \mathbb{R}^d$ be a set and let $|\cdot|_\infty$ denote the supremum norm on real vectors.

Definition 2.1.1. A function $\mu : \omega \to \omega$ such that $\mu(n) \neq 0 \Rightarrow \mu(n+1) > \mu(n)$ (i.e. μ is strictly increasing whenever it is not zero) is called a **modulus of continuity** of $f \in C(\Omega)$ if

$$|x - y|_\infty \leq 2^{-\mu(n)} \quad \Rightarrow \quad |f(x) - f(y)| < 2^{-n}$$

holds for all $x, y \in \Omega$ and $n \in \omega$.

To be exact, this notion of a modulus should be called modulus of uniform continuity: A function has a modulus of continuity if and only if it is uniformly continuous. However, since non-uniform moduli of continuity are not important for our purposes, we decide to omit the 'uniform'. In particular continuous functions on compact sets have moduli of continuity.

Example 2.1.2. For $\Omega \subseteq \mathbb{R}^d$ let $f : \Omega \to \mathbb{R}$ be a function such that its image $\mathrm{img}(f)$ is bounded. (This is for example the case whenever Ω is compact and f is continuous.) If f is Lipschitz continuous, it has a modulus of continuity of the form $\mu(n) = n + b$. Hölder continuity corresponds to linear moduli of continuity $\mu(n) = an + b$. More precisely if f satisfies

$$|f(x) - f(y)| \leq C\,|x - y|_\infty^\alpha\,,$$

then $\mu(n) := \frac{n+\lceil \mathrm{lb}(C)\rceil}{\alpha}$ is a modulus of continuity. Conversely, for $\mu(n) = an + b$ a modulus of continuity of f and M a bound on the diameter of $\mathrm{img}(f)$, then

$$\alpha := \frac{1}{a} \quad \text{and} \quad C := \max\{2^{\alpha b}M, 2^{\alpha b+1}\}$$

are as needed. This can be verified by considering the cases $2^{-\mu(n+1)} \leq |x-y|_\infty < 2^{-\mu(n)}$ for some n and $2^{-\mu(0)} \leq |x-y|_\infty$ separately.

Definition 2.1.1 differs from common definitions of moduli of continuity (compare for instance [Wei00]) in two ways: First off, the modulus is usually not required to fulfill any monotonicity condition. This, however, is not really a restriction as long as one is only interested in asymptotic growth: The modulus can be made monotone without changing the growth unless the function is constant. Secondly, the difference of the function values is usually not required to be strictly smaller than 2^{-n}. This thesis chooses this convention to make the standard representation in the upcoming Definition 2.1.6 an open map.

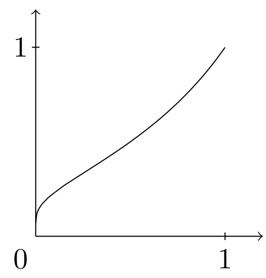


Figure 2.1: The function from Example 2.1.3

Example 2.1.3 (No polynomial modulus). Consider the function $f : [0,1] \to [0,1]$ defined by

$$f(x) = \begin{cases} 0 & \text{if } x = 0 \\ \frac{1}{1-\mathrm{lb}(x)} & \text{if } x \neq 0. \end{cases}$$

(compare Figure 2.1). Since $\left|f(0) - f\left(2^{1-2^n}\right)\right| = 2^{-n}$, this function does not have any modulus of continuity smaller than $n \mapsto 2^n - 1$. In particular it does not have any polynomial modulus of continuity. It is not difficult to construct for each $\nu : \omega \to \omega$ a continuous function that only has moduli of continuity that are pointwise bigger than ν.

2.1.2 Representing the continuous functions

Recall that we fixed a representation $\xi_\mathbb{R}$ of the real numbers and decided to use its range restriction on the unit interval $[0,1]$. Furthermore, recall that computability and polynomial-time computability of functions from the unit interval to the real numbers was defined via realizers. A proof that these notions coincide with the notions used in [PER89], [Ko91] and [Wei00] can be found in [Lam06]. The following result is the reason for the importance of moduli of continuity in real complexity theory and the starting point of many further investigations and generalizations:

Theorem 2.1.4. *A function $f : [0,1] \to \mathbb{R}$ is polynomial-time computable if and only if both of the following are fulfilled:*

- *There is a polynomial-time computable function $\psi : \mathbb{D} \times \omega \to \mathbb{D}$ such that for any $r \in (0,1) \cap \mathbb{D}$ and $n \in \omega$*
$$|\psi(r,n) - f(r)| < 2^{-n}.$$
- *There is a polynomial modulus of continuity of f.*

The proof is straightforward and can be found in [Ko91]. Thus, the function from Example 2.1.3 is not polynomial-time computable: It does not allow a polynomial modulus of continuity. The first bullet of the theorem describes how one would model real functions by arbitrary precision arithmetic with guaranteed output error. As example for how this Theorem leads to generalizations, note that moduli of continuity can be defined for functions between arbitrary metric spaces. The theorem can then be used to define polynomial-time computability between arbitrary compact metric spaces [LLM01].

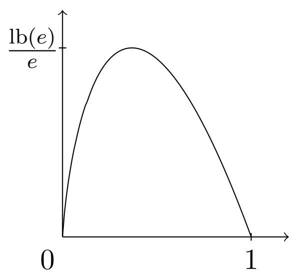


Figure 2.2: the function from Example 2.1.5

Example 2.1.5 (A polynomial-time computable function). The function from Figure 2.2, that is
$$f(x) = \begin{cases} 0 & \text{if } x = 0 \\ -x \cdot \mathrm{lb}(x) & \text{if } x \neq 0. \end{cases}$$
is polynomial-time computable: Since f is Hölder continuous it has a polynomial modulus of continuity (cf. Example 2.1.2). The logarithm of some non-zero dyadic number can be computed within time quadratic in the binary length of an encoding of that number. Together with the polynomial time algorithm for multiplication of dyadic numbers this leads to an algorithm for evaluating f on non-zero dyadic arguments.

Another application of Theorem 2.1.4 is to prove that the following representation induces the usual complexity classes:

Definition 2.1.6. Define the **standard second-order representation $\xi_{\mathcal{C}}$ of $\mathcal{C}([0,1])$** by letting a length-monotone string function $\varphi \in \Sigma^{**}$ be a $\xi_{\mathcal{C}}$-name of f if and only if for any $\mathbf{a}$ with $[\![\mathbf{a}]\!] \in (0,1) \cap \mathbb{D}$ and $1^n \in \omega$
$$|\,[\![\varphi(\langle \mathbf{a}, 1^n \rangle)]\!] - f([\![\mathbf{a}]\!])| < 2^{-n}$$
and $|\varphi|$ is a modulus of continuity of f.

To see that this is a second-order representation, note that a length-monotone name of an element can be found by first choosing a sequence of dyadic approximations and then padding the encodings of these approximations appropriately (cf. Lemma 1.1.1).

The computable Weierstraß Approximation Theorem, for instance proven in [PER89], shows that this representation is computably equivalent to the standard representation of $\mathcal{C}([0,1])$ as metric space, if a standard enumeration of the rational polynomials is chosen as dense sequence.

2.1.3 Some complexity theoretical results

The standard representation has been characterized as the weakest representation, up to polynomial-time equivalence, such that the evaluation operator defined by

$$\mathrm{eval} : C([0,1]) \times [0,1] \to \mathbb{R}, \quad (f,x) \mapsto f(x)$$

is polynomial-time computable. Recall that $\mathbb{R}$ and $[0,1]$ are represented spaces equipped with the standard representation $\xi_\mathbb{R}$ of the real numbers from Example 1.2.13 and its range restriction.

Theorem 2.1.7 (Minimality of $\xi_\mathcal{C}$). *For a second-order representation ξ of a subset F of $\mathcal{C}([0,1])$ the following are equivalent:*

- *The restriction of the evaluation operator is polynomial-time computable.*
- *The range restriction of $\xi_\mathcal{C}$ is polynomial time reducible to ξ.*

This is a slight generalization of a theorem from [KC10]. The proof given there generalizes straightforwardly. It is not reevaluated here, since the proof Theorem 2.2.12 is very similar.

The rest of this section recalls some hardness results for the integration of continuous functions in the standard representation and some of the background knowledge from complexity theory needed to understand these. The proofs can be found in [Ko91]. These results are important for motivational reasons and for the very end of the thesis, and may be skipped on first reading.

Recall that $\#\mathcal{P}$ is the class of functions $\varphi : \Sigma^* \to \mathbb{N}$ such that there exists a polynomial-time decidable set $V \subseteq \Sigma^*$, called the **verifier set**, and a polynomial p such that

$$\varphi(\mathbf{a}) \; = \; \# \left\{ \mathbf{b} \in \Sigma^{p(|\mathbf{a}|)} \middle| \langle \mathbf{a}, \mathbf{b} \rangle \in V \right\}.$$

Each $\mathcal{NP}$ problem asking for the existence of a witness verifiable within polynomial time, corresponds to a $\#\mathcal{P}$ function computing the number of witnesses. For the $\mathcal{NP}$-complete problem SAT, for example, the corresponding $\#\mathcal{P}$ function computes the number of satisfying assignments of a boolean formula. This function is called #SAT and is complete for $\#\mathcal{P}$. Furthermore, $\mathcal{FP} = \#\mathcal{P}$ implies $\mathcal{P} = \mathcal{NP}$. Toda's Theorem [Tod91] implies that the entire polynomial-time hierarchy lies below $\mathcal{P}^{\#\mathcal{P}}$, where $\mathcal{P}^{\#\mathcal{P}}$ is the class of decision problems solvable in polynomial time being allowed one query

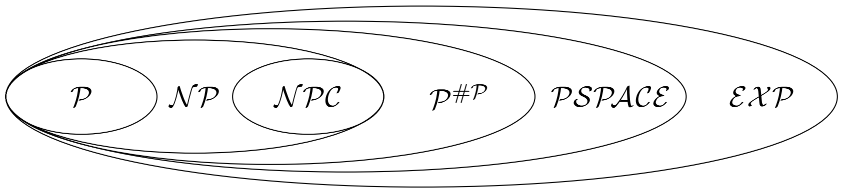


Figure 2.3: The inclusion relation between the complexity classes under common assumptions about strictness of inclusions. $\mathcal{NPC}$ are the $\mathcal{NP}$-complete problems.

to some $\varphi \in \#\mathcal{P}$. Together with $\mathcal{EXP}$ and $\mathcal{PSPACE}$, the classes of decision problems solvable in exponential time resp. polynomial space, one arrives at the usual picture of inclusions of complexity classes (cf. Figure 2.3). For none of the inclusions but the outermost $\mathcal{P} \subsetneq \mathcal{EXP}$ is it know whether they are strict or not. In particular it is not impossible, but considered highly unlikely that $\mathcal{P} = \mathcal{NPC}$.

The following theorem originates from [Fri84]. A similar connection exists between maximization and the millenium $\mathcal{P}$ vs. $\mathcal{NP}$ problem.

Theorem 2.1.8. *The following are equivalent:*

- *$\mathcal{FP} = \#\mathcal{P}$.*
- *The operator*

$$\mathrm{INT} : \mathcal{C}([0,1]) \to \mathcal{C}([0,1]), \quad \mathrm{INT}(f)(x) := \int_0^x f(t)\mathrm{d}t \qquad \text{(INT)}$$

preserves polynomial-time computability. I.e. if f is polynomial-time computable, then it has a polynomial-time computable antiderivative.

This is not a statement about polynomial-time computability of an operator on continuous functions. It is a pointwise statement: Even if $\mathcal{FP} = \#\mathcal{P}$ holds true there might not be a fast way to obtain a procedure to compute the antiderivative from a procedure to compute the function. Indeed the uniform statement is known to fail:

Theorem 2.1.9. *The integration operator from* (INT) *is not polynomial-time computable.*

A proof of this theorem can be constructed as follows: It is easy to see that this is equivalent to the operator

$$\mathrm{INT_u} : C([0,1]) \times [0,1]^2 \to \mathbb{R}, \quad (f, x, y) \mapsto \int_x^y f(t)dt$$

not being polynomial-time computable. If $\mathrm{INT_u}$ was polynomial-time computable also $f \mapsto \mathrm{INT_u}(f, 0, 1)$ would be polynomial-time computable. A proof that the last operator is not polynomial-time computable can be found in [Kaw+15]. These operators and their relationships are investigated in more detail in Section 2.2 and Chapter 6

2.2 Representing integrable functions

Fix some bounded measurable set $\Omega \subseteq \mathbb{R}^d$. Recall that two functions are said to be equal almost everywhere, if they only differ in a set of Lebesgue measure zero and that this defines an equivalence relation on the set of all functions. Furthermore recall that $\mathrm{L}^1(\Omega)$ denotes the set of equivalence classes up to equality almost everywhere of functions on Ω that are integrable with respect to the Lebesgue measure. The mapping $\|f\|_1 := \int_\Omega |f(t)|\,\mathrm{d}t$ defines a norm on $\mathrm{L}^1(\Omega)$. This section specifies the weakest representation of $\mathrm{L}^1(\Omega)$ that renders integration polynomial-time computable. More formally, the following operator is supposed to be polynomial-time computable:

$$\mathrm{INT_u} : \mathrm{L}^1(\Omega) \times \Omega^2 \to \mathbb{R}, \quad (f, x, y) \mapsto \int_{[x,y]\cap\Omega} f \mathrm{d}\lambda, \tag{INTu}$$

where $[x, y]$ denotes the smallest box with edges parallel to the axis and corners x and y. Here, $\mathbb{R}^d$ is equipped with the d-fold product of the standard representation of the real numbers and Ω with its range-restriction.

First consider the simplest case $\Omega = [0, 1]$: Note that the operator

$$\mathrm{INT} : \mathrm{L}^1([0, 1]) \to \mathcal{C}([0, 1]), \quad \mathrm{INT}(f)(x) := \int_0^x f(t)\mathrm{d}t.$$

is linear and continuous with $\|\mathrm{INT}\| = 1$. This operator translates the integration operator and the evaluation operator into each other:

$$\mathrm{eval}(\mathrm{INT}(f), x) = \mathrm{INT_u}(f, 0, x),$$
$$\mathrm{INT_u}(f, x, y) = \mathrm{eval}(\mathrm{INT}(f), y) - \mathrm{eval}(\mathrm{INT}(f), x).$$

The image of INT is the set $\mathcal{AC}_0([0, 1])$ of absolutely continuous functions that vanish in zero. The operator INT is injective and thereby invertible on its image.

From the above it follows that $\mathrm{INT}^{-1} \circ \xi_{\mathcal{C}}|^{\mathcal{AC}_0([0,1])}$ is a representation with the desired minimality property: Whenever ξ renders $\mathrm{INT_u}$ polynomial-time computable, $\mathrm{INT} \circ \xi$ renders evaluation polynomial-time computable. Thus, the polynomial-time reduction from $\xi_{\mathcal{C}}|^{\mathcal{AC}_0([0,1])}$ to $\mathrm{INT} \circ \xi$ that exists by the minimality of $\xi_{\mathcal{C}}$ from Theorem 2.1.7 is also a polynomial-time reduction from $\mathrm{INT}^{-1} \circ \xi_{\mathcal{C}}|^{\mathcal{AC}([0,1])}$ to $\mathrm{INT}^{-1} \circ \mathrm{INT} \circ \xi = \xi$.

Since INT^{-1} is discontinuous this representation is not continuous. Abstract arguments proving this can be found in [Sch02, Section 4.3]. This chapter specifies an alternative description of the above representation that allows for generalizations and gives a direct proof that the representation and its multidimensional generalizations are discontinuous.

2.2.1 Singularity moduli

With the notation from the introduction of this section: For $f \in \mathrm{L}^1([0, 1])$ a function μ is a modulus of continuity of $\mathrm{INT}(f)$ if and only if

$$|x - y| \leq 2^{-\mu(n)} \quad \Rightarrow \quad \left|\int_x^y f(t)\mathrm{d}t\right| < 2^{-n}.$$

This motivates the following definition. Let Ω be a measurable subset of $\mathbb{R}$ (Ω is no longer assumed to be bounded). For $f \in \mathrm{L}^1(\Omega)$ denote by $\tilde{f}$ the extension of f to all of the real line by zero.

Definition 2.2.1. A function $\mu : \omega \to \omega$ is called a **singularity modulus** of $f \in \mathrm{L}^1(\Omega)$, if for any $n \in \omega$ and $x, y \in \mathbb{R}$

$$|x - y| \leq 2^{-\mu(n)} \quad \Rightarrow \quad \left| \int_x^y \tilde{f}(t)\mathrm{d}t \right| < 2^{-n}.$$

Like any continuous function on the unit interval admits a modulus of continuity, any function from $\mathrm{L}^1(\Omega)$ possesses a singularity modulus. For $\Omega = [0, 1]$ any modulus of continuity of the function $\mathrm{INT}(f)$ from the introduction of this section may be chosen. It is possible to give a direct proof that also covers unbounded sets Ω:

Proposition 2.2.2 (Existence of moduli). *Any function from* $\mathrm{L}^1(\Omega)$ *has a singularity modulus.*

Proof. Note that $A \mapsto \int_A |f| \,\mathrm{d}\lambda$ is a measure and $x \mapsto \lambda(|f|^{-1}((x, \infty)))$ converges to zero as x tends to infinity. Thus there is an increasing sequence of positive integers m_n such that for all n

$$\int_{|f|^{-1}((2^{m_n},\infty))} |f| \,\mathrm{d}\lambda < 2^{-n-1}.$$

Note that for any $n \in \mathbb{N}$, $x \in \Omega$ and $0 \leq h \leq 2^{-m_n-n-1}$

$$\left| \int_x^{x+h} \tilde{f}(t)\mathrm{d}t \right| \leq \int_x^{x+2^{-m_n-n-1}} |\tilde{f}(t)|\mathrm{d}t \leq \int_{|f|^{-1}((2^{m_n},\infty))} |f| \,\mathrm{d}\lambda + 2^{-m_n-n-1}2^{m_n} < 2^{-n}.$$

Therefore, $\mu(n) := \max\{m_k + k + 1 \mid k \leq n\}$ defines a singularity modulus of f. ∎

If Ω is bounded, the existence of a singularity modulus implies integrability.

Remark 2.2.3. If the interior of Ω is unbounded the situation is more involved: On the one hand, there are non-integrable, but locally integrable functions that permit a singularity modulus. On the other hand some locally integrable functions have no singularity modulus. In the following, however, only bounded sets or functions with bounded support are considered.

It is not difficult to construct integrable functions with arbitrarily bad modulus of integrability:

Example 2.2.4 (Large moduli). To construct an integrable function that has no singularity modulus smaller than a given strictly increasing function $\nu : \omega \to \omega$ set

$$f := \sum_{i=0}^{\infty} 2^{\nu(i)-i} \chi_{(2^{-\nu(i)-1}, 2^{-\nu(i)})}.$$

Then $f \in \mathrm{L}^1([0,1])$ and

$$\int_0^{2^{-\nu(n)}} f(t)\mathrm{d}t = \sum_{i=n}^{\infty} 2^{\nu(i)-i}2^{-\nu(i)-1} = 2^{-n}.$$

Therefore, any modulus of integrability μ of f must fulfill $\mu(n) > \nu(n)$.

The next proposition uses L^p-spaces, that are recollected in the upcoming Section 4.2 in more detail. The case $p = \infty$, however, is understandable if one recalls that L^∞ are the essentially bounded functions and $\|\cdot\|_\infty$ the essential supremum norm.

Proposition 2.2.5 (Small moduli). *For a function $f \in \mathrm{L}^1([0,1])$ and an integer $C \in \omega$ the following hold:*

1. *if $n \mapsto n+C$ is a singularity modulus of f, then $f \in \mathrm{L}^\infty([0,1])$ and $\mathrm{lb}(\|f\|_\infty) \leq C$.*
2. *If $f \in \mathrm{L}^p([0,1])$ for some $1 < p \leq \infty$ and $C > \mathrm{lb}(\|f\|_p)$ and $D \geq \left(1 - \frac{1}{p}\right)^{-1}$ are integer constants, then $n \mapsto D(n+C)$ is a singularity modulus of f.*

The proof relies on the well-known Lebesgue Differentiation Theorem [Rud87].

Theorem 2.2.6 (Lebesgue Differentiation Theorem). *Let $f \in \mathrm{L}^1(\mathbb{R})$. Then for any representative g of f and almost all $x \in \mathbb{R}$ it holds that*

$$g(x) = \lim_{m\to\infty} 2^m \int_{x-2^{-m-1}}^{x+2^{-m-1}} g\mathrm{d}\lambda.$$

Proof (of Proposition 2.2.5). First prove item 1. For this assume that $n \mapsto n+C$ is a singularity modulus of f and let g be a representative of the function considered. By the Lebesgue Differentiation Theorem 2.2.6 there exists a set $A \subseteq [0,1]$ of measure one such that for any $x \in A$

$$|g(x)| = \lim_{m\to\infty} 2^m \left| \int_{x-2^{-m-1}}^{x+2^{-m-1}} g\mathrm{d}\lambda \right| = \lim_{n\to\infty} 2^{n+C} \left| \int_{x-2^{-n+C-1}}^{x+2^{-n+C-1}} f\mathrm{d}\lambda \right| < 2^C.$$

This proves that $\|f\|_\infty \leq 2^C$ and in particular that $f \in \mathrm{L}^\infty([0,1])$.

To prove item 2 use Hölder's inequality (see Corollary 4.2.2) to deduce

$$\left| \int_x^{x+h} f(t)\mathrm{d}t \right| \leq \int_x^{x+h} |f(t)|\mathrm{d}t \leq \|f\|_p h^{1-\frac{1}{p}}.$$

From this it is easy to see that the assertion is true. It remains true for $p = \infty$ if the convention $\frac{1}{\infty} = 0$ is used. ∎

In particular the functions with singularity modulus of form $n+C$ for some C are exactly the functions contained in L^∞. The class of functions with linear modulus with slope $(1-1/p)^{-1}$ contains L^p, however, for $p \neq \infty$ equality fails. For $p = 1$ this follows from Example 2.2.4. For $1 < p < \infty$ the gap is less severe but still there, as the next example shows. The corresponding classes for the modulus of continuity are the Lipschitz and Hölder-continuous functions.

Example 2.2.7. Consider the function $f(x) := x^{-\frac{1}{p}}$ on the unit interval. It is easy to see that $\mu(n) := \lceil (n+C)/(1-\frac{1}{p}) \rceil$ is a singularity modulus of f for any $C > \mathrm{lb}(1-\frac{1}{p})$. However, since x^{-1} is not integrable on the unit interval, this function is not included in $\mathrm{L}^p([0,1])$.

2.2.2 The singular representation in one dimension

Recall that the information about a continuous function can be divided into two parts by the characterization of polynomial-time computable functions from Theorem 2.1.4; the first part being approximations to the values on dyadic numbers and the second part being a modulus of continuity. This was used for the definition of the standard representation of continuous functions in Definition 2.1.6.

The following definition carries this idea to the set of integrable functions, where integrals over dyadic intervals replace the point evaluations and the singularity modulus replaces the modulus of continuity. Recall that $\mathbb{D}$ denotes the set of numbers of the form $\frac{m}{2^n}$ for $m \in \mathbb{Z}$ and $n \in \omega$, that $[\![\cdot]\!] : \Sigma^* \to \mathbb{D}$ is the encoding fixed in the introduction and that the length $|\varphi|$ of a length-monotone string function is given by $|\varphi|(|\mathbf{a}|) = |\varphi(\mathbf{a})|$.

Definition 2.2.8. Define the **singular representation** ξ_s of $\mathrm{L}^1([0,1])$: Let a length-monotone string function φ be a ξ_s-name of $f \in \mathrm{L}^1([0,1])$ if for all strings $\mathbf{a}, \mathbf{b}$ with $[\![\mathbf{a}]\!], [\![\mathbf{b}]\!] \in [0,1]$ and all $n \in \omega$

$$\left| \int_{[\![\mathbf{a}]\!]}^{[\![\mathbf{b}]\!]} f(t)\mathrm{d}t - [\![\varphi(\langle \mathbf{a}, \mathbf{b}, n \rangle)]\!] \right| < 2^{-n}$$

and $|\varphi|$ is a singularity modulus of f.

This indeed defines a second-order representation: Firstly, for any distinct integrable functions there exists a dyadic interval such that their integrals over this interval differ. Thus, ξ_s defines a partial function. Secondly, Proposition 2.2.2 proves that for any integrable function there is a singularity modulus and therefore, since padding of dyadic numbers is possible by Lemma 1.1.1, the mapping ξ_s is surjective.

The representation ξ_s is chosen such that it is polynomial-time equivalent to the representation from the introduction of this section. As a result, it possesses the same minimality property. We state this as a theorem, note however, that this is also covered by the following Theorem 2.2.12, featuring a more explicit statement and a direct proof.

Theorem 2.2.9 (Minimality). *ξ_s is a minimal representation of $\mathrm{L}^1([0,1])$ such that the integration operator is polynomial-time computable.*

2.2.3 Higher dimensions

One of the advantages of Definition 2.2.8 over the description from the beginning of this chapter is that it allows a straightforward generalization to higher dimensions. Fix some dimension d and let $\Omega \subseteq \mathbb{R}^d$ be a bounded measurable set. Recall that $\tilde{f}$ denotes the extension of a function to the whole space by zero.

Definition 2.2.10. A function $\mu : \omega \to \omega$ is called a **singularity modulus** of $f \in \mathrm{L}^1(\Omega)$ if it is a singularity modulus (in the sense of Definition 2.2.1) of each of the functions

$$f_i(x) := \int_{\mathbb{R}^{d-1}} \tilde{f}(x_1, \dots x_{i-1}, x, x_{i+1}, \dots, x_d)\mathrm{d}x_1 \cdots \mathrm{d}x_{i-1}\mathrm{d}x_{i+1} \cdots \mathrm{d}x_d.$$

(compare Figure 2.4).

Recall from the introduction that for $x, y \in \mathbb{R}^d$ the box with corners x and y and edges parallel to the axis is denoted by $[x, y]$ and that the box $[[\![\mathbf{a}]\!]_d, [\![\mathbf{b}]\!]_d]$ is abbreviated as $[\mathbf{a}, \mathbf{b}]$.

Definition 2.2.11. Define the **singular representation** ξ_s of $\mathrm{L}^1(\Omega)$: A length-monotone string function φ is a ξ_s-name of $f \in \mathrm{L}^1(\Omega)$ if for all strings $\mathbf{a}, \mathbf{b}$ with $[\![\mathbf{a}]\!]_d, [\![\mathbf{b}]\!]_d \in \mathbb{D}^d$ and $n \in \omega$

$$\left| \int_{[\mathbf{a},\mathbf{b}]} \tilde{f}\mathrm{d}\lambda - [\![\varphi(\langle \mathbf{a}, \mathbf{b}, n\rangle)]\!] \right| < 2^{-n}$$

and $|\varphi|$ is a singularity modulus of f.

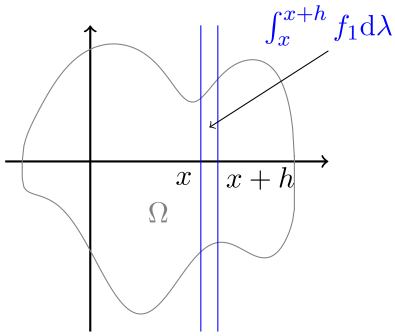


Figure 2.4: f_1 in two dimensions.

Since no source for a multidimensional generalization of the minimality of a standard representation of continuous functions from Theorem 2.1.7 is known to the author, a direct proof of the minimality is given.

Theorem 2.2.12 (Minimality). *For a second-order representation ξ of $\mathrm{L}^1(\Omega)$ the following are equivalent:*

- *The integration operator from* (INTu) *is computable in polynomial time with respect to ξ.*
- *ξ_s is polynomial-time reducible to ξ.*

Proof. The proof is very similar to the proof of [KC10, Lemma 4.9], i.e. of the minimality of $\xi_{\mathcal{C}}$ from Theorem 2.1.7.

First assume that ξ is a representation such that the operator from eq. (INTu) is polynomial-time computable. Construct an oracle Turing machine that whenever given a ξ-name φ of a function $f \in \mathrm{L}^1(\Omega)$ returns correct values of a ξ_s-name of f: This machine can simulate a machine computing the integration operator in polynomial-time to obtain approximations to the integrals over f from φ.

To get hold of a singularity modulus of the input function let P be a second-order polynomial such that the integration operator is computable in time P. Recall from Example 1.2.19 that whenever C is an upper bound on the supremum norm of the elements of $\Omega \times \Omega \subseteq \mathbb{R}^{2d}$, then the function $l(n) := 2d(n + C + 4)$ is a length of the representation used for $\Omega \times \Omega$. The function

$$\mu : \omega \to \omega, \quad n \mapsto P(\langle |\varphi|, l\rangle, n + 1)$$

is a singularity modulus of f: When queried for an approximation with quality 2^{-n-1} the machine computing the integration operator can at most take $\mu(n)$ steps. Therefore, it knows the boundaries a and b of the integral with precision at most $2^{-\mu(n)}$. Recall the definition of the singularity modulus from Definition 2.2.10, in particular that f_i arises from f by integrating over all but the i-th variable. Since Ω is bounded, there are some $\mathbf{c}$ and $\mathbf{d}$ such that $\Omega \subseteq [\mathbf{c},\mathbf{d}]^d$. Note that for any $x \in [\mathbf{c},\mathbf{d}]$ and $h \in \mathbb{R}$ with $|h| \leq 2^{-\mu(n)}$ there is a dyadic vector $a = [\![\mathbf{a}]\!]$ which is a valid $2^{-\mu(n)}$-approximation for both x and $x+h$.

The argument works the same for any i. Set $i = d$ from now on to simplify notation. Define length-monotone string functions φ_a^+ and φ_a^- by

$$\varphi_a^+(\mathbf{b}) := \langle \mathbf{d}, \ldots, \mathbf{d}, \mathbf{a} \rangle, \text{ resp. } \varphi_a^-(\mathbf{b}) := \langle \mathbf{c}, \ldots, \mathbf{c}, \mathbf{a} \rangle.$$

Let q be the approximation encoded in the output of the machine computing $\mathrm{INT_u}$ when handed $\langle \varphi, \varphi_a^-, \varphi_a^+ \rangle$ as oracle and $\mathtt{1}^{n+1}$ as precision requirement. Since a is a $2^{-\mu(n)}$-approximation to both x and $x+h$ and $[\mathbf{c},\mathbf{d}]^{d-1} \times [a,a]$ is a set of Lebesgue measure zero it holds that

$$\begin{aligned}\left|\int_x^{x+h} f_i \mathrm{d}\lambda\right| &\leq \left|\int_{\mathbb{R}^{d-1}\times[x,x+h]} \tilde{f}\mathrm{d}\lambda - q\right| + |q| \\ &\leq \left|\int_{[\mathbf{c},\mathbf{d}]^{d-1}\times[x,x+h]} \tilde{f}\mathrm{d}\lambda - q\right| + \left|q - \int_{[\mathbf{c},\mathbf{d}]^{d-1}\times[a,a]} \tilde{f}\mathrm{d}\lambda\right| < 2^{-n}.\end{aligned}$$

It is left to show that ξ_s renders the integration operator polynomial-time computable. Assume a ξ_s name φ of a function f, an oracle for a box and a precision requirement $\mathtt{1}^n$ are given. Get approximations to the vertices of the box with precision $\mathtt{1}^{|\varphi|(\mathtt{1}^n)+\lceil \mathrm{lb}(d)\rceil+1}$ and query φ for a 2^{-n-1}-approximation over this box. An triangle inequality argument shows that this is a valid approximation to the integral over the box. ■

The result includes null sets: In this case $\mathrm{L}^1(\Omega)$ only contains one element and the integration operator is the constant zero function.

2.2.4 Discontinuity

Since the proof of discontinuity is most naturally stated for the unit interval, we state a restricted version first:

Proposition 2.2.13 (Discontinuity). *The singular representation of* $\mathrm{L}^1([0,1])$ *is discontinuous with respect to the norm topology.*

Proof. Consider the sequence of functions

$$f_m : [0,1] \to [-1,1], \quad x \mapsto (-1)^{\min\{k\in\mathbb{N} \mid k2^{-m} \geq x\}}$$

(compare Figure 2.5). That is: Divide $[0,1]$ into 2^m equally sized intervals and let the function values alternate between constantly being -1 an 1 respectively on these intervals. The functions f_m are bounded by 1 and thus allow the common singularity modulus $n \mapsto n+1$ by Proposition 2.2.5. Observe that the absolute value of an integral of f_m over an interval is always smaller than the minimum of the length of the interval and 2^{-m}. From $\left|\langle \mathbf{a}, \mathbf{b}, 1^k \rangle\right| < 1^m$ it follows that $k < m$, thus it is possible to choose a name φ_m of f_m such that upon inputs $\mathbf{c}$ of length less than m the return value $\varphi_m(\mathbf{c})$ is the unique encoding of 0 of length $2(|\mathbf{c}|+1)$. This sequence φ_m converges to the function φ of length $2(n+1)$ always returning an encoding of zero. Obviously φ is a name of the zero function. However, $\xi_s(\varphi_m) = f_m$ has norm 1 for all m and therefore does not converge to the zero function in norm. This proves sequential and therefore also topological discontinuity of ξ_s. ∎

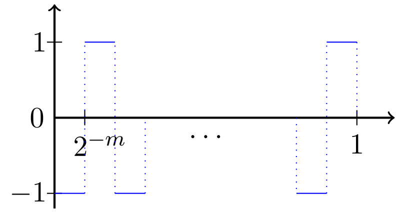


Figure 2.5: The function f_m

Theorem 2.2.14 (Discontinuity). *Whenever $\Omega \subset \mathbb{R}^d$ is a bounded set with non-empty interior, the singular representation ξ_s of $\mathrm{L}^1(\Omega)$ from Definition 2.2.8 is discontinuous with respect to the L^1-norm.*

Proof. Since the set Ω has non-empty interior, there exists a small box with edges parallel to the axis and dyadic endpoints that is completely included in Ω. Lift the function sequence from the proof of Proposition 2.2.13 to a box by assuming them to be independent of the additional variables. Scale this box and the functions to fit inside of Ω. The arguments of the above proof still work for this new sequence and show discontinuity. ∎

Proposition 2.2.13 can be improved by replacing the norm topology with another common but weaker topology: The weak toplogy.

Theorem 2.2.15. *The singular representation of $\mathrm{L}^1([0,1])$ is not continuous with respect to the weak topology.*

Proof. The weak topology on $\mathrm{L}^1([0,1])$ is the initial topology of the set linear functionals on $\mathrm{L}^1([0,1])$ continuous with respect to the norm topology. It is well known, that each continuous functional is of the form

$$\psi_g : \mathrm{L}^1([0,1]) \to \mathbb{R}, \quad f \mapsto \int_0^1 fg \mathrm{d}\lambda$$

for some $g \in \mathrm{L}^\infty([0,1])$.

To prove discontinuity it suffices to construct an element g of $\mathrm{L}^\infty([0,1])$ such that the corresponding functional ψ_g is discontinuous with respect to the final topology of the representation. Discontinuity of ψ_g is proven by specifying a sequence of functions $f_n \in \mathrm{L}^1([0,1])$ and a sequence of names φ_n of f_n such that the sequence φ_n converges to a name of the zero function but for all n

$$\psi_g(f_n) = \int_0^1 f_n g \mathrm{d}\lambda \geq \frac{1}{2}.$$

Define g as sketched in Figure 2.6, that is by

$$g := \sum_{m=1}^{\infty} \sum_{n=1}^{2^m} (-1)^n \chi_{[1-2^{-m}, 1-2^{-m}+n2^{-2m})}$$

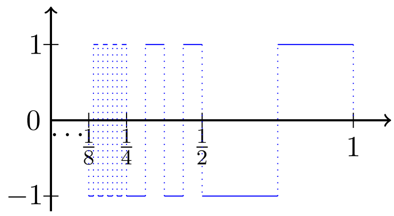


Figure 2.6: The function g

Define the functions f_m by restricting g to the interval $[1-2^{-m}, 1-2^{-m-1})$ and multiplying it by 2^m. Then

$$\int_0^1 f_m \cdot g \mathrm{d}\lambda = \int_0^1 2^m \chi_{[1-2^{-m}, 1-2^{-m-1})} \mathrm{d}\lambda = 1/2.$$

Note that for any a, b

$$\int_a^b f_m \mathrm{d}\lambda \leq 2^{-m}.$$

Furthermore, f_m is bounded by 2^m. The function $n \mapsto 2n+1$ is a common singularity modulus for all the f_m. Choose the names φ_m like they were chosen in the proof of Proposition 2.2.13. Again, the sequence of names converges to a name of the zero function in Baire space. Thus ξ_s is not continuous with respect to the weak topology. ■

Remark 2.2.16. It is not known to the author if the final topology of ξ_s coincides with any topology of $\mathrm{L}^1(\Omega)$ that has been considered before.

A discontinuous representation is unsatisfactory, as the metric cannot be computed. With respect to the singular representation not even the norm of a function can be computed.

2.3 Spreads

This chapter discusses a technique that allows us to lift the representation from the previous chapter to a representation computably equivalent to the Cauchy representation of L^1 while preserving the minimality property at least for the computably equivalent representations. However, it should be noted that this process does not lead to a representation that is likely to be useful in practice. As the only way to employ the additional information granted is to do an unbounded search and no operators that have not been bounded time computable before become bounded time computable afterwards.

This chapter applies a general construction due to Kawamura and Pauly [KP14] to regularize the singular representation. They construct from an arbitrary representation ξ of a space X another representation that has complexity theoretically pathological properties: Any function $f : \mathbf{Y} \to X$ that is computable with respect to ξ is polynomial-time computable with respect to the new representation. They call this representation the 'padding' of this representation. Since the term 'padding' already bears a different meaning in this thesis, we call this representation the spread instead.

2.3.1 The spread of a representation

Consider the following operation on second-order representations and second-order represented spaces:

Definition 2.3.1. Let $\mathbf{X} = (X, \xi)$ be a second-order represented space. The **spread** $\xi^s :\subseteq \mathcal{B} \to X$ of ξ is defined as follows: A length-monotone string function φ is a ξ^s-name of an element $x \in X$ if and only if there exists a ξ-name ψ and a function $\sigma : \omega \to \omega$ such that

$$\varphi(\mathbf{a}) = \begin{cases} \mathtt{1}\psi(\mathbf{b}) & \text{if } \mathbf{a} = \mathtt{1}^m\mathtt{0}\mathbf{b} \text{ for some } m \geq \sigma(|\mathbf{b}|) + 1 \\ \mathtt{0}^m & \text{otherwise.} \end{cases}$$

Let the **spread** $\mathbf{X}^s$ of $\mathbf{X}$ be the second order represented space (X, ξ^s).

The representation ξ^s is the minimal representation in the class of representations computably equivalent to ξ with respect to polynomial-time reductions:

Proposition 2.3.2 ([KP14]). *For any representation $\xi :\subseteq \mathcal{B} \to X$, the spread ξ^s is computably equivalent to ξ. Whenever ξ is computably reducible to a representation ξ', ξ^s is polynomial time reducible to ξ'.*

Proof. First note that a ξ^s name of an element can be obtained from an ξ-name φ by first removing leading `1`s, then an `0`, feeding this to φ and adding a `1` to the return value. This corresponds to the choice $\sigma(n) = 0$ in the definition.

For the opposite direction let $M^?$ be the machine that upon input $\mathbf{a}$ queries its oracle on $\mathtt{1}\mathtt{1}^m\mathtt{0}\mathbf{a}$ starting with $m = 0$ and increasing m each time the return value is empty or starts with `0`. Once it finds a return value that starts with `1` it removes the first digit and returns the rest.

Now let ξ' be an arbitrary representation that ξ^s can be computably reduced to. To prove that ξ^s is polynomial time reducible to ξ' let $M^?$ be an oracle Turing machine computing the translation. Produce a ξ^s-name from a ξ'-name φ in polynomial time as follows: Let $N^?$ be a machine that on input $\mathtt{1}\mathtt{1}^n\mathtt{0}\mathbf{a}$ and oracle φ simulates the runs of M^φ on the elements $\mathtt{0}^m$ for $m < |\mathbf{a}|$ and then on $\mathbf{b}$ for all strings of length $|\mathbf{a}|$ for n steps. If it runs out of steps it returns $\mathtt{0}^k$ where k is the longest return value M^φ produced so far. If it does not run out of time it returns what the simulation of M^φ on $\mathbf{a}$ returned. The machine $N^?$ can be checked to compute the ξ^s -name, where in the notation of Definition 2.3.1 ψ is the function computed by M^φ and $\sigma(n)$ is the time the simulations need for $|\mathbf{a}| = n$. ■

2.3.2 Joins

We need the following construction, that can for instance also be found in [Wei00; Sch02].

Definition 2.3.3. Let $F, G :\subseteq \mathcal{B} \rightrightarrows X$ be multivalued functions. The **join** $F \wedge G :\subseteq \mathcal{B} \rightrightarrows X$ of these functions is defined as follows:

$$(F \wedge G)(\langle \varphi, \psi \rangle) := F(\varphi) \cap G(\psi).$$

The following properties are immediate from the definition:

Lemma 2.3.4.

- *$F \wedge G$ is surjective if and only if both F and G are surjective.*
- *If either F or G is single-valued, then $F \wedge G$ is single-valued.*

In particular the join of a surjective function with a representation is a representation again. If F and G are both representations, taking the join is the same as restricting the product representation to the diagonal of $X \times X$ and identifying the diagonal with X. In this case a name of an element in the new representation is a pair of a name in each of the representations.

Proposition 2.3.5. *Let $f : \mathbf{X} \to \mathbf{Y}$ be a computable function between represented spaces. Then*

$$\xi_f := (f^{-1} \circ \xi_{\mathbf{Y}}) \wedge \xi^s_{\mathbf{X}}$$

is a representation such that for any representation ξ' computably equivalent to $\xi_{\mathbf{X}}$ such that if f polynomial-time computable with respect to ξ' it holds that ξ_f is reducible to ξ' in polynomial time.

Proof. Since $\delta_{\mathbf{Y}}$ is single-valued the composition is well-defined. As f is a total function, $f^{-1} \circ \xi_{\mathbf{Y}}$ is surjective. Thus $(f^{-1} \circ \xi_{\mathbf{Y}}) \wedge \xi^s_{\mathbf{X}}$ indeed defines a representation. That $\xi_{\mathbf{X}}$ is computably reducible to this representation follows from its reducibility to $\xi^s_{\mathbf{X}}$. For the reducibility in the other direction let F be a computable realizer of f and S a computable translation from $\xi_{\mathbf{X}}$ to $\xi^s_{\mathbf{X}}$. It is straight forward to check that $\varphi \mapsto \langle F(\varphi), S(\varphi)\rangle$ the desired translation.

To prove the minimality assume that ξ is a representation that is computably equivalent to $\xi_{\mathbf{X}}$ and renders f polynomial-time computable. By Proposition 2.3.2 $\xi^s_{\mathbf{X}}$ is polynomial time reducible to ξ. Let S and F be polynomial-time computable realizers of the translation resp. the function. Again, it is straightforward to check that $\varphi \mapsto \langle F(\varphi), S(\varphi)\rangle$ is a polynomial-time translation. ∎

In the previous result the computability condition can be omitted. In this case, however, the resulting representation may not be computably reducible to the original representation.

In the general case there cannot be said much about computability or bounded time computability of operations with respect to the join of two representations. However, for joins with spreads, a lot can be said.

Proposition 2.3.6. *Let ξ and ξ' be representations of the same space X. Then a function $f : X \to \mathbf{Y}$ can be computed in bounded time with respect to $\xi \wedge \xi'^s$ if and only if it can be computed in bounded time with respect to ξ.*

Proof. It is clear that f can be computed in bounded time with respect to $\xi \wedge \xi'^s$ if it can already be computed in bounded time with respect to ξ. For the other direction let

F be a realizer of f with respect to $\xi \wedge \xi'^s$. A realizer F' computing f with respect to ξ within a similar time bound can be defined by $F'(\varphi) := F(\langle \varphi, n \mapsto \varepsilon \rangle)$. To see that this indeed computes f let n be the number of steps F is granted on input $\mathbf{a}$ and oracle φ. Choosing σ such that $\sigma(i) \geq n$ for all i in Definition 2.3.1 shows that there is a valid ξ^s-name of $\xi(\varphi)$ that only returns ε on strings of length smaller than n. Therefore the runs of F on this name and the actual input coincide, and F' computes f. ∎

2.3.3 Applications

Our first application is regularizing the singular representation ξ_s of the integrable functions from Section 2.2. Let Ω be a bounded set, then the rational polynomials are a dense sequence in the space $\mathrm{L}^1(\Omega)$. Recall that ξ_s is the singular representation: The weakest representation such that integration is polynomial-time computable. This representation was shown to be discontinuous with respect to the L^1-norm in Theorem 2.2.14. We can fix this as follows.

Definition 2.3.7. Let ξ be the Cauchy representation of $\mathrm{L}^1(\Omega)$ with respect to the sequence above (compare to Definition 1.2.2). Define the **regular representation** ξ_r of $\mathrm{L}^1(\Omega)$ by

$$\xi_r := \xi_s \wedge \xi^s.$$

From the introduction of Section 2.2 we know that for $\Omega = [0,1]$ it holds that $\xi_s = \mathrm{INT}^{-1} \circ \xi_{\mathcal{C}}|^{\mathcal{AC}([0,1])}$ and therefore the above coincides with the representation ξ_{INT} from Proposition 2.3.5 and is the weakest representation equivalent to the standard representation such that the integration operator $\mathrm{INT} : \mathrm{L}^1([0,1]) \to \mathcal{C}([0,1])$ is polynomial time computable.

The following theorem is just collecting what we already know:

Theorem 2.3.8. *The regular representation is the weakest representation computably equivalent to the Cauchy representation such that integration is polynomial time computable: Whenever the regular representation is computably reducible to another representation ξ and integration is polynomial time computable with respect to ξ, then the regular representation is polynomial time reducible to ξ. Scalar multiplication and addition are polynomial-time computable with respect to the regular representation.*

Proof. The minimality follows directly from the minimality of the singular representation from Theorem 2.2.12 together with the minimality of ξ^s from Proposition 2.3.2.

Polynomial-time computability of addition and scalar multiplication follow from the polynomial-time computability of these operations with respect to the singular representation and computability with respect to the Cauchy representation together with the minimality of ξ^s from Proposition 2.3.2 again. ∎

The step from the singular representation ξ_s to the regular representation ξ_r takes us from a discontinuous representation to a representation such that the norm is computable. By Proposition 2.3.6, however, it does not change anything about the class of functions computable within bounded time. In particular neither the norm nor the metric is computable in bounded time with respect to the representation ξ_r.

Example 2.3.9. The construction from Proposition 2.3.5 can be applied with more than one function and more than one join to instead of ξ_r obtain a representation such that both integration and the norm are polynomial time computable. While this process preserves polynomial-time computability of the scalar multiplication it is not clear if it preserves polynomial time computability of addition. This can not easily be fixed by iterating the above construction further, as the addition is not defined on the represented space itself but the product of it with itself and it is not clear how to obtain from a product representation a representation of one of the factors.

Note that any unbounded function $l : \omega \to \omega$ is a length of the spread of a representation. Thus, if for an element x the value $f(x)$ has a name of (unbounded) length l in the representation $\xi_{\mathbf{Y}}$ in Proposition 2.3.5, then x has a name of length $2(l+1)$ in the representation ξ_f. It follows that the above process making the norm polynomial time computable only shifts the length of names by a multiple of $\mathrm{lb}(\|f\|_1)$. The results in the upcoming chapters prove that it is impossible to make the norm, the scalar multiplication and the addition polynomial-time computable at the same time without considerably increasing the length of names (compare the introduction of Chapter 5, Theorem 3.2.11 and Example 4.2.10).

Before we can construct representations that allow us to compute the norm in bounded time while maintaining polynomial-time computability of the addition and scalar multiplication in Chapter 5 we need to make a detour to complexity theory on general metric spaces.

3 Metric spaces

This chapter investigates in how far the Cauchy representation from computability theory can be generalized to complexity theory. More specifically it relates the existence of second-order representations that are well behaved in the sense that they have a small length and the metric is of low computational complexity to an inherent metric property of that space: The metric entropy. Metric entropy was first introduced by Kolmogorov and Tihomirov in [KT59]. It has been considered more or less directly throughout different fields of research: approximation theory [Tim94; Lor66], constructive analysis [Bis67] resp. [BB85], proof theory [Koh08] and also computable analysis and real complexity theory [Wei03].

Throughout this chapter consider metric spaces M, N, ... and denote the corresponding metrics by d_M, d_N etc. The reader is assumed to be familiar with the basic theory of metric spaces. Recall, that the balls

$$B_r(x) := \{y \in M \mid d_M(x, y) < r\} \tag{3.1}$$

form a base of the topology of a metric space. We adopt the convention that a 'ball' is an open ball and call a ball of radius r an r-ball. In arbitrary metric spaces the closed ball of radius r with center x, i.e. the set where the $<$ in (3.1) is replaced by $\leq$, may be larger than the closure of the ball of radius r around x. In normed vector spaces, however, the closed balls and the closures of balls coincide.

3.1 Metric entropies and spanning bounds

It is well known that in a complete metric space a subset is relatively compact (i.e. its closure is compact) if and only if it is totally bounded. The following notion is a straightforward quantitative version of total boundedness and can be used to measure the 'size' of compact subsets of metric spaces. The name of the following modulus is taken from [Koh05], see also [Koh08].

Definition 3.1.1. A function $\nu : \omega \to \omega$ is called **modulus of total boundedness** of $K \subseteq M$, if for any $n \in \omega$ the set K can be covered by $2^{\nu(n)}$ balls of radius 2^{-n}. The smallest modulus of total boundedness of a set K is called its **size** or its **metric entropy** and denoted by $|K|$.

Thus,

$$|K|(n) := \min\{k \in \omega \mid K \text{ can be covered by } 2^k \text{ balls of radius } 2^{-n}\}.$$

While most of the time a modulus of total boundedness is easily specified, the metric entropy of is often hard to get a hold of. In a complete metric space, a closed set has a modulus of total boundedness if and only if it is compact. The idea to classify compactness in complete metric spaces via total boundedness and closedness is due to Brouwer, a collection of his work can be found in [TD88a; TD88b].

Example 3.1.2 (Size of subsets of finite dimensional vector spaces). The function $\mu(n) := d \cdot (n + k + 1)$ is a modulus of total boundedness for the hyper cube $[0, 2^k]^d$ in $(\mathbb{R}^d, |\cdot|_\infty)$. Since $\mathbb{R}^d$ is the union of all these cubes, this implies that each compact subset of $\mathbb{R}^d$ has a linear modulus of total boundedness.

More examples can for instance be found in [KLN15, Section 2.2].

A modulus of total boundedness is an upper bound on the size of a compact set. For providing lower bounds, a slightly different notion is more convenient.

Definition 3.1.3. A function $\eta : \omega \to \omega$ is called a **spanning bound** of $K \subseteq M$, if for any $n \in \omega$ there exist $x_1, \ldots, x_{2^{\eta(n)}} \in K$ such that for all $i, j \in \{1, \ldots, 2^{\eta(n)}\}$

$$i \neq j \quad \Rightarrow \quad d(x_i, x_j) \geq 2^{-n+1}.$$

If there is a biggest spanning bound, it is called the **width** of K and denoted by $\mathrm{wd}\,(K)$.

The condition on the x_i can be read as 'the 2^{-n}-balls around the points x_i are disjoint'. In a complete metric space there exists a biggest spanning bound for a closed set K if and only if it is compact. The connection between spanning bounds and metric entropies is as follows (cf. [Lor66]).

Proposition 3.1.4. *Let K be a subset of a metric space. Whenever ν is a modulus of total boundedness and η a spanning bound of K, then*

$$\eta(n) \leq \nu(n).$$

Furthermore if K has a width, then

$$|K|\,(n) \leq \mathrm{wd}\,(K)\,(n+1) + 1.$$

Proof. Show by contradiction that $\nu(n) + 1$ cannot be the value of a spanning bound on n. Thus, assume it is a value of a spanning bound on n. This means that there are $2^{\nu(n)+1}$ elements x_i such that $d(x_i, x_j) \geq 2^{-n+1}$ for $i \neq j$. Since ν is a modulus of total boundedness there are $2^{\nu(n)}$ elements y_i such that the 2^{-n}-balls around the y_i cover K. Since each x_i has to lie in at least one of the balls and there are more x_i than y_i, there are indices $j \neq l$ and m such that x_j and x_l are both elements of the ball of radius 2^{-n} around y_m. By the triangle inequality it follows that

$$d(x_j, x_l) \leq d(x_j, y_m) + d(y_m, x_l) < 2 \cdot 2^{-n},$$

which is a contradiction to the choice of the x_i.

Concerning the second inequality: For each n choose a maximal set Y_n of elements of K such that the pairwise distance is bigger or equal 2^{-n}. Since $\mathrm{wd}(K)$ is the biggest spanning bound it holds that $\#Y_n < 2^{\mathrm{wd}(K)(n+1)+1}$. But from the maximality of Y_n it follows, that the 2^{-n}-balls around the elements of Y cover K: If x is not contained in any ball, the elements of $Y_n \cup \{x\}$ have pairwise distance bigger or equal 2^{-n}, which contradicts the maximality. Thus $n \mapsto \lceil \mathrm{lb}(\#Y_n)\rceil \leq \mathrm{wd}(K)(n+1)+1$ is a modulus of total boundedness on n and majorizes $|K|(n)$ ∎

Note that Proposition 3.1.4 in particular implies

$$\mathrm{wd}(K)(n) \leq |K|(n) \leq \mathrm{wd}(K)(n+1)+1 \leq |K|(n+1)+1.$$

Thus, the width and the metric entropy need not be equal but are always comparable. For this reason the width is not mentioned again.

3.1.1 Metric entropy in normed spaces

The metric entropy is a metric concept: The metric entropy of compact set may change drastically between metrics that induce the same topology. Metrics inducing the same topology are called **equivalent** metrics.

Example 3.1.5. Consider Cantor space, that is the space $\mathcal{C} := \Sigma^\omega$. This space can be equipped with either of the metrics

$$d(\chi,\tilde\chi) := \begin{cases} 2^{-\min\{n \mid \chi(n)\neq\tilde\chi(n)\}} & \text{if } \chi \neq \tilde\chi \\ 0 & \text{otherwise} \end{cases}$$

or

$$d'(\chi,\tilde\chi) := \begin{cases} \frac{1}{\min\{n \mid \chi(n)\neq\tilde\chi(n)\}+1} & \text{if } \chi \neq \tilde\chi \\ 0 & \text{otherwise.} \end{cases}$$

Both these metrics generate the same topology which makes Cantor space a compact space. However, the metric entropy of Cantor space is $n+1$ with respect to the first metric, while it is 2^{n+1} with respect to the second.

This can be turned into a general construction for an arbitrary compact metric space: Let M be a metric space with metric d and $f : \mathbb{R}^+ \to \mathbb{R}^+$ be monotone, subadditive (i.e. $f(x+y) \leq f(x)+f(y)$) and continuous with $f(x)=0 \Leftrightarrow x=0$. In this case

$$d'(x,y) := f(d(x,y))$$

defines an equivalent metric d'. Whenever $F : \omega \to \omega$ is a function such that $f(2^{-F(n)}) = 2^{-n}$ then

$$|K|'(n) = |K|(F(n)).$$

where $|K|'$ is the metric entropy of K with respect to d' and $|K|$ is the metric entropy with respect to d.

If M is a space of diameter less than 0.25 choose $f(x) := \frac{1}{1-\mathrm{lb}(x)}$ to be the function from Example 2.1.3. This function fulfills $f(0) = 0$, is continuous and monotone. It is also concave on $[0, 0.25]$ and therefore subadditive, thereby fulfilling all of the above conditions. The function $F(n) = 2^n - 1$ satisfies $f(2^{-F(n)}) = 2^{-n}$. Thus, the change of metrics leads to an exponential increase in the metric entropy of sets of at least linear metric entropy.

For normed spaces, however, the situation is a lot better:

Proposition 3.1.6. *Let $\|\cdot\|$ and $\|\cdot\|'$ be norms on a vector space V that induce the same topology. Then there exists a constant $C \in \omega$ such that whenever ν is a modulus of total boundedness of a subset K of V with respect to $\|\cdot\|$, then $\nu'(n) := \nu(n + C)$ is a modulus of total boundedness of K with respect to $\|\cdot\|'$ and vice versa.*

Proof. Note, that norms inducing the same topology are equivalent (for instance [Wer00, Satz I.2.4]). This means that there exists a constant C such that for any $x \in V$

$$2^{-C}\|x\| \leq \|x\|' \leq 2^C\|x\|.$$

From these inequalities it is straightforward to compute the relation between the metric entropies. ■

In particular the class of compact sets allowing a polynomial modulus of total boundedness is well defined in normable spaces.

Example 3.1.7. There is a straightforward way to define a bijection between the Cantor space and the Cantor set. Thus, regard the Cantor space as a subset of $\mathbb{R}$. With respect to the metric inherited from the norm of $\mathbb{R}$, Cantor space allows a linear bound of total boundedness (compare to Example 3.1.5).

A proof of the following can be found in [Wei03] (although the notation is a little different).

Proposition 3.1.8 ([Wei03]). *Let $L_{1,0}$ be the set of Lipschitz continuous functions with Lipschitz constant 1, such that $f(0) = 0$. Then $L_{1,0} \subseteq C([0,1])$ is compact and its metric entropy is given by*

$$|L_{1,0}|(n) = \lceil \mathrm{lb}(3)2^n \rceil.$$

It is not a coincidence that a subset of an infinite dimensional vector space is considered in the previous proposition: Any finite dimensional vector space is isomorphic to $\mathbb{R}^d$ for some dimension d. The compact subsets of $\mathbb{R}^d$ all allow a linear modulus of total boundedness as each compact set is included in a cube and the cubes allow a linear modulus of total boundedness by Example 3.1.2. For infinite dimensional vector spaces it is always possible to find subsets that are arbitrarily big in the following sense:

Theorem 3.1.9. *Whenever X is an infinite dimensional normed space and $\mu : \omega \to \omega$ is a function, then there is a compact subset K of X such that $|K| \geq \mu$.*

Proof. By Riesz's Lemma there exists a sequence of elements x_i such that $\|x_i\| = \frac{3}{4}$ and $\frac{3}{4} > \|x_i - x_j\| \geq \frac{1}{2}$ for all $i \neq j$. This means that a ball of radius 1 around zero contains all of the x_i, but a ball of radius $\frac{1}{4}$ contains at most one x_i. Define a set $K \subseteq X$ by

$$K := \{0\} \cup \bigcup_{i \in \mathbb{N}} \bigcup_{j=0}^{2^{\mu(2i)} - 2^{\mu(2(i-1))}} \{4^{-i+1} x_j\}.$$

To cover K with balls of radius 4^{-i} we need at least one ball for each of the $2^{\mu(2i)}$ many elements $x \in K$ with $\|x\| \geq 3 \cdot 4^{-i}$. Therefore $\big|K\big|(2i) \geq \mu(2i)$ for all $i \in \omega$. To get the metric entropy to be bigger than μ in every point, apply the above construction to the function $\tilde{\mu} : \omega \to \omega$ defined by

$$\tilde{\mu}(n) := \begin{cases} \max\{\mu(n), \mu(n+1)\} & \text{if } n \text{ is even} \\ \max\{\mu(n-1), \mu(n)\} & \text{if } n \text{ is odd} \end{cases}$$

instead of μ itself.

To argue that the constructed set K is compact, show sequential compactness. For metric spaces this is equivalent to compactness. Consider an arbitrary sequence in K. From the construction it is clear that the ball of any radius around zero contains all but finitely many elements of K. By the pigeonhole principle there is either an element of K that is visited infinitely often by the sequence, or there is an element of the sequence in any ball around zero. In the first case there is a constant subsequence, in the second case there is a subsequence that is convergent to zero. Therefore the set is compact. ■

It is well known that for topological vector spaces finite dimensionality and local compactness coincide. For Banach spaces additionally σ-compactness is equivalent to finite dimension. A similar characterization follows from the above:

Corollary 3.1.10. *For a normed space the following are equivalent:*

1. *It is infinite dimensional.*
2. *For any $\nu : \omega \to \omega$ there exists a compact subset K such that $|K| \geq \nu$.*
3. *There exists a compact subset that has no linear modulus of total boundedness.*

Proof.

1. $\Rightarrow$*2.*: This is Theorem 3.1.9.
2. $\Rightarrow$*3.*: This is trivial.
3. $\Rightarrow$*1.*: By contradiction assume that the normed vector space was finite dimensional. Choose a basis to identify it with $\mathbb{R}^d$. Since all norms on a finite dimensional vector space are equivalent, Proposition 3.1.6 shows that it is irrelevant which norm is used for whether or not all compact sets have a linear modulus of total boundedness. From Example 3.1.2 it is clear that with respect to the supremum norm all sets have a linear modulus of total boundedness. This contradicts the third item. ■

3.1.2 Moduli of continuity

Metric entropy is very tightly connected to moduli of continuity. The notion of a modulus of continuity of a function from Definition 2.1.1 generalizes to arbitrary metric spaces. To be formally correct we give the definition for metric spaces separately and for simplicity we omit the additional assumption that a modulus of continuity has to be strictly increasing whenever it is not zero. This condition is not needed for the results of this chapter, and if it is imposed the results remain correct. Note, however, that for the later chapters this condition is indeed needed and Definition 2.1.1 is used.

Definition 3.1.11. Let M and N be metric spaces and $f : M \to N$ be a function. A function $\mu : \omega \to \omega$ is called a **modulus of continuity** of f, if for all $x, y \in M$

$$d_M(x, y) \leq 2^{-\mu(n)} \quad \Rightarrow \quad d_N(f(x), f(y)) < 2^{-n}.$$

It is clear that each uniformly continuous function, in particular each continuous function on a compact domain, has a modulus of continuity. Conversely, a function that allows a modulus of continuity is uniformly continuous.

If f has μ as a modulus of continuity then for all $x \in M$

$$f(B_{2^{-\mu(n)}}(x)) \subseteq B_{2^{-n}}(f(x)). \tag{B}$$

Example 2.1.2 generalizes:

Example 3.1.12 (Small moduli). A function between metric spaces is Lipschitz continuous if and only if it has a modulus of continuity of the form $n \mapsto n + C$ and in the latter case $\mathrm{lb}(C)$ is a Lipschitz constant. If the domain is of finite diameter, Hölder continuity corresponds to having a linear modulus of continuity.

There exists a correlation between moduli of continuity and discrepancies of the metric entropy of a set and its image:

Lemma 3.1.13. *Let M and N be metric spaces. If $f : M \to N$ has a modulus of continuity μ and $K \subseteq M$ is compact then*

$$|f(K)| \leq |K| \circ \mu.$$

More generally: If ν is a modulus of total boundedness of K, then $\nu \circ \mu$ is one of $f(K)$.

Proof. Prove the more general statement. Let n be fixed. Since ν is a modulus of total boundedness of K, there is a collection of $2^{\nu(\mu(n))}$ balls of radius $2^{-\mu(n)}$ that cover K. Consider the collection of balls of radius 2^{-n} around the images of the centers of these balls. Since μ is a modulus of continuity, the original balls will be mapped to these balls (cf. Equation (B)) and cover $f(K)$. This proves that $\nu \circ \mu$ is a modulus of total boundedness for $f(K)$. ■

Lemma 3.1.14. *Let M and N be metric spaces and $f : M \to N$ be a function. If $\tilde{\mu} : \omega \to \omega$ is such that $|f(K)| \leq |K| \circ \tilde{\mu}$ for any compact set $K \subseteq M$, then $\mu(n) := \tilde{\mu}(n+1)$ is a modulus of continuity of f.*

Proof. The map

$$\nu(n) := \begin{cases} 0 & \text{if } d_M(x,y) < 2^{-n} \\ 1 & \text{if } d_M(x,y) \geq 2^{-n} \end{cases}$$

is modulus of total boundeness of the two point set $\{x,y\}$. By the assumption

$$(\nu \circ \tilde{\mu})(n) = \nu(\tilde{\mu}(n)) = \begin{cases} 0 & \text{if } d_M(x,y) < 2^{-\tilde{\mu}(n)} \\ 1 & \text{if } d_M(x,y) \geq 2^{-\tilde{\mu}(n)} \end{cases}$$

is a metric entropy of the two point set $\{f(x), f(y)\}$. But the metric entropy of a two point set being zero on n means by the triangle inequality that the elements are no further apart than 2^{-n+1}. Therefore, for all x, y and n

$$d_M(x,y) \leq 2^{-\tilde{\mu}(n+1)} \quad \Rightarrow \quad d_N(f(x), f(y)) < 2^{-n},$$

that is: $\mu(n) := \tilde{\mu}(n+1)$ is a modulus of continuity of f. ■

These lemmas together prove:

Theorem 3.1.15. *Let $\mathcal{A} \subseteq \omega^\omega$ be a class of functions that is closed under shifts, i.e. whenever $\mu \in \mathcal{A}$, then also $n \mapsto \mu(n+1) \in \mathcal{A}$. Let $f : M \to N$ be a function between metric spaces, then the following are equivalent:*

- *f has a modulus of continuity from $\mathcal{A}$.*
- *There is a function $\mu \in \mathcal{A}$ such that for any compact set $K \subseteq M$ it holds that $|f(K)| \leq |K| \circ \mu$.*

Chapter 4 investigates another, very different connection between metric entropy and moduli of continuity.

3.2 Metric entropy and complexity

This chapter investigates connections between the concept of metric entropy introduced in the previous Section 3.1 and computational complexity. For this we often consider the first order function that arises if the first order argument of a running time $T : \omega^\omega \times \omega \to \omega$ is fixed. I.e. the function $n \mapsto T(l,n)$ denoted by $T(l,\cdot)$. To formulate the main result we use the following notation:

Definition 3.2.1. A class $\mathcal{A} \subseteq \omega^\omega$ is called **stable** if it is closed under shift and polynomial application. Closure under shift means resp. polynomial application mean that

$$\mu \in \mathcal{A} \Rightarrow n \mapsto \mu(n+1) \in \mathcal{A} \quad \text{and} \quad \mu \in \mathcal{A},\ p \in \mathbb{N}[X] \Rightarrow n \mapsto p(\mu(n)+n) \in \mathcal{A}.$$

Examples of such classes are the class of polynomials or the classes of functions $2^{O(n)}$. The goal of this chapter is to show that the following holds:

Theorem 3.2.2. *Let M be a compact metric space. For any stable class $\mathcal{A} \subseteq \omega^\omega$ the following are equivalent:*

1. *M has a modulus of total boundedness $\mu \in \mathcal{A}$.*
2. *There exists a representation of M and a function $l : \omega \to \omega$ such that l is a length of the representation and the metric can be computed in time $T : \omega^\omega \times \omega \to \omega$ and $T(l, \cdot) \in \mathcal{A}$.*

Since both of the implications of the equivalence stated in the theorem can be made more general, we divide the proof into separate results and prove them in the sections of this chapter. The single theorems are put together to a proof of the above on page 53.

Recall the tuple functions from the introduction: For any given dimension d and given strings $\mathbf{a}_1, \ldots, \mathbf{a}_d$ denote by $\mathbf{c}_i = c_{i,1} \ldots c_{i,\max\{|a_i|\}+1}$ the padding of $\mathbf{a}_i$ to length $\max\{|\mathbf{a}_i| \mid i \in \{1, \ldots, d\}\} + 1$ by appending a $\mathtt{1}$ and then an appropriate number of $\mathtt{0}$s. Then set

$$\langle \mathbf{a}_1, \ldots, \mathbf{a}_d \rangle := c_{1,1} \ldots c_{d,1} c_{1,2} \ldots c_{d,2} \ldots c_{d,\max\{|a_i|\}+1}.$$

For the convenience of the reader we collect some properties of these tuple functions that are of importance in this chapter in a lemma.

Lemma 3.2.3 (Tuple functions). *The tuple functions fulfill for any d:*

- *Whenever the $d(n+1)$ initial segments of $\langle \mathbf{a}_1, \ldots, \mathbf{a}_d \rangle$ and $\langle \mathbf{b}_1, \ldots \mathbf{b}_d \rangle$ coincide, the n initial segments of $\mathbf{a}_i$ and $\mathbf{b}_i$ coincide for all $i \in \{1, \ldots, d\}$.*
- *Both $\langle \cdot, \ldots, \cdot \rangle$ and the projections defined by*

  $$\pi_i(\mathbf{b}) := \begin{cases} \mathtt{0}a_i & \text{if } \mathbf{b} = \langle \mathbf{a}_1, \ldots, \mathbf{a}_d \rangle \\ \varepsilon & \text{if } \mathbf{b} \notin \operatorname{img}(\langle \cdot, \ldots, \cdot \rangle) \end{cases}$$

  are computable in linear time.
- $|\langle \mathbf{a}_1, \ldots, \mathbf{a}_d \rangle| = d(\max\{|\mathbf{a}_i|\} + 1)$.

Note, that it is also possible to encode finite sequences of arbitrary length by the function $(\mathbf{a}_1, \ldots, \mathbf{a}_N) \mapsto \langle N, \langle \mathbf{a}_1, \ldots, \mathbf{a}_N \rangle \rangle$. To avoid ambiguous notation, this construction is written out explicitly when used.

In principle standard tuple functions could be used. However, in this case the bounds obtained are worse. Additionally some more work has to be invested to provide explicit bounds similar to those specified in Lemma 3.2.3.

3.2.1 From complexity to metric entropy

Fix a second-order representation of a metric space. The goal of this section is to obtain from a time bound of the metric a modulus of total boundedness of certain subsets of the metric space. The argument works in a very general setting, in particular the assumption about the space is weaker than computability of the metric in bounded time. This is not very surprising, as the metric entropy only mentions small balls and does not use any information about the exact values of the distance of points far away from each other.

Equality in a metric space equipped with a Cauchy representation is usually not decidable. However, for two given elements x, y of a represented metric space M it is decidable whether or not the elements are far apart. This can be formalized as computability of the following multivalued function:

Definition 3.2.4. For a metric space M define its **equality function** $\mathrm{eq} : M \times M \rightrightarrows \mathcal{B}$ by:

$$\mathrm{eq}(x,y) = \big\{\varphi \mid \forall n : (d(x,y) < 2^{-n-1} \Rightarrow \varphi(\mathtt{1}^n) = \mathtt{1} \wedge d(x,y) > 2^{-n} \Rightarrow \varphi(\mathtt{1}^n) = \mathtt{0})\big\}.$$

More intuitively one might write eq down by case distinction:

$$\mathrm{eq}(x,y)(\mathtt{1}^n) = \begin{cases} \mathtt{1} & \text{if } d(x,y) < 2^{-n-1} \\ \mathtt{0} & \text{if } d(x,y) \geq 2^{-n} \\ \text{arbitrary} & \text{otherwise.} \end{cases}$$

The following is the central notion we work with:

Definition 3.2.5. Let M be a second-order represented metric space. We say that **equality is approximable in time** T, if the equality function $\mathrm{eq} : M \times M \rightrightarrows \mathcal{B}$ is computable in time T.

Since equality on the real numbers is approximable in time $O(\tilde{T})$ for $\tilde{T}(l,n) = n$ (this uses the last property of the definition of the standard representation of real numbers, otherwise we would end up with $T(l,n) = l(n)$) the equality of a metric space is approximable in time $O(T + n)$ whenever its metric is computable in time T.

A running time bound of a machine computing a function restricts the access the machine has to the oracles. The following proposition describes this dependence in detail. It assigns to a machine a function $\mathcal{B} \times \Sigma^* \to \Sigma^*$ that takes as input an oracle and a string, and whose return value is a description of the communication between the machine and the oracle. In particular, if the first argument is changed the return values only change if the machine can distinguish the oracles in a computation with the second input as input string. Furthermore, it specifies a bound on the size of the values of this function from a running time of the machine. This function is very closely related to moduli of sequentiality for instance discussed in [BK02]. Our approach differs from the one taken there in that a 'dialog' describes the return values of the oracle instead of the queries.

Proposition 3.2.6 (Communication functions). *For any oracle Turing machine $M^?$ that runs in time $T : \omega^\omega \times \omega \to \omega$ there exists a function $L : \mathcal{B} \times \Sigma^* \to \Sigma^*$ such that*

(d): *$M^\psi(\mathbf{a})$ is **d**etermined by $L(\psi, \mathbf{a})$, that is for all $\psi \in \mathcal{B}$ and $\mathbf{a} \in \Sigma^*$*

$$L(\psi,\mathbf{a}) = L(\varphi,\mathbf{a}) \quad \Rightarrow \quad M^\psi(\mathbf{a}) = M^\varphi(\mathbf{a}).$$

(o): *The value of an* ***o****racle on a string either matters a lot or does not matter at all: If* $L(\varphi, \mathbf{b}) = L(\psi, \mathbf{b})$ *then for all* $\phi \in \mathcal{B}$

$$\{\mathbf{a} \mid \varphi(\mathbf{a}) = \psi(\mathbf{a})\} \subseteq \{\mathbf{a} \mid \varphi(\mathbf{a}) = \phi(\mathbf{a})\} \quad \Rightarrow \quad M^{\varphi}(\mathbf{b}) = M^{\phi}(\mathbf{b}).$$

(l): *The* ***l****ength of* L *can be bounded in terms of the running time* T*:*

$$|L(\psi, \mathbf{a})| \leq 2(T(|\psi|, |\mathbf{a}|) \cdot (T(|\psi|, |\mathbf{a}|) + 1) + 1).$$

Not that (o) implies (d), however, since the meaning of (d) is a lot easier to grasp and (o) is only needed to guarantee that pairings work as expected, we state them separately.

Proof. Let $L(\psi, \mathbf{a})$ be an encoding of the number of oracle queries together with a list of the $T(|\psi|\,, |\mathbf{a}|)$ first bits of the answers to the oracle calls during the run $M^{\psi}(\mathbf{a})$ of $M^?$ on input $\mathbf{a}$ with oracle ψ. I.e.

$$L(\psi)(\mathbf{a}) = \langle N, \langle \mathbf{b}_1, \ldots, \mathbf{b}_N \rangle\rangle$$

where $\mathbf{b}_i$ consist of the $T(|\psi|\,, |\mathbf{a}|)$ first bits of $\psi(\mathbf{a}_i)$ where $\mathbf{a}_i$ is the i-th of the N queries the machine asks to ψ.

As mentioned, the condition from (d) is implied by the one from (o). To see that the condition (o) holds note that the value $L(\varphi, \mathbf{b})$ determines the number N of queries the machine asks the oracle ψ and also their values $\mathbf{a}_1, \ldots \mathbf{a}_N$. Now $L(\varphi, \mathbf{b}) = L(\psi, \mathbf{b})$ implies that $\varphi(\mathbf{a}_i) = \psi(\mathbf{a}_i)$ for all i. Therefore from the other assumption of (o) it follows that the run of $M^?$ on $\mathbf{b}$ with oracle ϕ writes the same queries and gets the same answers. Thus $M^{\phi}(\mathbf{b})$ produces the same return value as both $M^{\varphi}(\mathbf{b})$ and $M^{\psi}(\mathbf{b})$.

From the restriction of the running time of $M^?$ it follows that the number N and each $|\mathbf{b}_i|$ can at most be $T(|\psi|, |\mathbf{a}|)$. This put together with the length estimations for the pairing functions from Lemma 3.2.3 leads to the bound on the length of $L(\psi, \mathbf{a})$. ∎

The conclusion $M^{\varphi}(\mathbf{b}) = M^{\phi}(\mathbf{b})$ from item (o) cannot be replaced by the stronger $L(\varphi, \mathbf{b}) = L(\phi, \mathbf{b})$. This is due to the use of initial segments of the oracle answers. The length of these initial segments depend on the value of the running time, which we have no control over.

Recall that we assigned to each $l : \omega \to \omega$ a compact subset K_l of Baire space by

$$K_l := \{\varphi \in \mathcal{B} \mid |\varphi| \leq l\}$$

and that the family $(K_l)_{l \in \omega^\omega}$ has the property that every compact subset of Baire space is contained in some K_l.

The proof of the following theorem which is the main theorem of this section is now a straightforward application of the previous proposition. Note that it does not require the metric space to be compact, but instead talks about certain relatively compact subsets of the space.

Theorem 3.2.7. *Let M be a metric space and ξ a second-order representation of M such that equality is approximable in time $T : \omega^\omega \times \omega \to \omega$. Then the set $\xi(K_l)$ has the function $n \mapsto 2(T(l,n) \cdot (T(l,n)+1)+1)$ as a modulus of total boundedness.*

Proof. Fix some n. Let $M^?$ be the machine approximating equality in time T and let $L : \mathcal{B} \times \Sigma^* \to \Sigma^*$ be the communication function assigned to $M^?$ by Proposition 3.2.6. Let I be the set of strings $\mathbf{a}$ such that there exists a $\psi \in \mathrm{dom}\{\xi\} \cap K_l$ such that $L(\langle\psi,\psi\rangle, \mathtt{1}^n) = \mathbf{a}$. For each $i \in I$ choose some $\psi_i \in K_l$ such that $L(\langle\psi_i,\psi_i\rangle, \mathtt{1}^n) = i$. From the size limit for $L(\psi, \mathtt{1}^n)$ from Proposition 3.2.6 item (l) it follows that $\mathrm{lb}(\#I) \leq 2(T(l,n) \cdot (T(l,n)+1)+1)$.

Claim that the 2^{-n}-balls around the $\xi(\psi_i)$ cover $\xi(K_l)$: Indeed, take an arbitrary $x \in \xi(K_l)$, that is $x = \xi(\psi)$ for some $\psi \in K_l$. By the definition of I there exists some $i \in I$ such that $L(\langle\psi_i,\psi_i\rangle, \mathtt{1}^n) = L(\langle\psi,\psi\rangle, \mathtt{1}^n)$. Note that

$$\langle\psi,\psi\rangle(\mathbf{b}) = \langle\psi_i,\psi_i\rangle(\mathbf{b}) \quad \Leftrightarrow \quad \psi(\mathbf{b}) = \psi_i(\mathbf{b}) \quad \Leftrightarrow \quad \langle\psi,\psi\rangle(\mathbf{b}) = \langle\psi_i,\psi\rangle(\mathbf{b}).$$

Therefore, using the property of L from item (o) of Proposition 3.2.6 for the functions $\langle\psi,\psi\rangle$, $\langle\psi_i,\psi_i\rangle$ and $\langle\psi_i,\psi\rangle$ it follows that

$$M^{\langle\psi_i,\psi\rangle}(\mathtt{1}^n) = M^{\langle\psi,\psi\rangle}(\mathtt{1}^n) = \mathtt{1}.$$

Since $M^?$ computes the function eq from Definition 3.2.4, this implies that $d(x, \xi(\psi_i)) < 2^{-n}$. Thus, $x \in B_{2^{-n}}(\xi(\psi_i))$ and, since $x \in \xi(K_l)$ was arbitrary, the 2^{-n}-balls around the images of the ψ_i cover $\xi(K_l)$. ■

Recall the singular representation of L^1 from Section 2.2. Theorem 2.2.14 proved this representation to be discontinuous, in particular the metric can not be computable, and much less so in bounded time. The previous result can be used to give an easier argument why bounded time computability is impossible:

Example 3.2.8 (The singular representation). Against better knowledge, assume that the singular representation renders the metric of L^1 computable in time bounded by T. By Theorem 3.2.7 for each fixed $l : \omega \to \omega$ the set $\xi_s(K_l)$ of functions that have some name of size l has to be totally bounded. However, the sequences constructed in the proof of discontinuity from Proposition 2.2.13 proves that the closure of this set in norm is not sequentially compact.

The generalization to discontinuity with respect to the weak topology from Theorem 2.2.15 proceeds in a similar way: It shows that the closure of the same set in the weak topology is not sequentially compact. However, since the weak topology on L^1 is not metrizable, the previous theorem is not applicable in this case.

The next corollary restates the implication item 2⇒item 1 of Theorem 3.2.2. In order to avoid to scatter the proof of the theorem too much, however, the proof of the theorem is postponed further to Section 3.2.2, page 53.

Corollary 3.2.9. *If M is a metric space and ξ is a representation of length l such that the metric is computable in time T, then*

$$|M| \in O((T(l,n)+n)^2),$$

i.e. there is a constant $C \in \omega$ such that $C(T(l,n)+n)^2 + C$ is a modulus of total boundedness of M.

Proof. To be able to apply Theorem 3.2.7, first translate the algorithm to compute the metric to an algorithm that computes the equality function. Since the equality function of the real numbers can be computed in time $O(n)$ the new algorithm runs in time $O(T(l,n)+n)$. Now applying Theorem 3.2.7 proves the assertion. ∎

In particular, any metric space that has a representation such that the constant one function is a length and the metric is computable in polynomial time is totaly bounded and of polynomial metric entropy. More generally:

Corollary 3.2.10. *If the metric of a represented metric space is polynomial-time computable, then the set of elements that have a name of length dominated by a fixed polynomial is of polynomial metric entropy.*

One implication of this is that the standard representation of continuous functions can not be expected to render the standard metric polynomial-time computable: Any Lipschitz 1 function bounded by 1 allows a name of length $n+1$, but the set of these functions has at least exponential metric entropy by Proposition 3.1.8.

Assuming openness of the representation it is possible to give lower bounds on the length of its range restrictions to compact sets from a running time and a lower bound of the size of the set it is restricted to:

Theorem 3.2.11. *Let ξ be an open representation of a metric space M such that the metric is computable in time T and let K be a compact subset of M. Then the range restriction of ξ to K has a length l that fulfills*

$$|K| \in O((T(l,n)+n)^2).$$

Proof. First note that openness is preserved under taking range restrictions (it is not preserved under restrictions, only under range restrictions). Proposition 1.2.20 proves that the range restriction as an open representation of a compact set has a length l. Running times are also preserved under range restrictions. Apply Corollary 3.2.9 and get the assertion. ∎

For an application of this see Example 4.1.5.

3.2.2 From metric entropy to complexity

This chapter investigates the opposite direction of Theorem 3.2.2. More specifically for a fixed compact metric space M a sufficient condition is given such that whenever a length function l and a running time T fulfill it, a representation of length l such that the metric is computable in time T exists. This condition only depends on the metric entropy of M.

The whole construction builds upon the following easy Lemma that can in a similar form also be found in [KLN15, Proposition 2.4]:

Lemma 3.2.12 (Uniformly dense sequence). *Whenever M is a metric space and K a subset with modulus of total boundedness μ, then there exists a sequence $(x_i)_{i\in\mathbb{N}}$ in K such that for any n the 2^{-n}-balls around the first $\sum_{i=0}^{n} 2^{\mu(i+1)} \leq 2^{\mu(n+1)+\lfloor \mathrm{lb}(n+1)\rfloor}$ elements cover K. If $K = M$, then the first $\sum_{i=0}^{n} 2^{\mu(i)} \leq 2^{\mu(n)+\lfloor \mathrm{lb}(n+1)\rfloor}$ elements suffice.*

Proof. Since μ is a metric entropy there is for any n a collection of $2^{\mu(n)}$ elements $(y_{n,i})_{i\in\{1,\ldots,2^{\mu(n)}\}}$ such that the 2^{-n}-balls around these elements cover K. If $K = M$ the y_i are elements of K and joining all the tuples in order of rising n to a sequence suffices. If $K \neq M$, the $y_{n,i}$ need not be elements of K. For each $y_{n,i}$ such that the intersection of the corresponding 2^{-n}-ball and K is not empty let $x_{n,i}$ be an element of this intersection. Since the 2^{-n+1}-ball around $x_{n,i}$ contains the 2^{-n}-ball around $y_{n,i}$, the sequence that arises by writing all these tuples after one another is as demanded. ■

Using this the main theorem of this section can be proven:

Theorem 3.2.13. *Let M be a compact metric space with metric entropy μ and let $l : \omega \to \omega$ be a length function such that there exists a time-constructible (see Definition 1.2.16) function $\tilde{T} : \omega^\omega \times \omega \to \omega$ monotone in the sense that*

$$k \leq k' \quad \Rightarrow \quad \tilde{T}(k,\cdot) \leq \tilde{T}(k',\cdot),$$

(where the inequalities have to be understood pointwise) with

$$\mu(n) \leq l(n)\tilde{T}(l,n) - \lfloor \mathrm{lb}(n+1)\rfloor. \tag{c}$$

Then there exists a second-order representation of M of length l such that the metric is computable in time $O(T)$ for

$$T(k,n) := (n+1)\cdot \tilde{T}(k,n+2)\cdot k(n+2).$$

Proof. Let $(x_i)_{i\in\mathbb{N}}$ be a sequence such that the balls of radius 2^{-n} around the first $2^{\mu(n)+\lfloor \mathrm{lb}(n+1)\rfloor}$ elements of the sequence cover M. Such a sequence exists by Lemma 3.2.12.

Define a second-order representation $\xi_{M,l,\tilde{T}}$ as follows: Let a length monotone string function φ be a $\xi_{M,l,\tilde{t}}$-name of an element $x \in M$ if and only if the following two conditions hold

(m): φ provides the values of the **m**etric on the sequence (x_i): For any $i, j \in \mathbb{N}$ the string $\mathbf{a}$ consisting of the first bit of each of the strings

$$\varphi(\mathtt{0}\mathtt{1}^{n+1}\mathtt{0}\langle i,j\rangle), \ldots, \varphi(\mathtt{0}^{n+1}\mathtt{1}^{n+1}\mathtt{0}\langle i,j\rangle)$$

fulfills

$$\left| d(x_i, x_j) - \frac{\nu_{\mathbb{N}}(\mathbf{a})}{2^n} \right| \leq 2^{-n}.$$

(i): φ provides **i**ndices of approximations: For the concatenation $\mathbf{a}$ of the $N := \tilde{T}(|\varphi|, n)$ strings $\mathbf{m}_0, \ldots, \mathbf{m}_N$ where $\mathbf{m}_i$ are the $|\varphi|(n)$ first bits of $\varphi(\mathtt{1}\langle \mathtt{1}^n, i\rangle)$ it holds that

$$d(x_{\nu_{\mathbb{N}}(\mathbf{a})}, x) \leq 2^{-n}.$$

This defines a second-order representation:

For any two distinct elements $x, y \in M$ there is an n such that $2^{-n} < d(x, y)$. Thus, if x_l resp. x_m are 2^{-n-1}-approximations of x resp. y, then $l \neq m$. Now assume that φ and ψ fulfill the conditions to be names of x resp. y. Then the strings $\mathbf{a}$ from (i) must differ. Thus, $\xi_{M,l,\tilde{T}}$ is single-valued.

The choice of the sequence $(x_i)_{i\in\mathbb{N}}$ and (c) make sure that condition (i) can be fulfilled by a function of length l, thereby leaving enough freedom in the choice of the function to make it also fulfill (m). Thus, $\xi_{M,l,\tilde{T}}$ has length l and is in particular surjective.

It is left to provide an appropriate algorithm for computing the metric. When given a name $\langle \varphi, \psi\rangle$ of some element $(x, y) \in M \times M$ as oracle this algorithm proceeds as follows: Note that the values of $|\varphi|$ and $|\psi|$ can be computed from $\langle \varphi, \psi\rangle$ in time $O(|\langle \varphi, \psi\rangle|(n))$ by using the projections. Thus, from the time-constructibility of $\tilde{T}$ it follows that

$$N := \tilde{T}(|\varphi|, n) \quad \text{and} \quad \tilde{N} := \tilde{T}(|\psi|, n)$$

can be computed in time $O(T)$. Without loss of generality assume $N \geq \tilde{N}$ for the following. Next the machine queries the oracle N times for $\mathtt{1}\langle n+2, i\rangle$, with the values of i going from 1 to N. Each time it writes copies of the first $|\varphi|(n+2)$, resp. $|\psi|(n+2)$ first bits of each of the values $\varphi(\mathtt{1}\langle \mathtt{1}^{n+2}, i\rangle)$ resp. $\psi(\mathtt{1}\langle \mathtt{1}^{n+2}, i\rangle)$ to separate parts of the memory band (the second projections are dismissed after $i \geq \tilde{N}$). This takes time less than $O(N \cdot |\langle \varphi, \psi\rangle|(n+2))$. In the end there are codes $\mathbf{a}$ and $\mathbf{b}$ of length $N \cdot |\varphi|(n+2)$ resp. $\tilde{N} \cdot |\psi|(n+2)$ of indices i, j of 2^{-n-2}-approximations x_i, x_j to x and y written to the memory band. These can be translated to an encoding of $\langle i, j\rangle$ in time $O(\max\{N\,|\varphi|(n+2), \tilde{N}\,|\psi|(n+2)\})$.

Finally, the machine queries the oracle $n+1$ times on the inputs $\mathtt{0}^m\mathtt{1}^{n+2}\mathtt{0}\langle i, j\rangle$ for $m \in \{1, \ldots, n+2\}$ and copies the first bits of the first projections of the results to the answer tape. Most time is consumed by writing the queries which takes time less than

$$O(n \cdot \max\{N \cdot |\varphi|(n+2), \tilde{N} \cdot |\psi|(n+2), n\}). \tag{O}$$

Using the triangle inequality and (m) one verifies that the result leads to a valid dyadic approximation to $d(x, y)$ being written on the output tape.

To obtain the running time bound note, that due to the monotonicity of $\tilde{T}$

$$\max\{N(|\varphi|(n+2)), \tilde{N}(|\psi|(n+2))\} \leq \tilde{T}(|\langle\varphi,\psi\rangle|, n+2)\,|\langle\varphi,\psi\rangle|(n+2).$$

Since the term of the running times specified in (O) majorizes all the others, it follows that the whole procedure can be carried out in the suggested time bound. ∎

There is no computability condition on the compact metric space whatsoever. Due to the time-constructibility of the function $\tilde{T}$, the function $\tilde{T}(l,\cdot)$ is computable from l. Thus, if the metric entropy grows faster than any computable function, then the length function cannot be computable. The constructed representation depends heavily on what uniformly dense sequence is chosen via Lemma 3.2.12. With respect to the second-order representation constructed, the elements of the chosen sequence need not have computable names: It might be impossible to choose a computable oracle for the distance function that needs to be included in each name. In particular it can not be guaranteed that the representation has any computable names.

The proof of Theorem 3.2.2 is now a straightforward application of the previous Theorem 3.2.13 and Corollary 3.2.9.

Proof (of Theorem 3.2.2).

2. ⇒ ***1.:*** This implication claims that if there is a second-order representation of length l such that the metric is computable in time T and $T(l,\cdot) \in \mathcal{A}$, then there is a modulus of total boundedness $\mu \in \mathcal{A}$. By Corollary 3.2.9 there exists a $C \in \omega$ such that the function $\mu(n) := C(T(l,n)+n)^2 + C$ is a modulus of total boundedness of the metric space. From $T(l,\cdot) \in \mathcal{A}$ and the closure of $\mathcal{A}$ under application of polynomials it follows that $\mu \in \mathcal{A}$.

1. ⇒ ***2.:*** This implication claims that if M has a modulus of total boundedness $\mu \in \mathcal{A}$, then there is a representation of finite length l such that the metric is computable in time T with $T(l,\cdot) \in \mathcal{A}$. To see this, first note that since $\mathcal{A}$ is closed under application of polynomials $n + \mu(n) \in \mathcal{A}$. Thus, assume $\mu(n) \geq n$ and use Theorem 3.2.13 with $l := \mu$. $\tilde{T}(k,n) := n+1$ is obviously time constructible and monotone, thus the theorem proves that there is a constant $C \in \omega$ such that

$$T(k,n) := C(k(n+2)+n+3)^3 + C \geq C(n+1)(n+3)k(n+2) + C$$

is a running time. Since $T(l,n) = C(\mu(n+2)+n+3)^3 + C$ it follows from the closure of $\mathcal{A}$ under shift and application of polynomials that $T(l,\cdot) \in \mathcal{A}$.

This concludes the proof. ∎

3.3 Variations of the results

This section discusses two versions of the previous results of this chapter in slightly different frameworks. The first one is the framework of representations in the sense of Weihrauch and should be regarded as an application. The second version replaces time-bounded computation by space-bounded computation. This should be considered as a refinement of the results.

3.3.1 Cantor space representations

The framework introduced in Section 1.2 is most appropriate for investigating complexity theory on a general class of spaces. However, there is another approach to computation over continuous structures by means of stream manipulations: The framework of representations in the sense of Weihrauch. This approach is computably equivalent and the more popular model for doing computability theory. The complexity theory induced by this model is only usable for a restricted class of spaces. Since the results of the previous chapter are strongly connected to the restrictions of the model, we invest some time to elaborate. More information and a proper introduction can be found in [Wei00].

Recall that **Cantor space** is the space $\mathcal{C} := \Sigma^\omega$ of infinite binary strings. Denote infinite strings by $\chi, \chi', \ldots$ and the n-th digit of a string χ by χ_n. The elements of Cantor space are binary strings, however, it is often useful to have more symbols available. Additional symbols can be simulated by binary strings without changing any of the complexity theoretical considerations carried out here.

Definition 3.3.1. A **Cantor space representation** of a space X is a partial surjective mapping $\delta : \mathcal{C} \to X$.

Again, we call the elements of $\delta^{-1}(x)$ the names or δ-names of x.

Identifying letters with one letter words and completing to a total function in a straightforward way gives a canonical inclusion

$$i : \mathcal{C} \to \Sigma^{**}, \quad i(\chi)(\mathbf{a}) = \chi_{|\mathbf{a}|}$$

of Cantor space into the length-monotone functions. The function mapping a string function to the string of the first digits of its values on $\mathtt{1}^n$ is a retraction of i and we denote it by π. For any Cantor space representation δ, $\delta \circ \pi$ restricts to a second-order representation of length one. Thus, the content of Section 1.2 can be used to define a computability and complexity of functions between Cantor space represented spaces.

This translation is one way only: A second-order represented space $\mathbf{X}$ such that no element has names of sub-exponential length can not be translated to a Cantor space representation while preserving the notion of polynomial-time computability of functions from $\mathbf{X}$ to $\mathbb{N}$. Even second-order representations such that the constant one function is a length are more general than Cantor space representations. The binary input allows for more densely packed information: For instance the space $L_{1,0}$ of Lipschitz one functions on the unit interval that map zero to zero allows a second-order representation of length one such that evaluation is polynomial-time computable. Such a Cantor space representation does not exist.

The naïve approach to complexity of functions between Cantor space represented spaces often leads to problems. For example consider the following Cantor space representation: Let χ be a name of $x \in \mathbb{R}$ if and only if

$$\chi = \mathbf{a}_0\#\mathbf{a}_1\#\ldots \quad \text{such that for all } n \in \mathbb{N} \quad \left|\frac{\nu_{\mathbb{N}}(\mathbf{a}_n)}{2^n} - x\right| < 2^{-n}.$$

With respect to this representation, any computable function from $\mathbb{R}$ to $\mathbb{R}$ is linear time computable: Assume M is a machine computing the function. A machine computing the same function in linear time can be defined as follows: As long as M does not produce a # it uses every second step to simulate the machine M and every other step to output a zero. Whenever M outputs a # it copies the string produced between this # and the one before to the output band and follows it with a #. Since leading zeros do not change the encoded number, this machine computes the same function. Since a symbol is produced every second step, it runs in linear time. Note the similarity to the arguments from the proof of Proposition 2.3.2.

The following definition from [Sch04] circumvents these complications:

Definition 3.3.2. A Cantor space representation is called **proper** if the pre-image of any compact set under the representation is a compact subset of Cantor space.

Obviously, this excludes the above example, as the pre-image of one point sets under the representation is not compact. Schröder also succeeds to show that those spaces allowing a well-behaved proper representation are exactly the separable metric spaces [Sch04, Theorem 5.3].

We need the representation Schröder uses to prove this:

Definition 3.3.3. Let $\mathcal{M} := ((M,d),(x_i)_{i\in\mathbb{N}})$ be a metric space together with a dense subsequence. Define the **standard Cantor space representation** $\delta_{\mathcal{M}}$ of M by letting an infinite string $\chi \in \mathcal{C}$ be a name of an element $x \in X$ if and only if $\chi = m_0\#m_1\#\cdots$ such that

$$\forall n : d(x, x_{m_n}) \leq 2^{-n} \quad \text{and} \quad m < m_n \Rightarrow d(x, x_m) \geq 2^{-n-1}.$$

This Cantor space representation is continuous and proper.

For elements $\chi, \chi' \in \mathcal{C}$ define the pairing $\langle \chi, \chi' \rangle \in \mathcal{C}$ by

$$\langle \chi, \chi' \rangle_n := \begin{cases} \chi_m & \text{if } n = 2m \\ \chi'_m & \text{if } n = 2m+1. \end{cases}$$

That is:

$$\langle \chi, \chi' \rangle = \chi_0 \chi'_0 \chi_1 \chi'_1 \cdots$$

This can be used to define products of Cantor space representations and Cantor space represented spaces.

The construction is well behaved with respect to the product introduced for second-order representations and the inclusion of the Cantor space representations into the second-order representations. I.e. for Cantor space representations δ and δ' it holds that $(\delta \circ \pi) \times (\delta' \circ \pi)$ and $(\delta \times \delta') \circ \pi$ are polynomial time-equivalent as second-order representations. In particular this can be used to define computations with respect to an oracle that takes binary input: Say $f : \mathcal{C} \to \mathcal{C}$ is computable resp. polynomial-time computable with oracle $\phi : \Sigma^* \to \{\mathtt{0}, \mathtt{1}\}$, if there is an oracle machine that computes $i(f(\chi))$ on oracle $\langle i(\chi), \phi \rangle$. The same notion of oracle computability arises if the TTE machine is equipped with an additional query tape.

We aim to prove that the following version of Theorem 3.2.2 for Cantor space representations holds:

Theorem 3.3.4. *A metric space has a polynomial modulus of total boundedness if and only if it allows a Cantor space representation such that the metric is computable in polynomial-time with respect to an oracle.*

For a Cantor space representation $\delta : \mathcal{C} \to X$ and an oracle $\phi : \Sigma^* \to \{\mathtt{0}, \mathtt{1}\}$

$$\xi_{\delta,\phi}^{-1}(x) := \{\langle i(\chi), \phi\rangle \mid \chi \in \delta^{-1}(x)\}$$

defines a second-order representation of constant length such that a polynomial-time computability relative to ϕ with respect to δ is translated to polynomial-time computability with respect to $\xi_{\delta,\phi}$. Thus, one direction of Theorem 3.3.4 follows directly from Theorem 3.2.7 (or more easily from Corollary 3.2.9):

Corollary 3.3.5. *Let M be a metric space for which there exists a Cantor space representation such that the metric is polynomial-time computable. Then M has a polynomial modulus of total boundedness.*

Note that for the opposite direction Theorem 3.2.13 can not be applied directly: On input of a polynomial μ a representation of length one is produced. However, it is unclear how to translate this representation into a Cantor space representation while maintaining the polynomial-time computability of the metric. Thus, we give another construction that proves a statement a little more general than the one aimed for:

Theorem 3.3.6. *Let M be a metric space of metric entropy μ. There exists a Cantor space representation of M such that the metric is computable in time $O(n(\mu(n+3) + \mathrm{lb}(n+1)))$ with respect to some oracle.*

Proof. Let $(x_i)_{i\in\mathbb{N}}$ be one of the uniformly dense sequences constructed in Lemma 3.2.12. That is: The balls of radius 2^{-n} around the first $2^{\mu(n)+\lfloor \mathrm{lb}(n+1)\rfloor}$ elements of the sequence cover M.

Let ρ be the standard Cantor space representation of $((M, d), (x_i))$ according to Definition 3.3.3, that is: An infinite string χ is a name of an element $x \in M$ if and only if it is of the form $\chi = m_0\#m_1\#\cdots$ with $m_i \in \mathbb{N}$ such that

$$\forall n \in \omega : \left(d(x_{m_n}, x) \leq 2^{-n} \quad \text{and} \quad \forall m < m_n : d(x_m, x) \geq 2^{-n-1}\right).$$

Let the oracle $\phi : \Sigma^* \to \{\mathtt{0}, \mathtt{1}\}$ be a function fulfilling the condition (m) in the proof of Theorem 3.2.13.

To complete the proof we describe an oracle TTE machine M that computes the metric within the specified time bound and when given ϕ as oracle: Assume M is given an intertwining $\langle\chi, \chi'\rangle$ of names χ and χ' of elements x and x' on its infinite read only input tape. For each $n = 1, 2, \ldots$ the machine proceeds as follows: It skips to the first digit of an index of an 2^{-n-2} approximation of x. To see that this can be done in time $O(n\cdot(\mu(n+3)+\mathrm{lb}(n)))$ claim that $|m_i| \leq \mu(i+1)+\lfloor \mathrm{lb}(i+2)\rfloor$ for any index m_i that shows up in a name: The sequence is such that the 2^{-i-1}-balls around indices of size up to this cover the space and Definition 3.3.3 of the standard Cantor space representation ρ forbids bigger indices to be used. Then the machine produces an intertwining $\mathbf{k}$ of indices of

2^{-n-2}-approximations of x and x' padded to the same length on the memory tape in time $O(\mu(n+2)+\mathrm{lb}(n))$. Next it queries the oracle ϕ on the inputs $\mathtt{0}^0\mathtt{1}^{n+3}\mathtt{0}\mathbf{k}, \ldots, \mathtt{0}^{n+3}\mathtt{1}^{n+3}\mathtt{0}\mathbf{k}$. And copies the answer bits to the answer tape. It ends the procedure for n by writing a separator # and carries on with $n+1$. By definition of the oracle ϕ the answers k_i written on the answer tape after one another are an encodings of a dyadic 2^{-n-2}-approximations of $d(x_i, x_{i'})$. By the triangle inequality these approximations are 2^{-n}-approximations to $d(x, x')$. Therefore, the machine computes a name of $d(x, y)$ in the standard Cantor space representation of $\mathbb{R}$ with respect to the standard enumeration of dyadics in time $O(n(\mu(n+3) + \mathrm{lb}(n+1)))$. ∎

The proof of Theorem 3.3.4 is immediate from the previous two results.

3.3.2 Space-bounded computation

This chapter discusses an improvement of Theorem 3.2.2 from time-bounded computation to space-bounded computation. Roughly the goal is to replace the occurrences of computation that take time T by computations that use space $\mathrm{lb}(T)$. To state this, consider a stable class $\mathcal{A}$. (That is: $\mathcal{A}$ is closed under application of polynomials and shift, see Definition 3.2.1) and denote by $\mathrm{lb}(\mathcal{A})$ the set of functions that are majorized by a function of the form $n \mapsto \lceil \mathrm{lb}(\mu(n) + 1) \rceil$ where μ is an element of $\mathcal{A}$.

The goal of this chapter is to prove the following:

Theorem 3.3.7. *Let M be a compact metric space and let $\mathcal{A} \subseteq \omega^\omega$ be a stable class. The following are equivalent:*

1. *M has a modulus of total boundedness $\mu \in \mathcal{A}$.*
2. *There exists a representation of M that has length l such that the metric can be computed in space S and $S(l, \cdot) \in \mathrm{lb}(\mathcal{A})$.*

The proof of this theorem can be found on page 61.

Proving this theorem requires a notion of space complexity. Given a function $S : \omega^\omega \times \omega \to \omega$, we want to restrict the memory available for a computation on input $\mathbf{a}$ with oracle ψ to use at most $S(|\psi|, |\mathbf{a}|)$ memory cells. It is not immediately clear what should be counted as a memory cell: Obviously the input tape should not. How about the oracle query tape? For the identity on Baire space to be logarithmic space computable, it is necessary to exempt the oracle query tape from the limitation. To avoid cheating, the query tape has to be divided into a write only query tape and a read only answer tape. The model of computation obtained in this way is too powerful as it allows polynomial depth iterated application (for details see [KO14]).

The adequate model of space-bounded computation turns out to be a machine that has a stack of unrestricted write only query tapes. Asking a query means popping the top element of the stack and writing the answer to an single unrestricted read only answer tape. To disallow iterated application of polynomial depth, the answer tape has to be erased each time a new query tape is pushed. Thus the machine is allowed to incorporate a polynomial number of digits from a previous query to form the next query. To make sure that any polynomial space computable function is computable in bounded time, it is necessary to restrict the height of the stack. The stack is required to be finite.

This model was originally introduced in [Wil88], is accepted as model for small complexity classes [ACN07] and is used in computable analysis [KO14] and for bigger complexity classes [Bus88].

Definition 3.3.8. Let $S : \omega^\omega \times \omega \to \omega$ be a function. An oracle stack Turing machine M is said to **use space** S if for all inputs $\mathbf{a}$ and oracles φ the computation $M^\varphi(\mathbf{a})$ uses less than $S(|\varphi|, \mathbf{a})$ cells of the memory tape and there exists a $C \in \omega$ such that it takes no more than $2^{CS(l,n)+C}$ time steps. It is said to **use space** $O(S)$ if there is a constant $C \in \omega$ such that it uses space $CS + C$.

The model of computation is carefully chosen such that the usual inclusions of complexity classes hold. This includes the additional bound on the running time of the machine. It is not known to the author if this additional assumption is necessary. It is for instance also included in the definitions in [ACN07].

The following is a straightforward application of the model:

Example 3.3.9. The composition

$$\circ : \mathcal{B} \times \mathcal{B} \to \mathcal{B}, \quad (\varphi, \psi) \mapsto \varphi \circ \psi$$

is computable using space $S(l, n) = \lceil \mathrm{lb}(\max\{l(n), n\} + 1) \rceil$.

A machine computing the composition from the oracle $\langle \varphi, \psi \rangle$ and a string $\mathbf{a}$ while obeying both the space bound and the time bound imposed by the definition might work as follows: First it pushes a query tape. This query tape is supposed to contain $\psi(\mathbf{a})$ in the end. Then it pushes another query tape and copies $\mathbf{a}$ to the upper tape. It proceeds to pop the second tape to obtain $\psi(\mathbf{a})$ and copies the answer from the answer tape to the first query tape. Then it pops the first query tape and copies the answer to the output tape. The only space that is consumed at all is by a counter that is needed to keep track of which digit of $\mathbf{a}$ resp. $\psi(\mathbf{a})$ to copy next. This counter needs to count up to the maximum of $|\psi(\mathbf{a})|$ and $|\mathbf{a}|$, which is possible within the space limit.

The stack of height two seems to be indispensable: Since asking the query means popping the stack and pushing a new query tape erases the answer tape, a stack height of one does not suffice. More generally, concatenating m functions requires a stack of height m.

Again, we need a constructibility condition:

Definition 3.3.10. A function $S : \omega^\omega \times \omega \to \omega$ is called **space-constructible** if there is an oracle stack Turing machine $M^?$ that uses space $O(S)$ and on input $\mathbf{a}$ with oracle φ stops with $S(|\varphi|, |\mathbf{a}|)$ written on the output tape.

The additional assumption about the running time guarantees that Theorem 3.2.7 can be applied and Corollary 3.2.9 can be restated as:

Corollary 3.3.11. *If M is a compact metric space that allows a representation of length l such that the metric is computable in space S, then*

$$\lceil \mathrm{lb}(|M|+1)\rceil \in O(S(l,n))$$

This results in a proof of the implication item 2⇒item 1 of Theorem 3.3.7 (to be carried out on page 61).

For the other direction we produce a logarithmic space version of Theorem 3.2.13.

Theorem 3.3.12. *Let M be a compact metric space with metric entropy μ. Let $l:\omega\to\omega$ be a length function such there exists a space constructible $\tilde{S}:\omega^\omega\times\omega\to\omega$ monotone in the sense that*

$$k\le k' \quad\Rightarrow\quad \tilde{S}(k,\cdot)\le\tilde{S}(k',\cdot)$$

(the inequalities should be read pointwise) such that

$$\mu(n)\le l(n)2^{\tilde{S}(l,n)}-\mathrm{lb}(n+1). \tag{c}$$

Then there exists a second-order representation of M of length l such that the metric is computable in space $O(S)$ for

$$S(k,n):=\tilde{S}(k,n+2)+\lceil\mathrm{lb}(k(n+2)+1)\rceil+\lceil\mathrm{lb}(n+1)\rceil.$$

Proof. Let $(x_i)_{i\in\mathbb{N}}$ be a sequence such that the balls of radius 2^{-n} around the first $2^{\mu(n)+\lfloor\mathrm{lb}(n+1)\rfloor}$ elements of the sequence cover M. Such a sequence exists by Lemma 3.2.12.

Now define a second-order representation $\xi_{M,l,\tilde{S}}$ as follows: Let a length monotone string function φ be a name of an element $x\in M$ if and only if the following two conditions hold:

(m): φ provides the values of the **m**etric on the sequence (x_i): For any strings $\mathbf{a}$ and $\mathbf{b}$ the string $\mathbf{c}$ consisting of the first bit of each of the strings

$$\varphi(\mathtt{0}\mathtt{1}^{n+1}\mathtt{0}\langle\mathbf{a},\mathbf{b}\rangle),\ldots,\varphi(\mathtt{0}^{n+1}\mathtt{1}^{n+1}\mathtt{0}\langle\mathbf{a},\mathbf{b}\rangle)$$

fulfills

$$\left|d(x_{\nu_{\mathbb{N}}(\mathbf{a})},x_{\nu_{\mathbb{N}}(\mathbf{b})})-\frac{\nu_{\mathbb{N}}(\mathbf{c})}{2^n}\right|\le 2^{-n}.$$

(i): φ provides **i**ndices of approximations: Whenever $\mathbf{a}$ is the concatenation of the $N:=2^{\tilde{S}(|\varphi|,n)}$ strings $\mathbf{m}_0,\ldots,\mathbf{m}_N$, where $\mathbf{m}_i$ are the $|\varphi|(n)$ first bits of $\varphi(\mathtt{1}\langle\mathtt{1}^n,i\rangle)$ it holds that

$$d(x_{\nu_{\mathbb{N}}(\mathbf{a})},x)\le 2^{-n}.$$

This defines a second-order representation:

Any two distinct elements $x,y\in M$ are of positive distance $d(x,y)>0$. Therefore, whenever $2^{-n}<d(x,y)$ and x_l resp. x_m are 2^{-n-1} approximations of x resp. y, then $l\neq m$. Now assume that φ and ψ fulfill the conditions to be names of x resp. y. Then the strings $\mathbf{a}$ from (i) must differ. Thus, $\xi_{M,l,\tilde{S}}$ is single-valued.

The choice of the sequence (x_i) and Equation (c) make sure that condition (i) can be fulfilled by a function of length l, thereby leaving enough freedom in the choice of the function to make it also fulfill (m). The assumption from (c) and the monotonicity of $\tilde{S}$ guarantee that it is possible to pad the function to fulfill (l) without making it longer than l. Thus, $\xi_{M,l,\tilde{S}}$ has length l and is in particular surjective.

It is left to describe an oracle stack Turing machine that computes the metric. When given a name $\langle \varphi, \psi \rangle$ of some element $(x, y) \in M \times M$ this machine proceeds as follows: Let $\mathtt{1}^n$ be the precision requirement the machine is given since the values of $|\varphi|$ and $|\psi|$ can be read from those of $\langle \varphi, \psi \rangle$ and due to the space-constructibility of $\tilde{S}$, the machine can write binary encodings N of $2^{\tilde{S}(|\varphi|,n+2)}$ and $\tilde{N}$ of $2^{\tilde{S}(|\psi|,n+2)}$ to the memory band while not using more space than $O(\tilde{S})$: The return value of $\tilde{S}(|\langle \varphi, \psi \rangle|, n+2)$ is in unary. It also saves $|\varphi|(n+2)$ and $|\psi|(n+2)$ in binary. Due to the monotonicity of $\tilde{S}$ all of this uses space $O(\tilde{S}(|\langle \varphi, \psi \rangle|, n+2) + \lceil \mathrm{lb}(|\langle \varphi, \psi \rangle|(n+2)+1) \rceil)$.

W.l.o.g assume $N \geq \tilde{N}$ for the following. Next the machine carries out the following loop for j going from 1 to $n+3$: It writes a $\mathtt{0}^j \mathtt{1}^{n+2}$ to the lowest oracle band. Then it carries out the following loop for i going from 1 to N: it pushes a query tape, writes $\mathtt{1}\langle \mathtt{1}^{n+2}, i \rangle$ onto it and queries the oracle. It copies the first $|\varphi|(n+2)$ even bits to the first empty even positions of the lowest query band of the stack, checks if $i \leq \tilde{N}$ and if so also copies the first $|\psi|(n+2)$ bits of the odd positions onto the first empty odd positions. When the loop over i is done it fills the holes on the query band to make the content an encoding of $\mathtt{0}^{\tilde{i}} \mathtt{1}^{n+3} \mathtt{0} \langle \mathbf{a}, \mathbf{b} \rangle$ where $\mathbf{a}$ and $\mathbf{b}$ are strings such that $x_{\nu_{\mathbb{N}}(\mathbf{a})}$ and $x_{\nu_{\mathbb{N}}(\mathbf{b})}$ are 2^{-n-2}-approximations to x and y. It issues the query of the lower band and copies the first bit of the answer to the answer tape. When the loop over j is also done a 2^{-n-1}-approximation to $d(x_i, x_j)$ is written on the answer tape. By the triangle inequality this is a 2^{-n}-approximation to $d(x, y)$.

From the description it is clear that the machine only uses space $O(S)$: The term $\tilde{S}(k, n+2)$ is necessary for saving and counting up to N and $\tilde{N}$ (it is reused several times), the term $\lceil \mathrm{lb}(k(n+2)+1) \rceil$ for saving $|\varphi|(n+2)$ and $|\psi|(n+2)$ and $\lceil \mathrm{lb}(n+1) \rceil$ counting up to n in the outer loop. The additional time restriction needs also to be checked: The inner loop takes time $O(N \cdot (N + n + |\langle \varphi, \psi \rangle|(n+2)))$ and is carried out n times. With $N = 2^{\tilde{S}(l,n)} \leq 2^{S(l,n)}$ the time of computation depends on the bound exponentially, but this is allowed. ∎

Example 3.3.13. Consider the space $L_{1,0}$ of Lipschitz 1 functions that are zero in zero with the metric induced by the supremum norm. By Proposition 3.1.8 this space has metric entropy $\mu(n) = \lceil \mathrm{lb}(3) 2^n \rceil$. Setting $l(n) := 1$ and $\tilde{S}(k, n) := n + \lceil \mathrm{lb}(n+1) \rceil + 1$, the previous theorem provides us with a representation of constant length such that the metric is computable using space $O(S)$ for

$$S(k, n) = n + \lceil \mathrm{lb}(k(n+2)+1) \rceil + \lceil \mathrm{lb}(n+1) \rceil.$$

Another choice would be to set $l(n) := \lceil \mathrm{lb}(\mu(n)) \rceil$ and $\tilde{S}(k, n) = k(n)$. This changes the space bound to $S(k, n) = k(n+2) + \lceil \mathrm{lb}(n+1) \rceil$.

Of course we know such a representation already: The restriction of the standard representation of continuous functions. If the sequence $(x_i)_{i \in \mathbb{N}}$ is chosen to be a standard enumeration of piecewise linear functions, both of the previous choices lead to representations that are polynomial-time equivalent to the range restriction of the standard representation.

Proof (of Theorem 3.3.7). First prove *2.⇒1.*: So assume that we are given a representation of length l such that the metric is computable using space S and $S(l,\cdot) \in \mathrm{lb}(\mathcal{A})$. That is $S(l,n) = \lceil \mathrm{lb}(\mu(n)+1)\rceil$ for some $\mu \in \mathcal{A}$ where $\mathcal{A}$ is a stable class. We need to specify a modulus of total boundedness of M from $\mathcal{A}$. Corollary 3.3.11 provides a constant $C \in \omega$ such that

$$\lceil \mathrm{lb}(|M|(n)+1)\rceil \leq CS(l,n) + C = C\lceil \mathrm{lb}(\mu(n)+1)\rceil + C.$$

Take the exponent on both sides and use the inequalities $k \leq 2^{\lceil \mathrm{lb}(k+1)\rceil} \leq 2(k+1)$ to get

$$|M|(n) \leq 2^{2C}(\mu(n)+1)^C \leq 2^{2C}(\mu(n)+n+1)^C$$

The right hand side is an element of $\mathcal{A}$ due to the closure of $\mathcal{A}$ under application of polynomials.

Next prove *1.⇒2.*. That is: Whenever a compact metric space M allows a modulus of total boundedness μ from a stable set $\mathcal{A}$, then there exists a representation of M that has finite length l and such that the metric can be computed in space T such that $T(l,\cdot) \in \mathrm{lb}(\mathcal{A})$. First note that since $\mathcal{A}$ is closed under application of polynomials $n + \mu(n) \in \mathcal{A}$ and it can be assumed that $\mu(n) \geq n$. Use Theorem 3.3.12 with $l(n) := \lceil \mathrm{lb}(\mu(n)+1)\rceil$ and $\tilde{S}(k,n) = k(n)$. The assumptions can easily be verified and the theorem thus provides a representation of length l such that the metric can be computed in space $O(S)$ for

$$T(k,n) = k(n+2) + \mathrm{lb}(n+1).$$

Now $T(l,n) \in \mathrm{lb}(\mathcal{A})$ follows from the closure of $\mathcal{A}$ under shift and polynomial application: The function $(\mu(n+2)+n+1)^2$ is an element of $\mathcal{A}$ and $T(l,n) \leq \lceil \mathrm{lb}((\mu(n+2)+n+1)^2+1)\rceil$. ∎

Note that in contrast to the proof of Theorem 3.2.13 from page 53 we deliberately chose the representation of reasonably small length instead of the representation of maximal reasonable length.

4 Arzelà-Ascoli and Fréchet-Kolmogorov

The previous chapter investigated represented metric spaces. It provided a concrete connection between the computational complexity of the metric and the sizes of a family of relatively compact sets: The images of the restrictions of the representation to the compact subsets K_l of the Baire space consisting of the string functions of length bounded by $l : \omega \to \omega$. For the standard representation of continuous functions it is known that this is a family of relatively compact sets. Or stated in other words: any equicontinuous, bounded set of continuous functions is relatively compact. In analysis the above relation is known as the Arzelà-Ascoli Theorem, and the results of the previous section can be applied to produce quantitative versions of this result from bounds of the running time or the space used in the computation of the metric in the standard representation.

This chapter first recalls the Arzelà-Ascoli Theorem, presents a quantitative version that originates from [Tim94] and uses it to demonstrate the connections to the content of the previous section. Similar bounds were implicitly contained in earlier works from constructive analysis (see for instance [BB85]). Then it turns to L^p spaces: A similar classification result of the compact subsets of L^p spaces is known as the Fréchet-Kolmogorov Theorem. The appropriate notion of a modulus is introduced and then provide a quantitative version of the Fréchet-Kolmogorov Theorem is proven. The Theorem generalizes a results from [Lor66] and to the knowledge of the author has not been stated in this generality before.

Quantitative refinements of both the Arzelà-Ascoli and Fréchet-Kolmogorov Theorems have been investigated before in different contexts: There has been extensive work on these topics in Approximation Theory. These results cannot straightforwardly be transferred to the context of this thesis since the notion of moduli considered differs by convention. As a result, the theorems usually use the inverse modulus instead of the modulus itself. Furthermore, the results are often only stated or valid for small moduli.

4.1 Arzelà-Ascoli

Recall the Arzelà-Ascoli Theorem from analysis (for instance [Wer00, Satz II.3.4]):

Theorem 4.1.1 (Arzelà-Ascoli). *A subset of $\mathcal{C}([0,1])$ is relatively compact if and only if it is bounded and equicontinuous.*

Equicontinuity of a subset of $\mathcal{C}([0,1])$ is equivalent to the existence of a common modulus of continuity of all of its elements. Thus, this theorem provides a direct connection between compactness of a set of functions and their moduli of continuity.

A quantitative refinement can be formulated by means of families of compact sets.

Definition 4.1.2. The family $(K^{\infty}_{l,C})_{l\in\omega^\omega,C\in\omega}$ of **Arzelà-Ascoli sets** $K^{\infty}_{l,C} \subseteq \mathcal{C}([0,1])$ is defined by

$$K^{\infty}_{l,C} := \left\{ f \in \mathcal{C}([0,1]) \mid l \text{ is a modulus of continuity of } f \text{ and } \|f\|_\infty \leq 2^C \right\}.$$

The classical Arzelà-Ascoli Theorem (Theorem 4.1.1) states that a set of functions is relatively compact if and only if it is contained in some $K^{\infty}_{l,C}$.

Example 4.1.3 (Running times and Arzelà-Ascoli). Define a representation $\tilde{\xi_{\mathcal{C}}}$ of $\mathcal{C}([0,1])$ as follows: A length monotone string function φ is a $\tilde{\xi_{\mathcal{C}}}$-name of $f \in \mathcal{C}([0,1])$ if and only if

- $|\varphi|(0)$ is an upper bound to $\mathrm{lb}(\|f\|_\infty)$.
- Whenever $\mathbf{a}$ is an encoding of a dyadic number, then for the first $2(\varphi(0)+n+1)$ bits $\mathbf{b}$ of $\varphi(\langle \mathbf{a}, 1^n\rangle)$ it holds that $|[\![\mathbf{b}]\!] - f([\![\mathbf{a}]\!])| < 2^{-n}$.
- $|\varphi|$ is a modulus of continuity of f.

This defines a representation: Any number less than $\|f\|_\infty$ has dyadic 2^{-n}-approximations with encodings of length less than $2(\mathrm{lb}(\|f\|_\infty)+n+1)$. Furthermore the defined representation is polynomial-time equivalent to the standard representation. With respect to $\tilde{\xi_{\mathcal{C}}}$, however, it is possible to evaluate a function in a time that does not iterate the length of oracle encoding both the number and the function.

Recall that K_l denotes the set of elements of the Baire space whose length is bounded by l. These sets are compact. We have

$$\tilde{\xi_{\mathcal{C}}}(K_l) \supseteq K^{\infty}_{l,\lceil l(0)/2\rceil-3}.$$

The requirement from the first bullet makes sure that we have enough space to encode the dyadic approximations of the function values. Corollary 3.2.9 is applicable to the range restriction of $\xi_{\mathcal{C}}$ to $\xi_{\mathcal{C}}(K_l)$, and whenever T is a running time of the metric, then there exists a constant $C \in \omega$ such that

$$\left|K^{\infty}_{l,\lceil l(0)/2\rceil-3}\right|(n) \leq \left|\tilde{\xi_{\mathcal{C}}}(K_l)\right|(n) \leq C(T(l,n)+n)^2 + C.$$

The straightforward algorithm for computing the metric runs in time $O(T)$ for

$$T(l,n) = (l(0)+n)2^{l(n+2)}.$$

Thus, Corollary 3.2.9 proves that

$$\left|K^{\infty}_{l,\lceil l(0)/2\rceil)-3}\right| \in O((l(0)+n)^2 2^{2l(n+2)} + n).$$

Direct methods lead to a slightly better bound and additionally provide a lower bound: The proof of the following quantitative refinement of the Arzelà-Ascoli Theorem originates from a similar result using the inverse modulus from [Tim94]. A restricted version for a class of functions with small moduli has been proven in the context of computable analysis before (see [Wei03]).

Theorem 4.1.4 (Quantitative Arzelà-Ascoli). *A set $K \subseteq \mathcal{C}([0,1])$ is relatively compact if and only if it is contained in $K^\infty_{l,C}$ for some l, C. Furthermore it holds that*

$$2^{l(n)} + n + C + 1 \leq \left|K^\infty_{l,C}\right|(n) \leq 2^{l(n)+1} + n + C + 2.$$

Proof. The first assertion follows from the classical Arzelà-Ascoli Theorem 4.1.4 and the proof is not repeated here.

To provide the upper bound on the size of $K^\infty_{l,C}$ fix some $n \in \omega$. A collection of balls of radius 2^{-n} that cover $K^\infty_{l,C}$ can be constructed as follows: Consider the index set

$$I := \{-2^{n+C+1}, \ldots, 2^{n+C+1}\} \times \{0, 1, -1\}^{2^{l(n)}}.$$

For $\sigma = (\sigma_0, \sigma_1, \ldots, \sigma_{2^{l(n)}}) \in I$ define a piecewise linear function $f_\sigma : [0,1] \to \mathbb{R}$ by

$$f_\sigma(x) = \begin{cases} \sigma_0 2^{-n-1} & \text{if } x = 0 \\ 2^{-n-1}\left(\sum_{i=0}^{j-1} \sigma_i + \sigma_j(2^{l(n)}x - j)\right) & \text{if } x \in \left(\frac{j-1}{2^{l(n)}}, \frac{j}{2^{l(n)}}\right] \text{ for some } j \in \mathbb{N}. \end{cases}$$

The 2^{-n}-balls centered at the functions f_σ cover the set $K^\infty_{l,C}$ and

$$|I| = (2^{n+C+2}+1)3^{2^{l(n)}} \leq 2^{2^{l(n)+1}+n+C+2}.$$

Since n was arbitrary, the right hand side is a modulus of total boundedness and therefore an upper bound on the size of $K^\infty_{l,C}$.
To establish the lower bound, specify a spanning bound in the sense of Definition 3.1.3: For any two different elements σ and σ' of I it holds that $\|f_\sigma - f_{\sigma'}\|_\infty \geq 2^{-n-1}$. However, not all of the functions f_σ need to lie within $K^\infty_{l,C}$: Whenever $l(n) - l(n-1)$ is bigger than 1, taking two consecutive steps in the same direction leads to a function not included in $K^\infty_{l,C}$. Instead of all of I use the subset I' of σ such that the values of f_σ vary by at most 2^{-n}. An easy counting argument shows that still

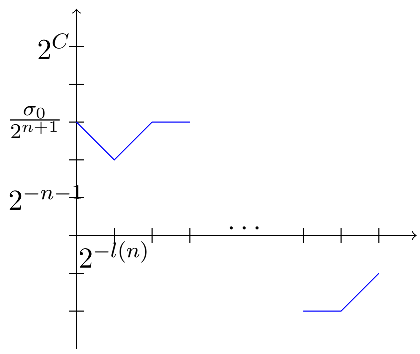


Figure 4.1: The function f_σ for $\sigma = (\sigma_0, -1, 1, 0, \ldots, 0, 1)$.

$$|I'| \geq 2^{2^{l(n)}+n+C+2}.$$

For the functions $\sigma \in I'$ it holds that $f_\sigma \in K^\infty_{l,C}$: The extra condition that a modulus of continuity has to be strictly increasing when non-zero implies that whenever the value of l on n allows the function to vary by 2^{-n} over an interval of length $2^{-l(n)}$ the subsequent values of l will not disallow this behavior. Since any spanning bound of a set has to be smaller than its size by Proposition 3.1.4, the lower bound on $\left|K^\infty_{l,C}\right|$ follows. ∎

The lower bound specified in the theorem is of particular interest: As an application consider the standard representation of the continuous functions on the unit interval from Section 2.1:

Example 4.1.5 (Another minimality property). Recall from Example 4.1.3 that the metric of $\mathcal{C}([0,1])$ is computable in exponential time with respect to the standard representation of continuous functions. Let ξ be an arbitrary open representation of the continuous functions such that the metric is computable in time $T(k,n) := 2^{k(n+2)}$. The running time bound is also a valid running time bound for the range restriction of the representation to one of the sets $K^{\infty}_{l,C}$. This range restriction is again open and therefore has a length k as proven in Proposition 1.2.20. This means that Corollary 3.2.9 is applicable and together with the lower bound from the Theorem 4.1.4 provides some $D \in \omega$ such that

$$2^{l(n)} + n + C + 1 \leq \left|K^{\infty}_{l,C}\right|(n) \leq D(2^{k(n+2)} + n)^2 + D.$$

From this one gets

$$\max\{k(n+2), \mathrm{lb}(n)\} \geq \frac{l(n) - \mathrm{lb}(D) - 1}{2}.$$

The length of $\xi_{\mathcal{C}}$ restricted to this set is l. Thus, as long as $l(n) \geq \mathrm{lb}(n)$, the length of any range restriction of $\xi_{\mathcal{C}}$ to a compact set is very close to optimal within the class of open representations that provide exponential-time computability of the metric. The above argument repeats part of the proof of Theorem 3.2.11.

4.2 L^p-spaces

This chapter prepares the statement of the quantitative version of the Fréchet-Kolmogorov Theorem. This theorem deals with L^p-spaces. We give a very brief introduction, more extensive information can be found in [Wer00]. We furthermore introduce L^p-moduli and Sobolev spaces, which are needed later on. A more in-depth discussion of these topics can be found in [Bre11] or [WKK09]. For later use some notions are introduced in a greater generality than needed in this chapter.

First we recall some basic facts about spaces of integrable functions: Let λ denote the Lebesgue measure on $\mathbb{R}^d$ for any dimension d. In the following $\Omega \subseteq \mathbb{R}^d$ denotes a bounded, measurable set of non-zero measure. Recall that for $p \in [1,\infty)$ the space $\mathrm{L}^p(\Omega)$ is the Banach space of equivalence classes of measurable functions up to equality almost everywhere such that

$$\|f\|_p = \|f\|_{p,\Omega} := \left(\int_\Omega |f|^p \,\mathrm{d}\lambda\right)^{\frac{1}{p}} < \infty,$$

and that for the case $p = \infty$ the norm $\|\cdot\|_\infty$ is defined to be the essential supremum norm. The concrete domain Ω is always either clear from the context or irrelevant and therefore dropped as index of the norm. Since most of this chapter only considers $\Omega = [0,1]$, the space $\mathrm{L}^p([0,1])$ is abbreviated as L^p.

The elements of $\mathrm{L}^p(\Omega)$ are not functions but equivalence classes of functions. Whenever the elements of $\mathrm{L}^p(\Omega)$ are treated like functions, there is a hidden claim involved that the choice of the representative f of an equivalence class $[f]$ is irrelevant. For instance $|[f]|^2$ means the equivalence class $[|f|^2]$. The class $[|g|^2]$ coincides with $|[f]|^2$ for any $g \in [f]$, thus this definition is independent of the representative. In other cases, the equivalence classes lead to slightly more inconvenience. For instance the usual definition of the **support** $\mathrm{supp}([f])$ as closure of the set of x with $f(x) \neq 0$ has to be replaced with the complement of the biggest open set U such that $f|_U \equiv 0$, i.e. such that the restriction is zero almost everywhere. From now on we name the equivalence classes like functions, i.e. f instead of $[f]$.

If Ω is bounded with non-zero Lebesgue measure then $\mathcal{C}(\Omega) \subseteq \mathrm{L}^p(\Omega) \subsetneq \mathrm{L}^q(\Omega)$, whenever $\infty \geq p > q \geq 1$. The corresponding inclusion mappings $\mathrm{L}^p(\Omega) \hookrightarrow \mathrm{L}^q(\Omega)$ are continuous as can be seen using the following well-known result from analysis:

Theorem 4.2.1 (Hölder's Inequality). *For any measurable subset $\Omega \subseteq \mathbb{R}^d$, any measurable functions f, g on Ω and any $p \in [1, \infty]$ the inequality*

$$\|fg\|_1 \leq \|f\|_p \|g\|_q$$

holds, where $q := \frac{1}{1-\frac{1}{p}}$ is the conjugate exponent of p and $q = \infty$ if $p = 1$.

A corollary from this is particularly useful for our purposes:

Corollary 4.2.2 (Version of Hölder's inequality). *For any measurable function f on Ω it holds that*

$$\int_\Omega |f| d\lambda \leq \lambda(\Omega)^{1-\frac{1}{p}} \|f\|_{p,\Omega}.$$

In the case $\Omega = [0, 1]$ the above with $f := |g|^r$ and $p := \frac{s}{r}$ shows that $\|g\|_r \leq \|g\|_s$ whenever $r < s$ and $g \in \mathrm{L}^s(\Omega)$.

4.2.1 L^p-moduli

The quantitative version of the Arzelà-Ascoli Theorem formulated in Theorem 4.1.4 uses a family of relatively compact sets to classify compactness in the space of continuous functions. These sets were indexed with a common modulus of continuity of their elements. To specify such a family for $\mathrm{L}^p(\Omega)$, an appropriate notion of modulus is needed.

For a measurable function f on $\Omega \subset \mathbb{R}^d$ let $\tilde{f}$ be its extension to all of $\mathbb{R}^d$ by 0, then $\|f\|_{p,\Omega} = \|\tilde{f}\|_{p,\mathbb{R}^d}$. For $h \in \mathbb{R}^d$ denote by τ_h the **shift by** h is, i.e.

$$\tau_h : \mathrm{L}^p(\Omega) \to \mathrm{L}^p(\Omega - h), \quad (\tau_h f)(x) := f(x + h).$$

Note that $\tau_h \tilde{f} = \widetilde{\tau_h f}$. Recall that $|\cdot|_\infty$ denotes the supremum norm on $\mathbb{R}^d$.

Definition 4.2.3. Let $1 \leq p < \infty$. A function $\mu : \omega \to \omega$ is called an L^p**-modulus** for $f \in \mathrm{L}^p(\Omega)$ if for all $h \in \mathbb{R}^d$ it holds that

$$|h|_\infty \leq 2^{-\mu(n)} \quad \Rightarrow \quad \|\tau_h \tilde{f} - \tilde{f}\|_p < 2^{-n},$$

and $\mu(n) \neq 0 \Rightarrow \mu(n+1) > \mu(n)$, i.e. μ is strictly increasing whenever it is not zero.

Any L^p-function has an L^p-modulus (cmp. [Bre11, Lemma 4.3]). The L^p-modulus encodes how accurately the function can be approximated by smooth functions within $\mathrm{L}^p(\Omega)$ (c.f. Lemma 5.1.7).

The rest of this chapter investigates some basic properties of L^p-moduli, compares them to the singularity modulus and the modulus of continuity and discusses examples of L^p-moduli for concrete functions. If the reader simply accepts this notion, he may move forward to Section 4.3 and return on demand.

Recall the notion of a singularity modulus from Definition 2.2.1 and Definition 2.2.10. It is possible to obtain such a modulus from an L^1-modulus of a function:

Lemma 4.2.4. *Any* L^1*-modulus of a function is also a singularity modulus.*

Proof. Let $f \in \mathrm{L}^p(\Omega)$ be the function under consideration. Recall from Definition 2.2.10 that f_i denotes the function where all but the i-th variable has been integrated over. Then, if e_i is the i-th unit vector

$$\left|\int_x^{x+h} f_i(t)\mathrm{d}t\right| \leq \left|\int_x^{\infty} f_i(t)\mathrm{d}t - \int_{x+h}^{\infty} f_i(t)\mathrm{d}t\right| \leq \int_{\mathbb{R}} |f_i - \tau_h f_i| \,\mathrm{d}\lambda$$
$$= \|f_i - \tau_h f_i\|_1 \leq \|f - \tau_{he_i} f\|_1$$

Since $|h| = |he_i|_\infty$, the assertion follows from μ being an L^1-modulus of f. ∎

The converse is false as the upcoming Example 4.2.10 shows.

Whenever $1 \leq q \leq p < \infty$, an L^q-modulus can be obtained from an L^p-modulus by shifting by constant. For this we need very weak assumption about the regularity of the domain Ω: Assume that the intersection of the boundary of Ω with the complement of Ω has zero Lebesgue measure. This is for instance true for closed sets, for countable unions of convex sets and for regular sets. The important consequence of this assumption is that it guarantees that any $f \in \mathrm{L}^p(\Omega)$ fulfills

$$\lambda(\operatorname{supp}(\tilde{f})) \leq \lambda(\Omega).$$

The following result can also be proven without the condition, but in that case $\lambda(\Omega)$ has to be replaced by $\lambda(\overline{\Omega})$.

Lemma 4.2.5 (Monotonicity of moduli in p). *Let* Ω *fulfill the regularity condition above,* $1 \leq q \leq p < \infty$ *and* $f \in \mathrm{L}^p(\Omega) \subseteq \mathrm{L}^q(\Omega)$*. If* μ *is an* L^p*-modulus of a function, then an* L^q*-modulus of the function is given by*

$$n \mapsto \mu\left(n + \left\lceil \mathrm{lb}(2\lambda(\Omega))\left(\frac{1}{q} - \frac{1}{p}\right)\right\rceil\right)$$

Proof. First prove an auxiliary statement: Let $g \in \mathrm{L}^q(\mathbb{R}^d)$ be arbitrary. Use the version of Hölder's inequality from Corollary 4.2.2 for $|g|^q$, the characteristic function of $\operatorname{supp}(g)$ and the value $r := \frac{p}{q} \geq 1$ for what is there called p to get

$$\|g\|_q^q = \int_{\operatorname{supp}(g)} |g|^q \,\mathrm{d}\lambda \leq \lambda(\operatorname{supp}(g))^{1-\frac{q}{p}} \left(\int_{\operatorname{supp}(g)} |g|^p \,\mathrm{d}\lambda\right)^{\frac{q}{p}},$$

and therefore
$$\|g\|_q \leq \lambda(\mathrm{supp}(g))^{\frac{1}{q}-\frac{1}{p}} \|g\|_p .$$
From the regularity condition on Ω it follows that $\lambda(\mathrm{supp}(\tau_h\tilde{f}-\tilde{f})) \leq 2\lambda(\Omega)$. Thus, for $h \in \mathbb{R}^d$
$$\|\tau_h\tilde{f}-\tilde{f}\|_q \leq 2^{\mathrm{lb}(2\lambda(\Omega))\left(\frac{1}{q}-\frac{1}{p}\right)} \|\tau_h\tilde{f}-\tilde{f}\|_p.$$
From this inequality the assertion now follows by using that μ is an L^p-modulus. ∎

For later reference, we list the combination of these two lemmas as a corollary:

Corollary 4.2.6 (From $\mathbf{L}^p$-modulus to singularity modulus). *Whenever μ is an L^p-modulus of a function from $\mathrm{L}^p(\Omega)$, then $n \mapsto \mu\left(n + \lceil \mathrm{lb}(2\lambda(\overline{\Omega}))(1-1/p)\rceil\right)$ is a singularity modulus of the function.*

These results avoid the case $p = \infty$ since the definition of an L^p-modulus does not make sense for $p = \infty$. However, the modulus of continuity restricted to a subset of $\mathcal{C}(\overline{\Omega}) \subseteq \mathrm{L}^\infty(\Omega)$ behaves like one would expect an L^∞-modulus to behave: Assume that $f \in \mathcal{C}(\Omega)$ is such that the extension $\tilde{f}$ to all of $\mathbb{R}^d$ by zero is continuous. In this case a modulus of continuity μ of f is also a modulus of continuity of $\tilde{f}$ and can be converted to an L^p-modulus of f: Whenever $|h| \leq 2^{-\mu(n)}$ then $|x-(x+h)| \leq 2^{-\mu(n)}$, and therefore
$$\begin{aligned}\|\tilde{f}-\tau_h\tilde{f}\|_p &= \left(\int_{\overline{\Omega\cup(\Omega-h)}} |\tilde{f}(x)-\tilde{f}(x+h)|^p \mathrm{d}x\right)^{\frac{1}{p}} \\ &\leq 2^{-n}\lambda(\overline{\Omega\cup(\Omega-h)})^{\frac{1}{p}} \\ &\leq 2^{-n+\frac{1+\mathrm{lb}(\lambda(\overline{\Omega}))}{p}}.\end{aligned}$$

It follows, that $n \mapsto \mu\big(n + \lceil \frac{\mathrm{lb}(2\lambda(\overline{\Omega}))}{p}\rceil\big)$ is an L^p-modulus of f. This is exactly what would be expected from Lemma 4.2.5.

If f can not be continuously extended, for instance if it is non-zero in a boundary point, additional information about the function and the domain is needed to obtain an L^p-modulus from a modulus of continuity (this case is handled later in Lemma 5.2.4).

The modulus of continuity does not contain any information about the norm $\|f\|_\infty$ of a function $f \in \mathcal{C}([0,1])$ as it does not change under shift by a constant function. In contrast to that a norm-bound can be deduced from an L^p-modulus. Recall that the diameter of a set is defined by
$$\mathrm{diam}(\Omega) := \sup\{|x-y|_\infty \mid x,y \in \Omega\},$$
where $|\cdot|_\infty$ denotes the supremum norm on $\mathbb{R}^d$.

Lemma 4.2.7. *Whenever μ is an L^p-modulus of some function $f \in \mathrm{L}^p(\Omega)$, then*
$$\|f\|_p < \lceil \mathrm{diam}(\Omega)\rceil 2^{\mu(0)}.$$

Proof. Fix the standard basis vector $e = (1, 0, \dots, 0) \in \mathbb{R}^d$. $\Omega + \lceil \mathrm{diam}(\Omega) 2^{\frac{1}{p}} \rceil e$ is disjoint from Ω. Thus:

$$2^{\frac{1}{p}} \|f\|_p = \left\| f - \tau_{\lceil \mathrm{diam}(\Omega) 2^{\frac{1}{p}} \rceil} f \right\|_p = \left\| \sum_{i=1}^{\lceil \mathrm{diam}(\Omega) 2^{\frac{1}{p}} \rceil 2^{\mu(0)}} \tilde{f} - \tau_{2^{-\mu(0)} e} \tilde{f} \right\|_p$$
$$\leq \sum_{i=1}^{\lceil \mathrm{diam}(\Omega) 2^{\frac{1}{p}} \rceil 2^{\mu(0)}} \|\tilde{f} - \tau_{2^{-\mu(0)} e} \tilde{f}\|_p < \lceil \mathrm{diam}(\Omega) 2^{\frac{1}{p}} \rceil 2^{\mu(0)} 2^0 \leq \lceil \mathrm{diam}(\Omega) \rceil 2^{\mu(0) + \frac{1}{p}}.$$

This proves the assertion. ■

The above can be straightforwardly improved to

$$\|f\|_p \leq \min_{n \in \mathbb{N}} \{ \lceil \mathrm{diam}(\Omega) \rceil 2^{\mu(n) - n} \}.$$

The assumption that an L^p-modulus is strictly increasing when not zero, however, implies that this only leads to a better value if the modulus stays constantly zero on a initial segment.

We need to know how the L^p-modulus changes under linear transformations.

Lemma 4.2.8. *Let μ be an L^p-modulus of a function $f \in \mathrm{L}^p([a, b])$. Then for any $c, d \in \mathbb{R}$*

$$n \mapsto \mu(n + \lceil \mathrm{lb}(|c|) p \rceil) + \lceil \mathrm{lb}(|c|) \rceil$$

is an L^p-modulus of the function $x \mapsto f(cx + d) \in \mathrm{L}^p([ca + d, cb + d])$.

Proof. Follows directly from an application of integration by substitution. ■

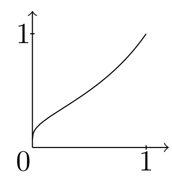
4.2.2 Examples

This section contains some examples of functions that highlight what can be expected and what cannot be expected from the L^p-modulus.

The first example illustrates that a fast growing modulus of continuity need not necessarily result in a fast growing L^p-modulus. This implies that the estimation obtained in the previous section may not be optimal.

Example 4.2.9 (Linear L^p-modulus but exp. mod. of cont.). Recall the function from Example 2.1.3:

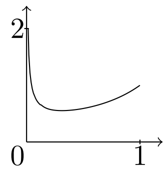
$$f : [0, 1] \to [0, 1], \quad f(x) = \begin{cases} 0 & \text{if } x = 0 \\ \frac{1}{1 - \mathrm{lb}(x)} & \text{if } x \neq 0. \end{cases}$$



This function is continuous but does not have a polynomial modulus of continuity. The following shows that it has a linear L^p-modulus for any $p < \infty$.

To find an L^p-modulus, handle the function close to 0 and far away from 0 separately. The derivative of this function is convex and given by

$$\frac{df}{dx}(x) = \frac{\ln(2)}{(\ln(2) - \ln(x))^2 x}.$$



Therefore, f' assumes its maximum on any closed subinterval of $(0, 1]$ at one of the end-points. For $0 \leq h \leq 1 - \delta$ and $x \in [\delta, 1 - h]$ it holds that

$$|f(x + h) - f(x)| \leq \sup_{y \in [\delta, 1]} \{f'(y)\} h = \max\{f'(\delta), f'(1)\} h.$$

Whenever $\delta \leq 2^{-5}$ the maximum is attained at δ and $f'(\delta) < \frac{1}{\delta}$.

For given δ and h separate $[0, 1]$ into $[0, \delta)$, $[\delta, 1 - h]$ and $(h, 1]$:

$$\begin{aligned} \|f - \tau_h f\|_p &\leq 2 \left(\int_0^\delta |f|^p \,\mathrm{d}\lambda \right)^{\frac{1}{p}} + \left(\int_\delta^{1-h} |f - \tau_h f|^p \,\mathrm{d}\lambda \right)^{\frac{1}{p}} + \left(\int_0^h |f|^p \,\mathrm{d}\lambda + \int_{1-h}^1 |f|^p \,\mathrm{d}\lambda \right)^{\frac{1}{p}} \\ &< 2\delta^{\frac{1}{p}} + \left(\frac{h}{\delta} \right)^{\frac{1}{p}} + (2h)^{\frac{1}{p}}. \end{aligned}$$

The right hand side of the above can for instance be made smaller than 2^{-n} by choosing some integer $D \geq p$ and setting

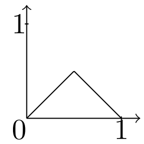
$$\delta := \min\left\{2^{-5}, 2^{-p(n+3)}\right\} \quad \text{and requiring} \quad h \leq \min\{2^{-n-6}, 2^{-D(n+1)-5}, 2^{-D(2n+4)}\}.$$

Thus the function $n \mapsto \max\{n + 6, D(n + 1) + 5, D(2n + 4)\}$ is an L^p-modulus of f. This function is eventually linear, and a linear L^p-modulus can be found by adding a constant.

However, continuity does not guarantee a linear L^p-modulus, in fact the next example shows that continuity does not allow to draw any conclusions about the L^p-modulus: The example constructs continuous functions with supremum norm one on the unit interval that have arbitrarily large L^1-moduli. Therefore, by the monotonicity of the moduli in p from Lemma 4.2.5 they do not have any substantially smaller L^p-moduli for any p.

The idea is that the function from Example 4.2.9 allows a linear L^1-modulus because its behavior is only bad on a small subset of the domain. Restricted to the subsets $[2^{-n}, 1]$ it has moduli of continuity that only depend on n polynomially. We avoid this problem by constructing functions that still have a bad modulus of continuity when restricted to any open subinterval of their domain.

Example 4.2.10 (Functions with bad L^p-modulus). Define a function $g : \mathbb{R} \to [0, \frac{1}{2}]$ as the 1-periodic extension of the function



$$\tilde{g} : (0,1] \to \left[0, \frac{1}{2}\right], \quad x \mapsto \min\{x, 1-x\} \tag{g}$$

This function is Lipschitz continuous with Lipschitz constant 1. For a given function $\mu : \omega \to \omega$ define a sequence of functions $(g_m)_{m \in \omega}$ by

$$g_m : [0,1] \to [0, 2^{-m-1}], \quad x \mapsto 2^{-m} g(2^{\mu(m)} x).$$

One easily checks that g_m is Lipschitz continuous with constant $L_m := 2^{\mu(m)-m}$. Consider the function $f : [0,1] \to [0,1]$ defined by

$$f(x) := \sum_{m \in \mathbb{N}} g_m(x).$$

As uniform limit of continuous functions, this function is continuous. Whenever μ is monotone such that $\mu(n) - \mu(n-1) \geq n+2$ then $n \mapsto \mu(n+1)$ is a modulus of continuity of f. For this assume $|x - y| \leq 2^{-\mu(n+1)}$, then

$$\begin{aligned}
|f(x) - f(y)| &= |\sum_{m \in \mathbb{N}} g_m(x) - g_m(y)| \\
&\leq \sum_{m=0}^{n} |g_m(x) - g_m(y)| + \sum_{m=n+1}^{\infty} |g_m(x) - g_m(y)| \\
&\leq \sum_{m=0}^{n} 2^{\mu(m)-\mu(n+1)-m} + 2^{-n-1} \\
&\leq 2^{-n-2} \sum_{m=0}^{n} 2^{-m} + 2^{-n-1} \leq 2^{-n}.
\end{aligned}$$

The claim is that this function does not allow an L^1-modulus significantly smaller than μ. A straightforward computation shows that

$$\int_0^1 |g_m(x) - g_m(x + 2^{-\mu(n)-1})| \mathrm{d}x \begin{cases} = 2^{-n} + 2^{-\mu(n)-n} & \text{if } n = m \\ \leq 2^{-n-2-m} & \text{otherwise.} \end{cases}$$

Using the inverse triangle inequality, this leads to

$$\begin{aligned}
\|f - \tau_{2^{-\mu(n)-1}} f\|_1 \geq& \int_0^1 |g_n(x) - g_n(x + 2^{-\mu(n)-1})| \mathrm{d}\lambda \\
&- \sum_{m \neq n} \int_0^1 |g_m(x) - g_m(x + 2^{-\mu(n)-1})| \mathrm{d}\lambda \\
\geq& \; 2^{-n-1}
\end{aligned}$$

Therefore, if $\tilde{\mu}$ is an L^1-modulus of f, it needs to fulfill $\tilde{\mu}(n+1) \geq \mu(n) + 1$.

4.2.3 Sobolev spaces

For the proof of the upper bound of the quantitative version of the Fréchet-Kolmogorov Theorem it is convenient to work with weak derivatives. Since Sobolev spaces are investigated in more detail in Section 5.2, we use this opportunity to recall the definition and basic facts about Sobolev spaces. Only the simplest domain $\Omega = [0,1]$ is considered. To simplify notation the domain omitted from the notations. Recall that $\mathcal{C}^\infty([0,1])$ denotes the infinitely often differentiable functions on $[0,1]$.

Definition 4.2.11. A function $f' \in \mathrm{L}^1$ is called the **weak derivative** of a function $f \in \mathrm{L}^1$ if for any $g \in \mathcal{C}^\infty([0,1])$ with $g(0) = 0 = g(1)$ it holds that

$$\int_{[0,1]} fg' \mathrm{d}\lambda = -\int_{[0,1]} f'g \mathrm{d}\lambda.$$

If it exists, the weak derivative of a function is uniquely determined (as an element of L^1). An example can be found in Figure 4.2. If $f \in \mathrm{L}^1$ has a weak derivative f', then there is a continuous representative of f that fulfills for all $x, y \in [0,1]$

$$f(y) - f(x) = \int_{[x,y]} f' \mathrm{d}\lambda.$$

(A proof of this can for instance be found in [Bre11]). In particular any weakly differentiable function is absolutely continuous and a modulus of continuity of a weakly differentiable function is exactly the same as a singularity modulus of the weak derivative (cf. the introduction of Section 2.2). Since Corollary 4.2.6 extracts a singularity modulus from an L^p-modulus, it follows that:

Lemma 4.2.12. *Let $1 \leq p < \infty$. Whenever μ is an L^p-modulus of $f' \in \mathrm{L}^p([0,1])$, then $n \mapsto \mu(n+1)$ is a modulus of continuity of f.*

Sobolev spaces are spaces of weakly differentiable functions and are of great importance in the theory of partial differential equations.

Definition 4.2.13. For $1 \leq p \leq \infty$ the **Sobolev space** $W^{1,p}$ is defined as the set of L^p-functions that have a weak derivative which is also an L^p-function.

Here, the first superscript 1 is for the existence of the first weak derivative and Sobolev spaces using weak derivatives of higher order are considered in Section 5.2. An example can be found in Figure 4.2. It is well known that $W^{1,p}$ can be characterized as space of functions fulfilling an 'L^p Lipschitz condition' (compare for instance [Bre11, Proposition 8.5]). Like for continuous functions this translates to an L^p-modulus of the form $n \mapsto n + C$. Since the given reference uses different terminology and the result is stated for the whole space and not the unit interval, we restate it and give a proof.

Lemma 4.2.14. *The following are equivalent for $f \in \mathrm{L}^p$ with $1 < p < \infty$:*

- *f is included in $W^{1,p}$ and its continuous representative vanishes in 0 and 1.*

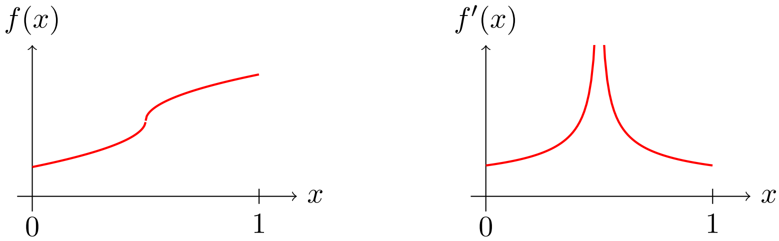


Figure 4.2: The function $f(x) := 1 + \operatorname{sign}(x - 1/2)\sqrt{|x - 1/2|}$ is not differentiable in $\frac{1}{2}$. Its weak derivative is $f'(x) = \frac{1}{2\sqrt{|x-1/2|}}$, thus $f \in W^{1,1}$ but $f \notin W^{1,2}$.

- *There is a $C \in \omega$ such that $n \mapsto n + C$ is an* L^p*-modulus of f.*

Furthermore, the constant C can be chosen as $\lceil \mathrm{lb}(\|f'\|_p) \rceil$.

Proof. First assume that $f \in W^{1,p}$ and that the continuous representative vanishes at 0 and 1. In this case the extension $\tilde{f}$ to the whole real line by zero is continuous and its weak derivative is the extension of the weak derivative of f by zero. Using the version of Hölder's inequality from Corollary 4.2.2 conclude

$$\begin{aligned}
\left\|\tilde{f} - \tau_h \tilde{f}\right\|_p &= \left(\int_{\mathbb{R}} \left|\int_x^{x+h} \tilde{f}' \mathrm{d}\lambda\right|^p \mathrm{d}x\right)^{\frac{1}{p}} \\
&\leq h \left(\int_{\mathbb{R}} \left(\int_0^1 \left|\tilde{f}'(x + sh)\right| \mathrm{d}s\right)^p \mathrm{d}x\right)^{\frac{1}{p}} \\
&\overset{4.2.2}{\leq} h \left(\int_{\mathbb{R}} \int_0^1 \left|\tilde{f}'(x + sh)\right|^p \mathrm{d}s\mathrm{d}x\right)^{\frac{1}{p}} \\
&= h \left\|f'\right\|_p .
\end{aligned}$$

From this it is easy to see that $n + \lceil \mathrm{lb}(\|f'\|_p) \rceil$ is an L^p-modulus of f.

For the other direction assume that $\mu(n) := n + C$ is an L^p-modulus of f. Recall that [Bre11, Proposition 8.5] states that a function $g \in \mathrm{L}^p(\mathbb{R})$ is an element of $W^{1,p}(\mathbb{R})$ if there is a D such that the inequality $\|g - \tau_h g\|_p \leq D\,|h|$ holds for all $h \in \mathbb{R}$. $\tilde{f}$ fulfills this for $D := 2^{2C+1}$: Given h first check if there is some n such that $2^{-\mu(n+1)} \leq |h| < 2^{-\mu(n)}$. If so, then

$$\left\|\tilde{f} - \tau_h \tilde{f}\right\|_p < 2^{-n} = 2^{-n+\mu(n+1)-\mu(n+1)} \leq 2^{C+1}\,|h| .$$

If there is no such n, then $2^{-\mu(0)} \leq |h|$ and using the norm-bound from the L^p-modulus by Lemma 4.2.7 conclude

$$\left\|\tilde{f} - \tau_h \tilde{f}\right\|_p \leq 2\left\|\tilde{f}\right\|_p < 2^{\mu(0)+1} \leq 2^{2C+1}\,|h| .$$

Thus, in any case

$$\left\|\tilde{f} - \tau_h \tilde{f}\right\|_p < 2^{2C+1}\,|h| . \tag{h}$$

It follows that the restriction of $\tilde{f}$ to $[0,1]$ is an element of the Sobolev space.

To show that the continuous representative of f vanishes on the boundary by contradiction, assume $f(0) \neq 0$, and without loss of generality $f(0) > 0$. Then there exists some $\varepsilon, \delta > 0$ such that $f(x) \geq \varepsilon$ for any $x \in [0, \delta]$. Set

$$h := \min\left\{\delta, \left(\varepsilon \cdot 2^{-2C-1}\right)^{\frac{1}{1-\frac{1}{p}}}\right\},$$

then

$$\left\|\tilde{f} - \tau_h \tilde{f}\right\|_p \geq \left(\int_0^h |f|^p \,\mathrm{d}\lambda\right)^{\frac{1}{p}} \geq h^{\frac{1}{p}}\varepsilon \geq 2^{2C+1}h = 2^{2C+1}\left|h\right|,$$

which contradicts (h). Therefore, f vanishes in zero. The argument for the other end of the interval is identical. ∎

In the case $p = 1$, one of the directions of the result fails: Characteristic functions of intervals have $n+1$ as L^1-modulus while not being weakly differentiable. The other direction still holds true.

In Example 2.1.2 the corresponding class of functions for the modulus of continuity was specified as the Lipschitz continuous functions. In Proposition 2.2.5 the corresponding class for the singularity modulus was proven to be L^∞.

4.3 The Fréchet-Kolmogorov Theorem

We are now prepared to state and prove the quantitative version of the Fréchet-Kolmogorov Theorem. As promised it replaces the modulus of continuity by the L^p-modulus:

Definition 4.3.1. For $1 \leq p < \infty$ define the family $(K_l^p)_{l \in \omega^\omega}$ of **Fréchet-Kolmogorov sets** $K_l^p \subseteq \mathrm{L}^p([0,1])$ by

$$K_l^p := \{f \in \mathrm{L}^p([0,1]) \mid f \text{ has } l \text{ as } \mathrm{L}^p\text{-modulus}\}.$$

There is no need to include an upper bound to the norm, since this bound can be extracted from an L^p-modulus by Lemma 4.2.7.

Theorem 4.3.2 (Quantitative Fréchet-Kolmogorov). *Let $1 \leq p < \infty$. A set $K \subseteq \mathrm{L}^p([0,1])$ is relatively compact if and only if it is contained in K_l^p for some l. Furthermore, if $n \geq 3$ and $l(n-3) \geq 9$, then*

$$2^{l(n-3)-4} - 1 \leq \left|K_l^p\right|(n) \leq 2^{n+3l(n+2)+9} + n + l(n+2) + l(0) + 1.$$

Again the first part of the statement follows from the classical Fréchet-Kolmogorov Theorem and is ignored. The methods to obtain these upper and lower bounds differ vastly and thus are split into separate theorems to be proven in separate sections.

4.3.1 The upper bound

This section follows the classical proof of the Fréchet-Kolmogorov Theorem to obtain a proof the following:

Theorem 4.3.3 (Upper Fréchet-Kolmogorov).

$$\left|K_l^p\right|(n) \leq 2^{n+3l(n+2)+9} + n + l(n+2) + l(0) + 1.$$

The proof of the classical Fréchet-Kolmogorov Theorem proceeds by first finding a smooth approximation of an L^p-function with fixed L^p-modulus and then applying the Arzelà-Ascoli Theorem. We follow the same guideline effectivizing the proofs. Thus, start by recalling some facts about the convolution.

Recall, that the convolution $h \star f : \mathbb{R} \to \mathbb{R}$ of a compactly supported integrable function $h : \mathbb{R} \to \mathbb{R}$ with an integrable function $f : \mathbb{R} \to \mathbb{R}$ is defined by

$$(h \star f)(x) := \int_{-\infty}^{\infty} h(x-y) f(y) \mathrm{d}y = \int_{-\infty}^{\infty} h(y) f(x-y) \mathrm{d}y.$$

Recall the following well-known result about convolutions (for instance from [Wer00]):

Proposition 4.3.4 (Convolution and differentiation). *Whenever f is integrable and g is weakly differentiable, then $g \star f$ is weakly differentiable and*

$$(g \star f)' = g' \star f.$$

This means that whenever an integrable functions is convoluted with a smooth function the outcome is smooth again. Moreover, recall Young's inequality for convolution, i.e. the following formula for the L^p-norms of convoluted functions:

Proposition 4.3.5 (Norms of convolutions). *Whenever $1 \leq p \leq \infty$, $g \in \mathrm{L}^p$ and $f \in \mathrm{L}^1$, then $g \star f \in \mathrm{L}^p$ and*

$$\|g \star f\|_p \leq \|g\|_p \|f\|_1 .$$

The appropriate amount of smoothness of the functions to convolute with is to be twice weakly differentiable. We smoothen an integrable function by convoluting with the following functions:

Definition 4.3.6. Let the **mollifier** $(g_m)_{m \in \mathbb{N}}$ be the sequence of functions $g_m : \mathbb{R} \to \mathbb{R}$ defined by

$$g_0(x) = \begin{cases} 2(x+1)^2 & \text{if } -1 \leq x \leq -\frac{1}{2} \\ 1 - 2x^2 & \text{if } -\frac{1}{2} < x < \frac{1}{2} \\ 2(x-1)^2 & \text{if } \frac{1}{2} \leq x \leq 1 \\ 0 & \text{otherwise.} \end{cases}$$

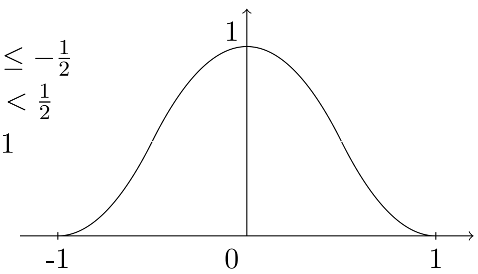
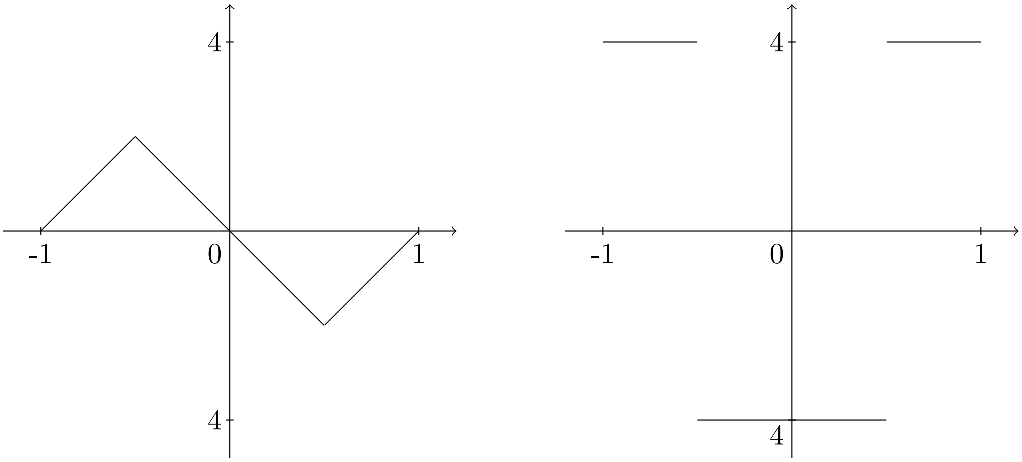


Figure 4.3: the derivatives of g_0:



(a) g_0' (b) g_0''

and

$$g_m(x) := 2^m g_0(2^m x).$$

In functional analysis a mollifier is a sequence of functions that resembles the identity with respect to convolution in the sense that for all integrable f

$$\lim_{m\to\infty} g_m \star f = f.$$

For readers not familiar with this concept the phrase 'mollifier' may be simply considered a name. The above property follows from the results below.

One easily verifies that the support of g_m is $[-2^{-m}, 2^{-m}]$ and that for any m

$$\int_{\mathbb{R}} g_m \mathrm{d}\lambda = \int_{[-2^{-m},2^{-m}]} g_m \mathrm{d}\lambda = 1.$$

Each g_m is twice weakly differentiable, the weak derivatives of g_0 are depicted in Figure 4.3. It holds that

$$\|g_m\|_\infty \le 2^m, \quad \|g_m'\|_\infty \le 2^{2m+1} \quad \text{and} \quad \|g_m''\|_\infty = 2^{3m+2}.$$

Definition 4.3.7. For a fixed function $f \in \mathrm{L}^p$ let its **sequence of differentiable approximations** $(f_m)_{m\in\mathbb{N}}$ consist of the functions $f_m : \mathbb{R} \to \mathbb{R}$ defined by

$$f_m := g_m \star \tilde{f}.$$

That is

$$f_m(x) = \int_{\mathbb{R}} g_m(y)\tilde{f}(x-y)dy.$$

The next two lemmas prove that the 'differentiable approximations' f_m are indeed weakly differentiable and approximate f.

Lemma 4.3.8 (Gradient estimate). *For any $f \in \mathrm{L}^p$ the functions f_m are twice weakly differentiable with*

$$\|f_m\|_\infty \leq 2^m \|f\|_1 \quad \text{and} \quad \|f_m''\|_\infty \leq 2^{3m+2}\|f\|_1.$$

Proof. The first assertion directly follows from Proposition 4.3.5 together with the property $\|g_m\|_\infty \leq 2^m$ following Definition 4.3.6. For the second assertion the differentiation formula for the convolution from Proposition 4.3.4 can be used iteratively:

$$\|f_m''\|_\infty = \|(g_m \star \tilde{f})''\|_\infty = \|g_m'' \star \tilde{f}\|_\infty \leq \|g_m''\|_\infty \|\tilde{f}\|_1 = 2^{3m+1}\|f\|_1.$$

This proves the assertion. ∎

From an L^p-modulus of f it can be estimated how good an approximation f_m is to f:

Lemma 4.3.9 (Approximation). *Let μ be an L^p-modulus of f, then*

$$\|\tilde{f} - f_{\mu(n)}\|_p < 2^{-n+1}.$$

Proof. Use $\int_{\mathbb{R}} g_m \mathrm{d}\lambda = 1$ to see

$$\begin{aligned}
\|\tilde{f} - f_m\|_p^p &= \int_{-\infty}^{\infty} \left|\tilde{f}(x) - \int_{-\infty}^{\infty} g_m(y)\tilde{f}(x-y)\mathrm{d}y\right|^p \mathrm{d}x \\
&= \int_{-\infty}^{\infty} \left|\int_{-\infty}^{\infty} (\tilde{f}(x) - \tilde{f}(x-y))g_m(y)\mathrm{d}y\right|^p \mathrm{d}x \\
&\leq \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} |\tilde{f}(x) - \tilde{f}(x-y)|g_m(y)\mathrm{d}y\right)^p \mathrm{d}x.
\end{aligned}$$

Next note that for any $y \in \mathbb{R}$ from $g_0 \leq 1$ and $p \geq 1$, setting $r := 1 - \frac{1}{p}$ it follows that

$$\begin{aligned}
\left(g_m(y)\lambda([-2^{-m}, 2^{-m}])^r\right)^p &= 2^{m+p-1}g_0(2^m y)^p \\
&\leq 2^{m+p-1}g_0(2^m y) \\
&= 2^{p-1}g_m(y).
\end{aligned} \tag{4.1}$$

Using that the support of g_m is $[-2^{-m}, 2^{-m}]$ and the version of Hölder's inequality from Corollary 4.2.2 conclude

$$\begin{aligned}
\left(\int_{\mathbb{R}} |\tilde{f} - \tau_y\tilde{f}|g_m \mathrm{d}\lambda\right)^p &\leq \left(\lambda([-2^{-m}, 2^{-m}])^r \left\|(\tilde{f} - \tau_y\tilde{f})g_m\right\|_p\right)^p \\
&\overset{(4.1)}{\leq} 2^{p-1}\int_{[-2^{-m},2^{-m}]} |\tilde{f} - \tau_y\tilde{f}|^p g_m \mathrm{d}\lambda.
\end{aligned}$$

Therefore applying Fubini's Theorem leads to

$$\begin{aligned}
\|\tilde{f} - f_m\|_p^p &\leq 2^{p-1}\int_{-2^{-m}}^{2^{-m}} \int_{\mathbb{R}} |\tilde{f} - \tau_y\tilde{f}|^p \mathrm{d}\lambda g_m(y)\mathrm{d}y \\
&= 2^{p-1}\int_{-2^{-m}}^{2^{-m}} \|\tilde{f} - \tau_y\tilde{f}\|_p^p g_m(y)\mathrm{d}y.
\end{aligned}$$

Finally, put $m := \mu(n)$ and use that μ is an L^p-modulus to obtain

$$\|\tilde{f} - f_{\mu(n)}\|_p < 2^{-n+1-\frac{1}{p}}.$$

This proves the lemma. ■

Now it is possible to prove the main result of this section:

Proof (of Theorem 4.3.3). Fix some $n \in \omega$. A cover of K_l^p by open balls of radius 2^{-n} can be constructed as follows: First note that by Lemma 4.3.9 any function from K_l^p has a weakly differentiable function $f_{l(n+2)}$ in its 2^{-n-1}-neighborhood. Lemma 4.3.8 provides an estimate on the supremum norm of the second weak derivative of the function. Note that the support of $f'_{l(n+2)}$ is contained in $[-2^{-l(n+2)}, 1 + 2^{-l(n+2)}]$. Thus Lemma 4.2.14 is applicable to the function $g(x) := f'_{l(n+2)}\big((1 - 2^{-l(n+2)+1})x + 2^{-l(n+2)}\big)$. Since g and $f'_{l(n+2)}$ are related by a linear transformation Lemma 4.2.8 can be used to translate the modulus of g to a modulus of $f'_{l(n+2)}$, namely $m \mapsto m + 3l(n+2) + 6$ is an L^p-modulus of $f'_{l(n+2)}$. Lemma 4.2.12 shows that this implies that $l'(m) := m + 3l(n+2) + 7$ is a modulus of continuity of $f_{l(n+2)}$. Therefore

$$f_{l(n+2)} \in K_{l'}^{\infty}.$$

By the quantitative Arzelà-Ascoli Theorem 4.1.4

$$\begin{aligned} |K_{l'}^{\infty}|\,(n+1) &\leq 2^{l'(n+1)+1} + n + \lceil \mathrm{lb}(\left\|f_{l(n+2)}\right\|_\infty)\rceil \\ &= 2^{n+3l(n+2)+9} + n + \lceil \mathrm{lb}(\left\|f_{l(n+2)}\right\|_\infty)\rceil. \end{aligned}$$

Use the estimate of $\|f_{l(n+2)}\|_\infty$ by $\|f\|_1$ from Lemma 4.3.8, $\|f\|_1 \leq \|f\|_p$ and the bound on $\|f\|_p$ from the L^p-modulus from Lemma 4.2.7 to estimate the supremum norm of $f_{l(n+2)}$ and get

$$\begin{aligned} |K_{l'}^{\infty}|\,(n+1) &\leq 2^{n+3l(n+2)+9} + n + l(n+2) + \lceil \mathrm{lb}(\|f\|_1)\rceil \\ &\leq 2^{n+3l(n+2)+9} + n + l(n+2) + l(0) + 1. \end{aligned}$$

Since the balls in supremum norm are included in the balls in L^p-norm, the 2^{-n}-balls in L^p around the same centers cover K_l^p. ■

4.3.2 The lower bound

To find a lower bound, we use the technique Lorentz used in [Lor66] for the proof of his Lemma 8. Namely we use the following lemma from coding theory:

Lemma 4.3.10. *For any natural number $N \geq 500$ and $M < \frac{N}{3}$ there exists a set I of binary strings of length N that differ pairwise in at least M places and such that*

$$\#I = \left\lfloor 2^{\frac{N}{16}-1} \right\rfloor.$$

Proof. Prove by induction that there is a set I of $\#I$ of strings of length $N \geq 500$ that differ in at least M bits whenever

$$\#I \leq \left\lfloor 2^{\frac{N}{16}} \left(\frac{e}{2^{\frac{3}{2}}\pi} + 1 \right)^{-1} \right\rfloor$$

and $M = \frac{N}{3}$. Note that the right hand side of the above is bigger than $2^{\frac{N}{16}-1}$ this thus in particular proves the assertion.

The induction parameter is the size $\#I$ of the set I. For $\#I = 2$ choose the constant zero string and the constant one string.

Now assume that I is a set of

$$\#I < 2^{\frac{N}{16}} \left(\frac{e}{2^{\frac{3}{2}}\pi} + 1 \right)$$

of strings that differ pairwise in at least M elements. Then using Stirling's Formula that the number of strings that differ in less than M digits from one of the elements of I is at most

$$\begin{aligned}
\#I \sum_{i=0}^{M} \binom{N}{i} &\overset{M \leq \frac{N}{2}}{\leq} \#I(M\binom{N}{M} + 1) \overset{\text{Stirling}}{\leq} \#I \left(\frac{Me}{2\pi} \frac{N^{N+\frac{1}{2}}}{M^{M+\frac{1}{2}}(N-M)^{N-M+\frac{1}{2}}} + 1 \right) \\
&\overset{M=\frac{N}{3}}{=} \#I \left(\frac{\sqrt{N}e}{6\pi} 3^{N+1} 2^{-\frac{2}{3}N-\frac{1}{2}} + 1 \right) \leq \#I \left(\frac{e}{2^{\frac{3}{2}}\pi} + 1 \right) 2^{\mathrm{lb}(2-\frac{2}{3})N + \frac{\mathrm{lb}(N)}{2}} \\
&\overset{N \geq 500}{\leq} \#I \left(\frac{e}{2^{\frac{3}{2}}\pi} + 1 \right) 2^{-\frac{N}{16}} 2^N.
\end{aligned}$$

By induction hypothesis the right hand side is strictly smaller than 2^N. Since the left hand side is an integer it is at most $2^N - 1$. Since there are 2^N strings of length N, at least one string does not lie in the union of these sets and therefore differs from each of the elements of I in M digits. We can add this string to the set I to increase its size by one. ■

Remark 4.3.11. From coding theory it is known that these bounds are not optimal. In particular the assumption $N \geq 500$ can be removed. See for instance [Sud01].

Proposition 4.3.12. *Whenever $n \geq 3$, $l(n-3) \geq 9$ and $1 \leq p < \infty$, then*

$$\left| K_l^p \right| (n) \geq 2^{l(n-3)-4} - 1.$$

Proof. Fix some $n \in \omega$. The assumption $l(n-3) \geq 9$ guarantees that Lemma 4.3.10 can be applied with $N := 2^{l(n-3)} \geq 2^9 = 512$ and $M := 2^{l(n-3)-2} = \frac{N}{4}$ to find a subset I of $\Sigma^{2^{l(n-3)}}$ such that

$$\#I = 2^{2^{l(n-3)-4}-1},$$

and whose elements differ in at least

$$a(n) := M = 2^{l(n-3)-2}$$

digits.

For each string $\sigma \in I$ define a function f_σ as follows: First Consider the 'hat' function

$$f : \mathbb{R} \to [0,1], \quad x \mapsto \max\left\{0, 1 - 2\left|x - \frac{1}{2}\right|\right\}.$$

Set

$$w(n) := 2^{-l(n-3)}, \quad \text{and} \quad h(n) := (p+1)^{\frac{1}{p}} 2^{-n+1} \leq 2^{-n+2}.$$

The last inequality follows from $x \mapsto (x+1)^{\frac{1}{x}}$ being decaying on the positive real line and taking value 2 in 1. For each $\sigma \in I$ define a function f_σ by

$$f_\sigma := h(n) \sum_{i=1}^{2^{l(n-3)}} \sigma_i f\left(\frac{x - iw(n)}{w(n)}\right).$$

That is: Divide $[0,1]$ into intervals of width $w(n)$ and consider the set of functions that may or may not have a hat of height $h(n)$ in each of the intervals (see Figure 4.5).

Since at most one hat is put in each interval for each string σ and $x \in [0,1]$ it is true that almost everywhere $f_\sigma(x) < h(n)$ and therefore $\|f_\sigma\|_p < h(n)$. The weak derivative of f is constantly 2 on $[0, 0.5]$ and the negative of that on $[0.5, 1]$. Thus for the weak derivative of f_σ it holds that $\|f'_\sigma\|_\infty \leq 2h(n)/w(n)$. To obtain the spanning bound prove that $\{f_\sigma \mid \sigma \in I\} \subseteq K^p_l$ and that the elements of this family are of pairwise distance more than 2^{-n}.

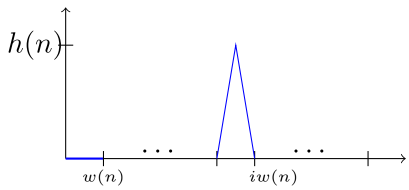


Figure 4.5: The function f_σ for a string σ with $\sigma_1 = 0$ and $\sigma_i = 1$.

To show that these functions are elements of K^p_l, claim that the smallest valid modulus function μ such that $\mu(n-3) = l(n-3)$ is an L^p-modulus of f, i.e. the function

$$\mu(m) = \begin{cases} 0 & \text{if } m < n-3 \\ m + l(n-3) - n + 3 & \text{if } m \geq n-3. \end{cases}$$

Indeed: For an arbitrary shift y and any σ

$$\|f_\sigma - \tau_y f_\sigma\|_p \leq 2\|f_\sigma\|_p < 2h(n) \leq 2^{-n+3}.$$

Thus, for any $m < n-3$ zero is a valid value of an L^p-modulus of f_σ. To see the statement for $m \geq n-3$ use Lemma 4.2.14, which says that it suffices to estimate the L^p-norm of the weak derivative of f_σ:

$$\|f'_\sigma\|_p \leq \frac{2h(n)}{w(n)} = 2^{l(n-3)-n+2} < 2^{l(n-3)-n+3}.$$

Thus, $m \mapsto m + l(n-3) - n + 3$ is an L^p-modulus of f_σ.

Finally estimate the pairwise distance: The set I was chosen such that whenever $\sigma \neq \sigma'$, then σ and σ' differ in at least $a(n) = 2^{l(n-3)-2}$ places. Thus

$$\|f_\sigma - f_{\sigma'}\|_p \geq 2h(n) \left(\frac{a(n)w(n)}{p+1} \right)^{\frac{1}{p}} = 2^{-n+2-\frac{2}{p}} \geq 2^{-n}.$$

This proves the assertion. ∎

5 Representing spaces of integrable functions

Let us summarize the steps we have taken so far. In a search for a good representation for the integrable functions, Section 2.2 constructed the weakest representation of $\mathrm{L}^1(\Omega)$ such that integration is polynomial-time computable and called it the singular representation. It is known that the standard representation of the continuous functions on the unit interval has this very property for the evaluation instead of integration. In contrast to the standard representation of the continuous functions, the singular representation turned out to be discontinuous with respect to the L^1-norm. In particular it leads to a different notion of computability than the metric representation of L^1 and does neither render the norm nor the metric of the space computable.

Section 3.2, more specifically Theorem 3.2.7, proved that the existence of a representation such that the metric is computable in bounded time implies the existence of a family of compact sets of a certain size. This in particular leads to a bound of the length that an open representation computing the metric in bounded time must have when restricted to a compact subset of a certain size (cf. Theorem 3.2.11).

Finally Chapter 4 classified the compact subsets of $\mathrm{L}^p([0,1])$ and found that their size can be parameterized by a common bound of the L^p-modulus of the elements. The size bound turned out to be exponential in the values of the modulus. This is in agreement with the case for the continuous functions on the unit interval, where the relation between the modulus of continuity and the size of the corresponding sets is also exponential (cf. Theorem 4.1.4). Applying the results of Section 3.2 this can be seen as a consequence of the representation rendering the metric exponential time computable (cf. Example 4.1.3). Any representation that renders the metric of L^p exponential time computable cannot have a length significantly less than l when range restricted to K_l^p. The argument for this is completely analogous to the same statement for the continuous functions from Example 4.1.5.

This chapter proceeds as follows: It replaces the singularity modulus in the definition of the singular representation by the L^p-modulus, thereby producing a representation such that the range restrictions are long enough to potentially provide exponential time computability of the metric. In this definition the restriction $\Omega = [0,1]$ can be dropped. It then proves that these representations are computably equivalent to the Cauchy representation of L^p, where as dense sequence the piecewise constant functions with dyadic breakpoints and values are chosen. Indeed exponential time computability of the norm is proven. This indicates that the choice of discrete information encoded is appropriate.

In the second part of the chapter the construction is generalized to Sobolev spaces.

5.1 Representing $\mathrm{L}^p(\Omega)$

Throughout this section let d be a dimension and $\Omega \subseteq \mathbb{R}^d$ a bounded mesurable set. Recall the notations $[\![\cdot]\!]$ of $\mathbb{D}$ and $[\![\cdot]\!]_d$ of $\mathbb{D}^d$ (the index d is omitted if no ambiguity arises) and the rectangles associated to pairs of strings $\mathbf{a}, \mathbf{b} \in \Sigma^*$ encoding dyadic vectors, i.e.

$$[\mathbf{a}, \mathbf{b}] = [[\![\mathbf{a}]\!]_d, [\![\mathbf{b}]\!]_d] := \{x \in \mathbb{R}^d \mid [\![\mathbf{a}]\!]_d \leq x \leq [\![\mathbf{b}]\!]_d\}.$$

We only consider the case $1 \leq p < \infty$ and one should keep in mind that $\mathrm{L}^p(\Omega) \subseteq \mathrm{L}^q(\Omega)$ holds for bounded Ω and whenever $p \geq q$. Also recall that $\tilde{f}$ denotes the extension of f to all of $\mathbb{R}^d$ by zero.

Definition 5.1.1. Define the **representation** ξ_{L^p} **of** $\mathrm{L}^p(\Omega)$ as follows: A length-monotone string function φ is a ξ_{L^p}-name of $f \in \mathrm{L}^p(\Omega)$ if and only if for all strings $\mathbf{a}, \mathbf{b} \in \Sigma^*$ encoding dyadic vectors and $n \in \omega$

$$\left| \int_{[\mathbf{a},\mathbf{b}]} \tilde{f} \mathrm{d}\lambda - [\![\varphi(\langle \mathbf{a}, \mathbf{b}, n \rangle)]\!] \right| < 2^{-n},$$

and $|\varphi|$ is an L^p-modulus of f (compare to Definition 4.2.3).

This indeed defines a representation: Any two distinct functions differ in the values of their integrals over some dyadic rectangle. Thus, ξ_{L^p} is a function. Any function from L^p has an L^p-modulus. Therefore, ξ_{L^p} is surjective. From now on L^p is always equipped with the representation ξ_{L^p} if not explicitly stated otherwise.

Since the singular representation ξ_s from Section 2.2 is the weakest representation such that integration is polynomial-time computable the following implies that it is polynomial-time reducible to the representation ξ_{L^p}.

Theorem 5.1.2. *For each $1 \leq p < \infty$, the restriction of the integration operator eq.* (INTu) *from page 26 to $\mathrm{L}^p(\Omega)$ is polynomial-time computable.*

Proof. For any ξ_{L^p}-name φ of a function, $|\varphi|$ is an L^p-modulus. By Corollary 4.2.6 the mapping $n \mapsto |\varphi| (n + \lceil \mathrm{lb}(2\lambda(\overline{\Omega})) \rceil)$ is a singularity modulus of the same function. Thus the mapping padding the names to have this length is a reduction from ξ_{L^p} to ξ_s. This mapping is polynomial-time computable. The assertion now follows from the polynomial-time computability of the integration operator with respect to the singular representation which was proven as part of the minimality of the singular representation from Theorem 2.2.12. ∎

5.1.1 Reducibility to the Cauchy representation

From now on let the dimension d and the domain Ω be fixed. To simplify notation, they are often omitted. In particular we write L^p for both $\mathrm{L}^p(\Omega)$ and $\mathrm{L}^p(\mathbb{R}^d)$, if it is clear from the context or irrelevant which we mean.

Recall the Cauchy representation of $\mathrm{L}^p(\Omega)$ with respect to a dense sequence from Definition 1.2.2. We have to choose some canonical dense sequence. The most popular choice are dyadic step functions: Call a function a dyadic step function, if it is a dyadic linear combination of characteristic functions of sets of the form $[r, q]$ for some dyadic vectors $r, q \in \mathbb{D}^d$.

An enumeration of this set can be obtained as follows: When given some $n \in \mathbb{N}$ remove the leading digit to get a string. If this string is a finite sequence of pairs of encodings of dyadic boxes and dyadic numbers, then let x_n be the corresponding dyadic step function. If the string is not such a list then let x_n be the zero function.

Together with this sequence, $\mathrm{L}^p(\Omega)$ is an effective metric space. Thus, it inherits a Cauchy representation according to Definition 1.2.2. This representation is well established at least for investigating computability in $\mathrm{L}^p([0,1])$ (cf. [PER89; ZZ99; Zho99] and many more). It was used in reverse mathematics even earlier (see [BS86]).

The goal of this section is to prove the following:

Theorem 5.1.3. ξ_{L^p} *computably equivalent to the Cauchy representation of* L^p*.*

One of the two reducibilities is easy to prove and proven now. The other direction is more complicated and postponed until the end of the chapter.

Proof (ξ_{L^p} is reducible to the Cauchy representation). [1] An oracle Turing machine that translates a name φ of a function f in the Cauchy representation into a ξ_{L^p}-name can be specified as follows: Given φ as oracle and a string $\mathbf{c}$ as input set $n := |\mathbf{c}|$. The machine obtains a valid value $\mu(n)$ of an L^p-modulus of f as follows: Let f_{n+2} be the function encoded by $\varphi(n+2)$. Since this function is encoded as a list of the boxes it does not vanish on and its values on these boxes, the machine can obtain a bound 2^k on the number of boxes, 2^l of their diameters and 2^m of the values. Note that a dyadic step function that is defined as a linear combination of 2^k characteristic functions on sets of size 2^l can, when shifted by y, at most differ from the original function on a set of size $d \cdot |y|_\infty \cdot 2^{(d-1)l+k}$. Thus, since the difference can be majorized by 2^{m+1} on the set where it is nonzero, get

$$\sup_{|y|_\infty \le h} \|f_{n+2} - \tau_y f_{n+2}\|_p \le 2^{m+1} \cdot \left(d \cdot h \cdot 2^{(d-1)l} 2^k\right)^{\frac{1}{p}}.$$

This means that $r := \lceil p \rceil (d-1)l + k + \lceil \mathrm{lb}(d) \rceil + \lceil p \rceil (m+n+2)$ is a valid value of an L^p-modulus of f_{n+2} in $n+1$. Now, whenever $|y|_\infty \le 2^{-r}$ then

$$\begin{aligned} \|f - \tau_y f\|_p &\le \|f - f_{n+2} - \tau_y f + \tau_y f_{n+2}\|_p + \|f_{n+2} - \tau_y f_{n+2}\|_p \\ &\le 2\|f - f_{n+2}\|_p + \|f_{n+2} - \tau_y f_{n+2}\|_p < 2^{-n}. \end{aligned}$$

Thus, r is indeed a candidate for a value of an L^p-modulus of f in n. By repeating the procedure for all values of n smaller than $|\mathbf{c}|$ and increasing r if necessary, the machine computes a value $\mu(|\mathbf{c}|)$ of an L^p-modulus of f.

Next, the machine checks if the input string is of the form $\mathbf{c} = \langle \mathbf{a}, \mathbf{b}, 1^n \rangle$. If it is, it computes approximations of the integrals by returning the integrals of a dyadic step

[1] This proof was considerably simplified due to remarks by an anonymous referee.

function which approximates the function accurately enough in L^p. Before returning it, it pads the encoding of that approximation to length at least $\mu(|\mathbf{c}|)$. If the input string $\mathbf{c}$ is not of the form $\langle \mathbf{a}, \mathbf{b}, 1^n \rangle$, the machine returns $1^{\mu(|\mathbf{c}|)}$ (in this case only the length is relevant). ■

The other reduction is constructed in the proof of Theorem 5.1.8.

5.1.2 Equivalence to the Cauchy representation

The basic idea for the other direction is to approximate the function from $L^p(\Omega)$ by dyadic step functions where the values are the integrals over small boxes. However: The following example shows that there might be regions where the error of such step functions converges slowly, at least locally.

Example 5.1.4. Consider $\Omega = [0,1]$ and the function $f(x) := x^{-\frac{1}{2}} \in L^1(\Omega)$. For $m \in \omega$ define a function $\tilde{f}_m : [0,1] \to \mathbb{R}$ by

$$\tilde{f}_m(x) := \begin{cases} \int_{2^{-m}(i-1)}^{2^{-m}i} f(t)\mathrm{d}t & \text{if } 2^{-m}(i-1) \leq x < 2^{-m}i \\ 0 & \text{if } x = 1. \end{cases}$$

That is: $\tilde{f}_m$ is the step function that arises from f by averaging over small intervals, then for all $m, n \in \omega$ it holds that

$$\begin{aligned} \|f - \tilde{f}_m\|_{1,[0,2^{-n}]} &= \int_0^{2^{-n}} \left| f(x) - \lambda([0,2^{-n}])^{-1} \int_0^{2^{-n}} f(t)dt \right| dx \\ &= \int_0^{2^{-n}} \left| x^{-\frac{1}{2}} - 2^{\frac{n}{2}+1} \right| dx \\ &= \int_0^{2^{-n-2}} (2^{\frac{n}{2}+1} - x^{-\frac{1}{2}})dx + \int_{2^{-n-2}}^{2^{-n}} (x^{-\frac{1}{2}} - 2^{\frac{n}{2}+1})dx \\ &= 2^{-\frac{n}{2}}. \end{aligned}$$

One might suspect that there are functions that have enough singularities that they have a similar error everywhere in the unit interval and therefore the L^1-norm of of the difference does not converge to zero.

It is still possible to prove that the step functions approximate, but to do this the L^p-function has to be globally approximated by a continuous function first. Curiously this leads us back to methods very similar to those used in Section 4.3.1 to prove the upper bound of the quantitative version of the Fréchet-Kolmogorov Theorem. Namely: smoothing L^p-functions by convolution. However, the information provided by the representation dictates the use of a merely piecewise continuous mollifier and we do not restrict to one dimension. For the sake of readers not familiar with convolution we avoid to mention it.

For easier notation write $[x]_m := x + [-2^{-m-1}, 2^{-m-1}]^d$. That is, $[x]_m$ denotes the closed ball of radius 2^{-m-1} around x in supremum norm. The Lebesgue measure of these sets is given by $\lambda([x]_m) = 2^{-dm}$. Recall that $\tilde{f}$ denotes the extension of a function to all of $\mathbb{R}^d$ by zero.

Definition 5.1.5. Let $f \in \mathrm{L}^p$ be a function. Define the **sequence of continuous approximations** $(f_m)_{m\in\mathbb{N}}$ to f by

$$f_m(x) := 2^{dm} \int_{[x]_m} \tilde{f} \mathrm{d}\lambda.$$

The next two lemmas show that quantitative information about how accurately the f_m approximate f in L^p and about their modulus of continuity can be obtained from an L^p-modulus of f.

Lemma 5.1.6 (Continuity). *Whenever μ is an L^p-modulus of $f \in \mathrm{L}^p$, the function $n \mapsto \mu(n + \lceil \frac{dm}{p} \rceil)$ is a modulus of continuity of f_m.*

Proof. Use the version of Hölder's inequality from Corollary 4.2.2 to conclude

$$\begin{aligned} |f_m(x) - f_m(y)| &= 2^{dm} \left| \int_{[x]_m} \tilde{f} \mathrm{d}\lambda - \int_{[y]_m} \tilde{f} \mathrm{d}\lambda \right| \\ &\leq 2^{dm} \int_{[x]_m} \left| \tilde{f}(t) - \tilde{f}(t - (x-y)) \right| \mathrm{d}t \\ &\leq 2^{\frac{d}{p}m} \left\| \tilde{f} - \tau_{x-y}\tilde{f} \right\|_p. \end{aligned}$$

From this the assertion is obvious. ∎

How good an approximation f_m is to f can be read off from an L^p-modulus of f:

Lemma 5.1.7 (Approximation). *Let μ be an L^p-modulus of f. Then*

$$\|\tilde{f} - f_{\mu(n)}\|_p < 2^{-n}.$$

Proof. Using $\int_{[0]_m} 2^{dm} \mathrm{d}\lambda = 1$ one sees that

$$\begin{aligned} \|\tilde{f} - f_m\|_p^p &\leq \int_{\mathbb{R}^d} \left| \tilde{f}(s) - 2^{dm} \int_{[0]_m} \tilde{f}(t+s) \mathrm{d}t \right|^p \mathrm{d}s \\ &= 2^{dmp} \int_{\mathbb{R}^d} \left| \int_{[0]_m} (\tilde{f}(s) - \tilde{f}(t+s)) \mathrm{d}t \right|^p \mathrm{d}s \\ &\leq 2^{dmp} \int_{\mathbb{R}^d} \left(\int_{[0]_m} |\tilde{f}(s) - \tilde{f}(t+s)| \mathrm{d}t \right)^p \mathrm{d}s. \end{aligned}$$

Use the version of Hölder's inequality from Corollary 4.2.2 and Fubini's Theorem to get

$$\|\tilde{f} - f_m\|_p^p \leq 2^{dm} \int_{[0]_m} \int_{\mathbb{R}^d} |\tilde{f}(s) - \tilde{f}(t+s)|^p \mathrm{d}s\mathrm{d}t.$$

Set $m := \mu(n)$ and use that μ is an L^p-modulus to see

$$\|\tilde{f} - f_m\|_p < \left(2^{dm} \int_{[0]_m} 2^{-pn} \mathrm{d}t\right)^{\frac{1}{p}} = 2^{-n},$$

which proves the assertion. ∎

We can now prove a theorem that strengthens one of the implications of Theorem 5.1.3 in that it specifies a time bound of the reduction.

Theorem 5.1.8. *For any $1 \leq p < \infty$ and bounded, measurable $\Omega \subseteq \mathbb{R}^d$ the Cauchy representation of $\mathrm{L}^p(\Omega)$ is computably reducible to ξ_{L^p}. There is a machine that computes the translation in time $2^{O(T)}$ for*

$$T(l,n) = l\left(n + \left\lceil\frac{d}{p}\right\rceil l(n+1) + \left\lceil\frac{\mathrm{lb}(\lambda(\Omega)}{p}\right\rceil + 2\right) \lceil \mathrm{lb}(l(3dl(n+1) + 3n + 9))\rceil.$$

Proof (also of Theorem 5.1.3). Let φ be a ξ_{L^p}-name of $f \in \mathrm{L}^p(\Omega)$. Set $C := \lceil \mathrm{lb}(\lambda(\Omega))/p\rceil$. For any $z \in \mathbb{D}^d$ fix some binary encoding $\mathbf{a}_z \in \Sigma^*$ (for example the unique canceled encoding such that the encoding of the enumerator has no leading zeros). Furthermore, let e be the constant one vector $(1,\ldots,1)$ and set

$$d_{z,k,N} := 2^{dN} \left[\!\left[\varphi(\langle \mathbf{a}_{z-2^{-N-1}e}, \mathbf{a}_{z+2^{-N-1}e}, 1^k\rangle)\right]\!\right] \quad \text{and} \quad \mu := |\varphi|.$$

Thus, $d_{z,k,N}$ is a 2^{-k}-approximation to mean value of f on the small box

$$[z - 2^{-N-1}e, z + 2^{-N-1}e] = [z]_N$$

around z and therefore also an 2^{-k}-approximation to the value of f_N in z.

Consider the step function

$$F_{k,N,M} := \sum_{z \in \mathbb{D}_M} d_{z,k,N} \chi_{[z]_M},$$

where $\mathbb{D}_M$ denotes the set of $z \in \mathbb{D}^d$ such that each component is of the form $\frac{m}{2^M}$ and such that $[z]_M \cap \Omega \neq \emptyset$. Since Ω is bounded there is a constant D such that $\#\mathbb{D}_M \leq 2^{dM+D}$.

Obviously, the step function $F_{k,N,M}$ can be uniformly computed from the name φ and the constants k, N and M. To see how to choose k, N and M write

$$\|f - F_{k,N,M}\|_p \leq \|f - f_N\|_p + \|f_N - F_{k,N,M}\|_p$$

By Lemma 5.1.7 for the first summand to be smaller than 2^{-n-1}, N should be chosen $\mu(n+1)$. For the second summand, note that each $x \in \Omega$ is 2^{-M} close to some $z \in \mathbb{D}_M$ and that for these z

$$|f_N(z) - F_{k,N,M}(z)| < 2^{dN-k}.$$

Choosing $M := \mu(n+\lceil \frac{dN}{p} \rceil + C+2)$ and $k := dN+n+2$, using the modulus of continuity of f_N from Lemma 5.1.6 and that $F_{k,N,M}$ is piecewise constant obtain

$$\|f_N - F_{k,N,M}\|_p \leq \lambda(\Omega)^{\frac{1}{p}} \|f_N - F_{k,N,M}\|_\infty < 2^{-n-1}$$

Summing up, the result is smaller than 2^{-n}.

From the above it is clear that the machine can specify the function $F_{k,N,M}$ in time $2^{O(T)}$: To compile a list describing $F_{k,N,M}$ it has to write an query of length less than $3(k+1)$ and copy the answer of length $|\varphi|(3k+3)$ for each of the 2^{dM+D} values of z where $F_{k,N,M}$ can be non-zero. The rest is a question of filling in the numbers. ■

The Cauchy representation of L^p is continuous. Thus, the above proves that ξ_{L^p} is a continuous mapping. Whenever p is computable, the L^p-norm is computable with respect to the standard representation, and therefore also with respect to ξ_{L^p}.

Corollary 5.1.9. *Whenever p is computable, the norm on L^p is computable. If p is polynomial-time computable, the norm on L^p is computable in exponential time. That is: in time $2^{P(l,n)}$ for some second-order polynomial P.*

Proof. With respect to the Cauchy representation of L^p the norm is polynomial-time computable relative to p. The assertion follows by composing this algorithm with the exponential time reduction from the previous theorem. ■

The above can be improved to exponential time computability relative to p.

In particular we constructed a representation of $L^1([0,1])$ such that integration, addition and scalar multiplication are polynomial-time computable and the norm is exponential time computable. It is not to difficult to see that a minimality result like the ones for the representation of continuous functions from Theorem 2.1.7, the singular representation from Theorem 2.2.12 and the regular representation from Theorem 2.3.8 cannot be proven for this representation.

Remark 5.1.10. Just as in Example 4.1.3 we can use the running time of the norm of L^p to get bounds on the size of sets of functions that have a name of a certain size. For the case $\Omega = [0,1]$ the extracted bound is significantly worse than the one specified in Theorem 4.3.2 as it includes an iteration of the length function in the exponent. This is due to the use of non-smooth approximations. The above justifies that the methods used to obtain the reduction and the upper bound of the Fréchet-Kolmogorov Theorem are so similar.

5.2 Representing Sobolev spaces

This section only considers the case $\Omega = [0,1]$. To avoid confusion with the convention of omitting the domain from the previous section, we always explicitly denote the domain at for L^p but define the Sobolev spaces only for the unit interval without the domain denoted explicitly.

Recall that in Section 4.2.3 the weak derivative and the Sobolev space $W^{1,p}$ were introduced. Denote the m times iterated weak derivative of a function f by $f^{(m)}$. The **Sobolev space** $W^{m,p}$ is the space of all functions $f \in \mathrm{L}^p([0,1])$ such that the weak derivatives $f', \dots, f^{(m)}$ exist and are from $\mathrm{L}^p([0,1])$. Equipped with the norm

$$\|f\|_{m,p} := \sqrt[p]{\|f\|_p^p + \|f^{(m)}\|_p^p}$$

this space is a Banach space, and for $p = 2$ a Hilbert space.

Remark 5.2.1. While the above are the natural choices for norms, many other choices of equivalent norms are possible. Under reasonable assumptions on the domain for instance the norm

$$\|f\|'_{m,p} := \int_\Omega f \mathrm{d}\lambda + \left\|f^{(m)}\right\|_p$$

is equivalent to the norm $\|\cdot\|_{m,p}$. By Proposition 3.1.6, the metric entropy does not change essentially under the change of equivalent norms. We take the above as an indicator that a name should encode an L^p-moduls of the highest derivative and integrals over the original function. This is by no means a rigorous argument; the choice justifies itself by its results.

In dimension one from $f^{(m)} \in \mathrm{L}^p([0,1])$ it follows that $f^{(m-1)}$ has a continuous representative and hence is in $\mathrm{L}^p([0,1])$: Corollary 4.2.6 transforms an L^p-modulus of $f^{(m)}$ to a modulus of continuity of $f^{(m-1)}$.

Definition 5.2.2. Define a second-order representation $\xi_{m,p}$ of $W^{m,p}$ by letting a length-monotone string function φ be a name of a function $f \in W^{m,p}$ if and only if for any strings $\mathbf{a}, \mathbf{b}$ such that $[\![\mathbf{a}]\!], [\![\mathbf{b}]\!] \in [0,1]$ and any $n \in \mathbb{N}$

$$\left| \int_{[\mathbf{a},\mathbf{b}]} f \mathrm{d}\lambda - [\![\varphi(\langle \mathbf{a}, \mathbf{b}, 1^n \rangle)]\!] \right| < 2^{-n},$$

and $|\varphi|$ is an L^p-modulus (see Definition 4.2.3) of the highest derivative $f^{(m)}$ of f.

The arguments for this to define a representation are the same as for ξ_{L^p}. From now on we always equip $W^{m,p}$ with the second-order representation $\xi_{m,p}$. The representations ξ_{L^p} from Definition 5.1.1 coincide with $\xi_{0,p}$, so no ambiguities arise. The space $\mathcal{C}([0,1])$ is always equipped with its standard representation $\xi_{\mathcal{C}}$ from Section 2.1.

5.2.1 The space $W^{1,p}$

Before investigating the space $W^{m,p}$ consider the simplest non-trivial case $m = 1$. As a set $W^{1,p}$ is contained in $\mathrm{L}^p([0,1])$. From the definition of the norm on $W^{1,p}$ it follows, that the embedding $W^{1,p} \hookrightarrow \mathrm{L}^p([0,1])$ is continuous.

Theorem 5.2.3 (Embedding $W^{1,p}$ into L^p). *The embedding of $W^{1,p}$ into $\mathrm{L}^p([0,1])$ is computable in polynomial time.*

For the proof it is necessary to obtain an L^p-modulus of a function from a modulus of continuity and some extra information. The corresponding result is interesting on its own behalf. Therefore, we state it separately and in more generality.

Lemma 5.2.4. *Let μ be a modulus of continuity of some function $f \in \mathcal{C}(\Omega)$ and let ν be an L^p-modulus of the characteristic function of Ω. Then an L^p-modulus of f is given by*

$$\eta(n) := \max\left\{\mu(n + \lceil \mathrm{lb}(\lambda(\overline{\Omega}))/p\rceil + 1), \nu(n + \lceil \mathrm{lb}(\|f\|_\infty)\rceil + 1)\right\}.$$

Proof. Recall that for sets A and B the symmetric difference $A\Delta B$ is defined by

$$A\Delta B := (A \cup B) \setminus (A \cap B).$$

Note that ν being an L^p-modulus of the characteristic function of Ω means exactly that $\lambda\big(\Omega\Delta(\Omega+h)\big)^{\frac{1}{p}} < 2^{-n}$, whenever $|h| \leq 2^{-\nu(n)}$. From this get that, whenever $|h| \leq 2^{-\eta(n)}$, it holds that:

$$\begin{aligned}
\|f - \tau_h f\|_p &\leq \|\chi_{\Omega\setminus(\Omega+h)\cup\Omega\setminus(\Omega-h)} f\|_p + \|\chi_{\Omega\cap(\Omega+h)}(f - \tau_h f)\|_p \\
&\leq \|f\|_\infty \cdot \lambda(\Omega\Delta(\Omega+h))^{\frac{1}{p}} + \left(\int_{\Omega\cap(\Omega+h))} |f - \tau_h f|^p \mathrm{d}\lambda\right)^{\frac{1}{p}} \\
&< 2^{\mathrm{lb}(\|f\|_\infty)} 2^{-n-\lceil(\mathrm{lb}(\|f\|_\infty))\rceil - 1} + 2^{\frac{\mathrm{lb}(\lambda(\overline{\Omega}))}{p}} 2^{-n-\left\lceil\frac{\mathrm{lb}(\lambda(\overline{\Omega}))}{p}\right\rceil - 1} \\
&\leq 2^{-n}.
\end{aligned}$$

This proves the claim. ∎

For $\Omega = [0,1]$ the characteristic function has $n \mapsto n+1$ as modulus and the previous result states that up to a bound on the norm, a modulus of continuity contains strictly more information about the function than an L^p-modulus.

Proof (Proof of Theorem 5.2.3). The following specifies an oracle Turing machine that transforms a $\xi_{1,p}$-name φ of f into a $\xi_{0,p}$-name of f: The approximations to the integrals for the $\xi_{0,p}$-name can be read from φ. To find the right length of the output, access to an L^p-modulus of the function is needed. Since $|\varphi|$ is an L^p-modulus of f', by Lemma 4.2.12 $\mu(n) := |\varphi|(n+1)$ is a modulus of continuity of f. Recall from Lemma 5.2.4 that to obtain an L^p-modulus of f from a modulus of continuity of f

it suffices to have a bound on the supremum norm. By the mean value theorem for integration

$$\int_0^1 f \mathrm{d}\lambda = f(y)$$

for some $y \in [0,1]$. Let $\mathbf{a}$ and $\mathbf{b}$ be encodings of 0 and 1 as dyadic numbers. Then

$$\begin{aligned} |f(y)| &\leq \left| f(y) - \int_0^1 f \mathrm{d}\lambda \right| + \left| \int_0^1 f \mathrm{d}\lambda - [\![\varphi(\langle \mathbf{a}, \mathbf{b}, \varepsilon \rangle)]\!] \right| + |[\![\varphi(\langle \mathbf{a}, \mathbf{b}, \varepsilon \rangle)]\!]| \\ &\leq |[\![\varphi(\langle \mathbf{a}, \mathbf{b}, \varepsilon \rangle)]\!]| + 1. \end{aligned}$$

Choose some integer Q such that 2^Q is a bound for $|[\![\varphi(\langle \mathbf{a}, \mathbf{b}, \varepsilon \rangle)]\!]| + 1$. Bound the supremum norm of f by using the modulus of continuity and the triangle inequality: Fix some $x \in [0,1]$ and set

$$x_i := x + \frac{i+1}{2^{\mu(0)}}(y - x),$$

then $x_0 = x$, $x_{2^{\mu(0)}} = y$ and $x_i - x_{i+1} \leq 2^{-\mu(0)}$. Thus,

$$|f(x)| \leq \sum_{i=0}^{2^{\mu(0)}-1} |f(x_i) - f(x_{i+1})| + |f(y)| \leq 2^{\max\{\mu(0), Q\}+1}.$$

Taking the supremum on both sides gives $\|f\|_\infty \leq 2^{\max\{\mu(0),Q\}+1}$.

Lemma 5.2.4 says that the function

$$n \mapsto \max\{\mu(n+1), n + \max\{\mu(0), Q\} + 3\}$$

is an L^p modulus of f. This function can be computed in polynomial-time from μ. Thus, the only thing the machine has to do is to pad the encodings of the return values of φ to be longer than the above. ■

In dimension one, the Sobolev spaces consist of continuous functions and the embedding $W^{1,p} \hookrightarrow \mathcal{C}([0,1])$ is well known to be continuous (for $1 < p \leq \infty$ it is compact).

Theorem 5.2.5 (Embedding in continuous functions). *For any $1 \leq p < \infty$ the embedding of $W^{1,p}$ into $\mathcal{C}([0,1])$ is polynomial-time computable.*

Proof. Let φ be a $\xi_{1,p}$-name of a function $f \in W^{1,p}$. Describe an oracle Turing machine that transforms this name into a $\xi_{\mathcal{C}}$-name of f: Assume the machine is given some input $\mathbf{c}$ and provided φ as oracle. Note that by Lemma 4.2.12 the mapping $\mu(n) := |\varphi|(n+1)$ is a modulus of continuity of the continuous representative of f. Therefore the necessary length of the return value is known. If the input is not of the form $\mathbf{c} = \langle \mathbf{a}, \mathbf{1}^n \rangle$, where $\mathbf{a}$ is the encoding of some dyadic number $d \in [0,1]$ return a sufficiently long sequence of zeros. If it is of that form an approximation to $f(d)$ can be obtained as follows: By the mean value theorem

$$2^{\mu(n+1)+1} \int_{d-2^{-\mu(n+1)}}^{d+2^{-\mu(n+1)}} f \mathrm{d}\lambda = f(y)$$

for some $y \in [d - 2^{-\mu(n+1)}, d + 2^{-\mu(n+1)}]$ and therefore

$$\left| f(d) - 2^{\mu(n+1)+1} \int_{d-2^{-\mu(n+1)}}^{d+2^{-\mu(n+1)}} f \mathrm{d}\lambda \right| < 2^{-n-1}.$$

Let $\mathbf{b}^{\pm}$ denote encodings of $d \pm 2^{-\mu(n+1)}$. These encodings $\mathbf{b}^{\pm}$ are easily obtained from $\mathbf{a}$. Set

$$q := 2^{\mu(n+1)+1} \left[\!\left[\varphi(\langle \mathbf{b}^{-}, \mathbf{b}^{+}, \mathtt{1}^{\mu(n+1)+n+2} \rangle) \right]\!\right].$$

This is an approximation to $f(d)$ and (a sufficiently long encoding is) a valid return value. ∎

Note that this result does not imply the polynomial-time computability of the inclusion into $\mathrm{L}^p([0,1])$ from Theorem 5.2.3: Polynomial time computability of the restriction of the integration operator from (INTu) on page 26 is known to fail on $\mathcal{C}([0,1])$ (for instance [Kaw+15, Example 6h]). On L^p on the other hand this operator is polynomial-time computable by Theorem 5.1.2. Thus, the embedding $\mathcal{C}([0,1]) \hookrightarrow \mathrm{L}^p$ is not polynomial-time computable.

Corollary 5.2.6 (Differentiation). *The operator*

$$\frac{d}{dx} : W^{1,p} \to \mathrm{L}^p, \quad f \mapsto f'$$

is polynomial-time computable.

Proof. A given $\xi_{1,p}$-name φ of a function $f \in W^{1,p}$ can be transformed into a ξ_{L^p} name of f' in polynomial-time as follows: An L^p-modulus for f' is contained in the $\xi_{1,p}$-name. It remains to compute the integrals. By Theorem 5.2.5 it is possible to obtain approximations to the values of f on dyadic numbers. Using the formula

$$f(y) - f(x) = \int_x^y f' \mathrm{d}\lambda$$

and the triangle inequality these can be converted to approximations of the integrals. ∎

5.2.2 The space $W^{m,p}$

Recall from Definition 5.2.2 that a name of a $W^{m,p}$ function contains information about the integrals of the function over dyadic intervals and an L^p-modulus of the highest derivative of f. If $m > 1$ it is not so easy to combine information contained in the L^p-modulus and in the integrals of the function. The key is to iteratively apply the mean value theorem:

Lemma 5.2.7. *Whenever $f \in W^{m,p}$ and $(x_i)_{i \in \{1,\ldots,2^{m-1}\}} \subseteq [0,1]$ are of pairwise distance at least 2^{-m} such that $|f(x_i)| \leq C$, then there exists some $z \in [0,1]$ such that $\left|f^{(m-1)}(z)\right| \leq 2^{m^2-1}C$.*

Proof. Recursively for any $k < m$ construct a family of points $(x_i^k)_{i\in\{1,\dots,2^{m-k-1}\}}$ of pairwise distance at least 2^{-m} such that $\left|f^{(k)}(x_i^k)\right| \leq 2^{k(m+1)}C$.

The case $k = 0$ is taken care of by the assumption. Now assume availability of a family $\left(x_i^{k-1}\right)$ as needed. Since $f^{(k-1)}$ is a continuously differentiable function whenever $k < m$, the mean value theorem states that for any $j \in \{1,\dots,2^{m-k-1}\}$ there is some element $x_j^k \in [x_{2j-1}^{k-1}, x_{2j}^{k-1}]$ such that

$$f^{(k)}(x_j^k) = \frac{f^{(k-1)}(x_{2j-1}^{k-1}) - f^{(k-1)}(x_{2j}^{k-1})}{x_{2j-1}^{k-1} - x_{2j}^{k-1}}$$

and therefore

$$\left|f^{(k)}(x_j^k)\right| \leq 2\cdot 2^{(k-1)(m+1)}C\cdot 2^m = 2^{k(m+1)}C.$$

Obviously, the distance of the points does not decrease.

Setting $k = m-1$ proves the lemma. ∎

Proposition 5.2.8 (Sobolev embedding). *The Sobolev embedding, i.e. the inclusion $W^{m,p} \hookrightarrow W^{m-1,p}$ is polynomial-time computable.*

Proof. Let φ be a $\xi_{m,p}$-name of a function $f \in W^{m,p}$. Compute the value of a $\xi_{m-1,p}$ name of f on a string $\mathbf{a}$ using φ as oracle as follows: To get an L^p-modulus of $f^{(m-1)}$ from the L^p-modulus of $f^{(m)}$ use the previous Lemma: By Lemma 4.2.12 the function $\mu(n) := |\varphi|(n+1)$ is a modulus of continuity of $f^{(m-1)}$. Use the mean value theorem for integrals like in the proof of Theorem 5.2.5 to produce a family of points and a constant C that fulfill the assumption of Lemma 5.2.7. The lemma provides an explicit bound for the values of $f^{(m-1)}$. Combine this with the modulus of continuity like at the end of the proof of Theorem 5.2.5 to get an integer bound Q on $\mathrm{lb}(\left\|f^{(m-1)}\right\|_\infty)$. By Lemma 5.2.4

$$n \mapsto \max\{\mu(n+1), n+Q+1\}$$

is an L^p-modulus of $f^{(m-1)}$. This function can be computed in polynomial-time and the padded return values of φ are valid return values. ∎

Remark 5.2.9. The algorithm specified in this proof accesses the oracle 2^m times. This does not lead to exponential time consumption as m is fixed, however it might lead to large constants in the polynomials for the running time. This can be avoided by providing approximations to the norms of the lower derivatives directly.

The following are generalizations of Theorems 5.2.5 and 5.2.3 and can be proven by induction, where Theorems 5.2.5 and 5.2.3 are the base cases and the previous proposition is the induction step.

Theorem 5.2.10. *For any $m \in \mathbb{N}$ and $1 \leq p < \infty$ the inclusion $W^{m,p} \hookrightarrow \mathrm{L}^p([0,1])$ is polynomial-time computable.*

Theorem 5.2.11. *For any $m \in \mathbb{N}$ and $1 \leq p < \infty$ the inclusion $W^{m,p} \hookrightarrow \mathcal{C}([0,1])$ is polynomial-time computable.*

Finally consider the differentiation operator:

Corollary 5.2.12. *The k-wise differentiation operator*

$$\frac{d^k}{dx^k} : W^{m,p} \to W^{m-k,p}, \quad f \mapsto f^{(k)}$$

is polynomial-time computable for all $k \leq m$.

Proof. By Proposition 5.2.8 obtain approximations to the values of f on dyadic elements. By means of

$$f(x) - f(y) = \int_y^x f' \mathrm{d}\lambda$$

convert these into approximations of the integrals over f'. Iterate this process k-times to obtain approximations to the integrals over $f^{(k)}$. ∎

6 Comparing operators

The previous chapters proved many operators on spaces of integrable functions to be polynomial-time computable. However, not all operations are computable in polynomial time. Recall that up until now we considered the integration operator to be the operator

$$\mathrm{INT_u} : \mathcal{C}([0,1]) \times [0,1]^2 \to \mathbb{R}, \quad (f,x,y) \mapsto \int_x^y f \mathrm{d}\lambda.$$

By currying we can also consider integration as an operator

$$\mathrm{INT} : \mathcal{C}([0,1]) \to \mathcal{C}([0,1]), \quad \mathrm{INT}(f)(x) := \int_0^x f \mathrm{d}\lambda,$$

as has already been done in the introduction of Section 2.2. Recall from Section 2.1.3 (Theorem 2.1.9) that neither of these operators is polynomial-time computable. The goal of this chapter is to compare several operators on function spaces to the above operator.

The tool for comparison are polynomial-time Weihrauch reductions. These reductions do not work well with currying. Thus, this chapter considers integration as an operation on functions. That is: instead of $\mathrm{INT_u}$ which was the important operator throughout Sections 2.2 and 5.1 we consider INT.

6.1 Weihrauch reductions and integration

For this chapter it is necessary to compute on continuous functions on more general compact domains D than $[0,1]$. A definition of a well-behaved second-order representation on these spaces is postponed to Section 6.1.2, however, the next section already needs a notion of computability on these spaces. For compact subsets $D \subseteq \mathbb{R}^d$, the rational polynomials are dense in $\mathcal{C}(D)$. Thus we choose a standard enumeration and equip the space $\mathcal{C}(D)$ with the metric representation from Definition 1.2.2.

6.1.1 Weihrauch reductions

Weihrauch reductions were originally introduced by Klaus Weihrauch as a tool to compare the degree of incomputability of computational tasks. An overview and further references can for instance be found in [BG11b]. Every multivalued function $f :\subseteq \mathbf{X} \rightrightarrows \mathbf{Y}$ between represented spaces $\mathbf{X}$ and $\mathbf{Y}$ corresponds to a computational task. Namely: 'given information about x and the additional assumption $x \in \mathrm{dom}(f)$ find suitable information about some $y \in f(x)$'. What information about x resp. y is provided resp. asked for is reflected in the choice of representations for $\mathbf{X}$ and $\mathbf{Y}$.

Example 6.1.1 (Closed choice on the integers). Let $\mathcal{A}(\mathbb{N})$ be the represented spaces of closed subsets of the integers: The underlying set is the power set of the integers. The representation of $\mathcal{A}(\mathbb{N})$ is defined as follows: A string function $\varphi \in \mathcal{B}$ is a name of the set $\{n \in \mathbb{N} \mid \neg\exists \mathbf{a} : \varphi(\mathbf{a}) = n\}$. That is: a name of a set enumerates the complement of the set.

Consider the multivalued function $\mathcal{C}_{\mathbb{N}} :\subseteq \mathcal{A}(\mathbb{N}) \rightrightarrows \mathbb{N}$ defined on nonempty sets by

$$y \in \mathcal{C}_{\mathbb{N}}(A) \Leftrightarrow y \in A.$$

The corresponding task is 'given an enumeration of the complement of a set of natural numbers and provided that it is not empty, return an element of the set'. $\mathcal{C}_{\mathbb{N}}$ has no computable realizer: Whenever a machine decides that n should be an element of the set, it has only read a finite beginning segment of the enumeration of the complement. The next value might as well be n.

From the point of view of multi-valued functions as computational tasks, it makes sense to compare their difficulty by comparing the corresponding multivalued functions. Weihrauch reductions are a formalization of such a comparison. They define a rather fine pre-order on multivalued functions between represented spaces. Recall that a realizer of a multivalued function $f : \mathbf{X} \to \mathbf{Y}$ between represented spaces is a function $F : \mathcal{B} \to \mathcal{B}$ on the Baire space such that $\delta_{\mathbf{X}}(\varphi) = x$ implies that $\delta_{\mathbf{Y}}(F(\varphi)) \in f(x)$ (cf. Definition 1.2.4 and Figure 1.1).

Definition 6.1.2. Let f and g be partial, multivalued functions between represented spaces. Say that f is **Weihrauch reducible** to g, in symbols $f \leq_W g$, if there are computable functions $K :\subseteq \mathcal{B} \times \mathcal{B} \to \mathcal{B}$ and $H :\subseteq \mathcal{B} \to \mathcal{B}$ such that whenever G is a realizer of g, the function $F(\varphi) := K(\varphi, G(H(\varphi)))$ is a realizer for f.

H is called the **pre-processor** and K the **post-processor** of the Weihrauch reduction. This definition and the nomenclature is illustrated in Figure 6.1. The relation $\leq_W$ is reflexive and transitive. We use $\equiv_W$ to denote that reductions in both directions exist and $<_W$ if the reduction in the other direction does not exist. The equivalence class of a multivalued function with respect to the equivalence relation $\equiv_W$ is called the **Weihrauch degree** of the function.

The Weihrauch degree corresponding to $\mathrm{C}_{\mathbb{N}}$ regularly turns up in classifications of computational tasks (see for instance [BG11a; BBP12; BGH15]). Some examples for tasks that can be classified as being computationally equivalent to $\mathrm{C}_{\mathbb{N}}$ taken form the paper [PS16] are the following:

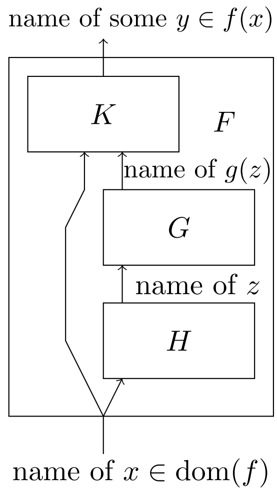


Figure 6.1: Weihrauch reduction

Theorem 6.1.3 ([PS16]). *Let D denote the complex unit disc. The following are Weihrauch equivalent:*

- $\mathrm{C}_{\mathbb{N}}$, *that is: Closed choice on the integers (cf. Example 6.1.1).*

- Sum, *that is: The partial mapping from* $\mathbb{C}^{\mathbb{N}}$ *to* $\mathcal{C}(D)$ *defined on the sequences with radius of convergence strictly larger than one by*

$$\mathrm{Sum}((a_k)_{k\in\mathbb{N}})(x) := \sum_{k\in\mathbb{N}} a_k x^k.$$

I.e. summing power series.

- Diff, *that is: The partial mapping from* $\mathcal{C}(D)$ *to* $\mathbb{C}$ *defined on analytic functions by*

$$\mathrm{Diff}(f) := f'(1).$$

I.e. evaluating the derivative of an analytic function in 1*.*

Weihrauch reducibility can easily be lifted to a complexity level:

Definition 6.1.4 ([KC10]). A Weihrauch reduction $f \leq_W g$ is a **polynomial-time Weihrauch reduction**, if both the pre- and post-processor are polynomial-time computable. We denote the existence of a polynomial-time Weihrauch reduction by $f \leq_P g$.

In [Kaw11; KC10] this kind of reduction is called Turing reduction and denoted by $\leq_T^2$. We denote equivalence and strict reducibility by $\equiv_P$ and $<_W$ and get a preorder (in particular the polynomial-time version is also transitive).

Example 6.1.5 (Currying and integration). To see that $\mathrm{INT_u} \leq_P \mathrm{INT}$, note that the operators

$$\tilde{H} : \mathcal{C}([0,1]) \times [0,1]^2 \to C([0,1]), \quad (f,x,y) \mapsto f$$

and

$$\tilde{K} : \mathcal{C}([0,1]) \times \mathcal{C}([0,1]) \times [0,1]^2 \to \mathbb{R}, \quad (f,g,x,y) \mapsto g(y) - g(x)$$

are polynomial-time computable. Thus let the pre- and post-processors H and K be polynomial-time computable realizers. It is easy to check that H and K can be chosen polynomial-time computable and that this leads to a polynomial-time Weihrauch reduction. Polynomial-time reducibility in the other direction does not hold since the pre-processor can not access the input of the function (cf. the discussion in [Kaw11]).

6.1.2 Sequences

Later in this Chapter we need to compute on sequences of elements from a represented space.

Definition 6.1.6. Let $\mathbf{X}$ be a represented space. Denote by $\mathbf{X}^{\mathbb{N}}$ the represented space of sequences in $\mathbf{X}$, where a length-monotone string function φ is a name of a sequence $(x_m)_{m\in\mathbb{N}} \in \mathbf{X}^{\mathbb{N}}$ if and only if for each $m \in \mathbb{N}$ the function $n \mapsto \varphi(\langle n, m\rangle)$ is a name of $x_m \in \mathbf{X}$.

Thus, a machine computing a sequence in polynomial time is granted time polynomial in the binary length of the index. Depending on the application it can be more appropriate to grant such a machine time polynomial in the value of the index instead. An example of this would be the equivalence of polynomial-time computability of a sequence and of the analytic function corresponding to the sequence. This can be simulated by regular sequences by requiring them to be equal on indices of the same length.

The product construction and the sequence space construction interact in the expected way:

Lemma 6.1.7 (Sequences and products). *Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be represented spaces. Identify* $(\mathbf{X}\times\mathbf{Y})^{\mathbb{N}}$ *with* $\mathbf{X}^{\mathbb{N}}\times\mathbf{Y}^{\mathbb{N}}$ *by the usual isomorphism. The two representations obtained in this way are polynomial-time equivalent.*

Proof. Recall from the definition of the product representation from Definition 1.2.3 and the pairing of string functions from Section 1.1.3 that the translation needs to transform inputs of the form $\langle\langle \mathbf{a},\mathbf{a}\rangle, m\rangle$ and $\langle\langle \mathbf{a}, m\rangle, \langle \mathbf{a}, m\rangle\rangle$ into each other. This can clearly be done in polynomial time. ■

We are particularly interested in the space $\mathcal{C}(\Omega)^{\mathbb{N}}$ of sequences of continuous functions on a common domain. Recall that Theorem 2.1.4 separated polynomial-time computability of a function into a discrete and a topological part. The following result carries this characterization to sequences of functions on more general domains. For later applications we consider the subsets $\overline{B}$, ∂B and $[0,1]^d$ of $\mathbb{R}^d$. For $\overline{B}$ and $[0,1]^d$ it is clear what the dyadic elements are: The dyadic vectors included in the interior of the set. For ∂B the interior is empty and it is not clear how many dyadic vectors lie on this surface. Thus, in this case we regard the **dyadic points** to be the points with dyadic spherical coordinates and also encode them this way.

In the following we consider $\mathbb{R}^d$ to be equipped with the euclidean norm and denote it by $\|\|$. This is for practical reasons and in contrast to the proceeding in the previous Chapters. Note that since all norms on finite dimensional vector spaces are equivalent, a different choice does not change the results qualitatively. Thus we use the notion of a modulus of continuity from Definition 3.1.11.

Theorem 6.1.8. *A sequence* $(f_m)_{m\in\mathbb{N}} \in \mathcal{C}(D)^{\mathbb{N}}$ *of functions with common domain* $D \in \{\overline{B}, \partial B, [0,1]^d\}$ *is polynomial-time computable if and only if both of the following conditions are fulfilled:*

- *There is a polynomial-time computable function* $\phi : \{0,1\}^* \to \mathbb{D}$ *such that whenever* $[\![\mathbf{a}]\!]$ *is a dyadic point of* D *and* $m \in \mathbb{N}$, $1^n \in \omega$, *then*

$$|f_m([\![\mathbf{a}]\!]) - \phi(\langle m, \mathbf{a}, 1^n\rangle)| \le 2^{-n}.$$

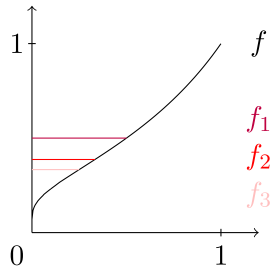
- *There is a polynomial* μ *such that for any* m *the function* $n \mapsto \mu(n + |m|)$ *is a modulus of continuity of* f_m.

We call a function μ such that $n \mapsto \mu(n + |m|)$ is a modulus of continuity of f_m a **modulus function** of the sequence $(f_m)_{m\in\mathbb{N}}$.

No source for this theorem is known to the author, however, it is a very straight forward generalization of the proof from [Ko91] of Theorem 2.1.4 and therefore omitted. The crucial property of the sequence of dyadic points that is needed for the above equivalence to hold is that from a name of some element $x \in D$ of the domain it is possible in polynomial time to find an index of an element of the sequence that approximates x with precision 2^{-n}.

Example 6.1.9. Let $f(x) := 1/(1 - \mathrm{lb}(x))$ be the function from Example 2.1.3 that does not have a polynomial modulus of continuity. Consider the function sequence

$$f_m(x) := \begin{cases} f(x) & \text{if } x \geq \frac{1}{m+1} \\ f(\frac{1}{m+1}) & \text{otherwise.} \end{cases}$$



By the previous theorem this sequence of functions is polynomial-time computable. Restricted to the set $[2^{-m}, 1]$ the functions f_m coincide with the limit function and it is possible to extract a procedure to compute the approximations of the limit function on dyadic points. However, its limit function f is not polynomial-time computable. This is because it does not have a polynomial modulus of continuity. To extract a modulus of continuity of the limit function from the sequence we need a rate of convergence in supremum norm. But the sequence converges slowly in supremum norm.

The above Theorem 6.1.8 with constant sequences can be used to show that the following definition leads to the right complexity classes.

Definition 6.1.10. For $D \in \left\{\overline{B}, \partial B, [0,1]^d\right\}$ let $(d_k)_{k\in\mathbb{N}}$ be a canonical enumeration of the dyadic points. The **standard representation** $\xi_\mathcal{C}$ of $\mathcal{C}(D)$ is defined as follows: A length-monotone string function φ is a name of a function $f \in \mathcal{C}(D)$ if and only if for all $n \in \omega$

$$|f(d_k) - \varphi(\langle k, n\rangle)| \leq 2^{-n}$$

and $|\varphi|$ is a modulus of continuity of f (cf Definition 3.1.11).

Given an operator $T : \mathbf{X} \to \mathbf{Y}$ between represented spaces $\mathbf{X}$ and $\mathbf{Y}$ the operator

$$T^\mathbb{N} : \mathbf{X}^\mathbb{N} \to \mathbf{Y}^\mathbb{N}, \quad (x_n)_{n\in\mathbb{N}} \mapsto (Tx_n)_{n\in\mathbb{N}}$$

is called the **parallelization** of T. While the operator T is computable resp. polynomial-time computable if and only if the operator $T^\mathbb{N}$ is, it need not hold that T and $T^\mathbb{N}$ are polynomial-time Weihrauch equivalent and even their Weihrauch degrees need not coincide. The same is true for $T \times T$. The intuition behind this is that $T \times T$ and $T^{\mathbb{N}}$ allow to apply the operator twice resp. a countable number of times in parallel, while T only allows to apply the operator once. The following properties of these operations are easy to proof:

Lemma 6.1.11. *If $T \leq_P S$, then $T \times T \leq_P S \times S$ and $T^{\mathbb{N}} \leq_P S^{\mathbb{N}}$. Furthermore, for any operator T it holds that $(T \times T)^{\mathbb{N}} \equiv_P T^{\mathbb{N}}$.*

The parallelization operator leads to an operation on the lattice of Weihrauch degrees (cf. for instance to [HP13]).

Example 6.1.12. To see that $\mathrm{INT} \leq_P \mathrm{INT}_{\mathrm{u}}^{\mathbb{N}}$ let the preprocessor H be a machine producing from a function f the sequence $((f, 0, d_n))_{n\in\mathbb{N}}$, where d_n is a standard enumeration of the dyadic numbers. Applying $\mathrm{INT}_{\mathrm{u}}^{\mathbb{N}}$ produces a sequence of real numbers $(y_n)_{n\in\mathbb{N}}$ where $y_n = \int_0^{d_n} f(t)\mathrm{d}t$. The post-processor can read approximations to the values of the antiderivative from the sequence (y_n). It can also find a integer upper bound C of the supremum norm of F by evaluating the modulus of continuity of f and the value of f in zero (this procedure is described in more detail in the proof of Theorem 5.2.5). Then $n + C$ is a modulus of continuity of the antiderivative.

In the upcoming sectionss we need the following:

Lemma 6.1.13 (Multiplication with a sequence). *Let $(f_m)_{m\in\mathbb{N}} \in \mathcal{C}(D)^{\mathbb{N}}$ be a polynomial-time computable sequence. Then the operator*

$$(f_m)_{m\in\mathbb{N}} \cdot (\cdot) : \mathcal{C}(D) \to C(D)^{\mathbb{N}}, \quad g \mapsto (f_m g)_{m\in\mathbb{N}}$$

is polynomial-time computable.

Proof. Since the pointwise multiplication operator from $\mathcal{C}(D) \times \mathcal{C}(D) \to \mathcal{C}(D)$ is polynomial-time computable, so is the multiplication operator $(\mathcal{C}(D) \times \mathcal{C}(D))^{\mathbb{N}} \to \mathcal{C}(D)^{\mathbb{N}}$. The product and sequence constructions commute by Lemma 6.1.7. Therefore the multiplication operator as operator from $\mathcal{C}(D)^{\mathbb{N}} \times \mathcal{C}(D)^{\mathbb{N}} \to C(D)^{\mathbb{N}}$ is polynomial-time computable. It is left to note that the inclusion of $\mathcal{C}(D)$ into $\mathcal{C}(D)^{\mathbb{N}}$ mapping a function to the constant sequence is polynomial-time computable. Now the lemma follows by fixing the first argument to a polynomial-time computable element. ■

6.1.3 The complexity of integration

As formalization of the problem of integrating a continuous function choose the operator INT defined by

$$\mathrm{INT} : \mathcal{C}([0,1]) \to \mathcal{C}([0,1]), \quad \mathrm{INT}(f)(x) = \int_0^x f \mathrm{d}\lambda. \tag{INT}$$

This section uses polynomial-time Weihrauch reductions to classify the difficulty of integration. Namely, it is proven to be polynomial-time Weihrauch equivalent to the following operator on the Baire space (compare for instance [Kaw11]).

Definition 6.1.14. Let $\#\mathrm{EXIST} : \mathcal{B} \to \mathcal{B}$ be defined by

$$\#\mathrm{EXIST}(\varphi)(\mathbf{a}) := \#\left\{\mathbf{b} \mid |\mathbf{b}| \leq |\mathbf{a}| \text{ and } \varphi(\langle \mathbf{b}, \mathbf{a} \rangle) \neq \varepsilon\right\}.$$

From the definition it is immediate that #EXIST preserves polynomial-time computability if and only if $\mathcal{FP} = \#\mathcal{P}$. Moreover, #EXIST is not polynomial-time computable: An algorithm computing it must query the oracle for $\varphi(\langle \mathbf{b}, \mathbf{a} \rangle)$ for each of the $2^{|\mathbf{a}|}$ values of $\mathbf{b}$. Otherwise a false output could be forced by replacing φ by a function whose output differs in exactly one of the places the machine did not look at. #EXIST is, however, polynomial-space computable, as each query and the counting of non-zero return values can easily be done in polynomial space. In particular it follows, that the higher-order complexity classes of polynomial-time and -space computable functions are easy to separate. This is in contrast to situation in classical complexity theory (see the beginning of Section 2.1.3, in particular Figure 2.3 on page 25) where question whether or not the classes $\mathcal{P}$ and $\mathcal{PSPACE}$ can be separated is unsolved and regarded to be notoriously difficult. Furthermore:

Lemma 6.1.15. $\#\mathrm{EXIST} \equiv_P \#\mathrm{EXIST} \times \#\mathrm{EXIST} \equiv_P \#\mathrm{EXIST}^{\mathbb{N}}$.

Proof. It is clear that $\#\mathrm{EXIST} \leq_P \#\mathrm{EXIST} \times \#\mathrm{EXIST} \leq_P \#\mathrm{EXIST}^{\mathbb{N}}$. Therefore, it suffices to specify a proof that $\#\mathrm{EXIST}^{\mathbb{N}} \leq_P \#\mathrm{EXIST}$. Let the pre-processor be defined by

$$H(\varphi)(\mathbf{a}) := \begin{cases} \varphi(\langle \mathbf{c}, \langle \mathbf{b}, \mathbf{d} \rangle \rangle) & \text{if } \mathbf{a} = \langle \langle \mathbf{c}, \mathbf{b} \rangle, \langle \mathbf{c}, \mathbf{d} \rangle \rangle \\ \varepsilon & \text{otherwise.} \end{cases}$$

Thus,

$$\#\{\mathbf{b} \mid |\mathbf{b}| \leq |\langle \mathbf{c}, \mathbf{d} \rangle|, H(\varphi)(\langle \mathbf{b}, \langle \mathbf{c}, \mathbf{d} \rangle \rangle) \neq \varepsilon\} = \#\{\mathbf{b} \mid |\mathbf{b}| \leq |\mathbf{d}|, \varphi(\langle \mathbf{c}, \langle \mathbf{b}, \mathbf{d} \rangle \rangle) \neq \varepsilon\}$$

and the right hand side is exactly the value of $\#\mathrm{EXIST}^{\mathbb{N}}(\varphi)(\langle \mathbf{c}, \mathbf{d} \rangle)$ while the left hand side is the value of $\#\mathrm{EXIST}(\langle c, d \rangle)$. This means that $(\varphi, \psi) \mapsto \psi$ is an appropriate post-processor. ∎

All of the above properties are preserved under polynomial-time Weihrauch reductions.

For applications, the following integration operators are often more convenient to use than the operator INT:

Definition 6.1.16. For $d, d' \in \mathbb{N}$ denote by $\mathrm{INT}_d^{d'}$ the operator from $\mathcal{C}([0,1]^d \times [0,1]^{d'})$ to $\mathcal{C}([0,1]^d)$ that integrates the last d' variables over the whole interval.

Recall that for any represented space $\mathbf{X}$ the space $\mathbf{X}^{\mathbb{N}}$ of sequences in $\mathbf{X}$ is a represented space (cf. Definition 6.1.6). Also recall that for an operator $T : \mathbf{X} \to \mathbf{Y}$, the parallelization $T^{\mathbb{N}} : \mathbf{X}^{\mathbb{N}} \to \mathbf{Y}^{\mathbb{N}}$ is defined by pointwise application. The following result is an important building block of the rest of the section. Very similar results can be found in [Kaw11, §4.3.2].

Theorem 6.1.17. *The following are polynomial-time Weihrauch equivalent:*

1. #EXIST.
2. $(\mathrm{INT}_d^{d'})^{\mathbb{N}}$ *for any* $d, d' \geq 1$.
3. INT.
4. $\mathrm{INT}\,|_{L_{1,0}}$*: The restriction of*INT *to the Lipschitz* 1 *functions that vanish in zero.*

Proof. Build a circle of polynomial-time Weihrauch reductions:

$(\mathrm{INT}_d^{d'})^{\mathbb{N}} \leq_P$ #EXIST: It suffices to proof $\mathrm{INT}_d^{d'} \leq_P$ #EXIST: By Lemma 6.1.7 this implies that $(\mathrm{INT}_d^{d'})^{\mathbb{N}} \leq_P \#\mathrm{EXIST}^{\mathbb{N}}$ and by Lemma 6.1.15 the right hand side is polynomial-time Weihrauch equivalent to #EXIST. Note that it is possible to extract a bound on the absolute value of a function from a name. Thus the function can be shifted and scaled to take values between 0 and 1. This can easily be reversed by the post-processor.

For later use we give the pre-processor a special name. Define the pre-processor Ψ as follows: Let $\Psi(\varphi)(\mathbf{a}) = \mathtt{1}$ if and only if

$$\mathbf{a} = \langle\langle \mathbf{s}, \mathbf{t}\rangle, \langle \mathbf{q}, \mathtt{1}^n\rangle\rangle,$$

such that $\mathbf{s}, \mathbf{t}$ and $\mathbf{q}$ are encodings of dyadic points s, t and q. Furthermore, require s and t to be strictly larger than 0, s to be strictly smaller than 1 and the encodings $\mathbf{s}$ and $\mathbf{t}$ to be such that the denominator has length $|\varphi|(n)$ resp. n. Finally, demand that

$$0 < [\![\mathbf{t}]\!] \leq [\![\varphi(\langle\langle \mathbf{q}, \mathbf{s}\rangle, \mathtt{1}^n\rangle)]\!]\,.$$

If any of this is false, set $\Psi(\varphi)(\mathbf{a}) := \mathtt{0}$. All of these conditions can be checked in polynomial time, therefore Ψ is polynomial-time computable. Note that the conditions imply for a $\xi_{\mathcal{C}}$-name φ of a function f that

$$0 < t < f(q,s) - 2^{-n} \Rightarrow \varphi(\mathbf{a}) = \mathtt{1} \quad \text{and} \quad f(q,s) + 2^{-n} < t \Rightarrow \varphi(\mathbf{a}) = \mathtt{0}.$$

Figure 6.2 visualizes $\Psi(\varphi)$.

Calling #EXIST on $\Psi(\varphi)$ returns the function

$$\psi(\mathbf{a}) := \#\{\mathbf{b} \mid |\mathbf{b}| \leq |\mathbf{a}| \text{ and } \Psi(\varphi)(\langle \mathbf{b}, \mathbf{a}\rangle) \neq 0\}.$$

When given input $\mathbf{a} = \langle \mathbf{q}, \mathtt{1}^n\rangle$ such that $|\mathbf{a}| \geq \left|\langle \mathtt{1}^{|\varphi|(n)}, \mathtt{1}^n\rangle\right|$ the return value is the number of colored cubes below the function in Figure 6.2. The conditions on $\mathbf{s}$ and $\mathbf{t}$ make sure that each cube is counted exactly once. It is straightforward to check that multiplying this number by the size of the cubes gives an approximation to the value of the integrated function and can be done by the post-processor in polynomial time. Thus it is only left to acquire a modulus of continuity for the integrated function. Note that any modulus of continuity μ for f is also one for $\mathrm{INT}_d^{d'} f$: If we assume $\|x - x'\| \leq 2^{-\mu(n)}$, then

$$\left\| \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x' \\ y \end{pmatrix} \right\| = \|x - x'\| \leq 2^{-\mu(n)},$$

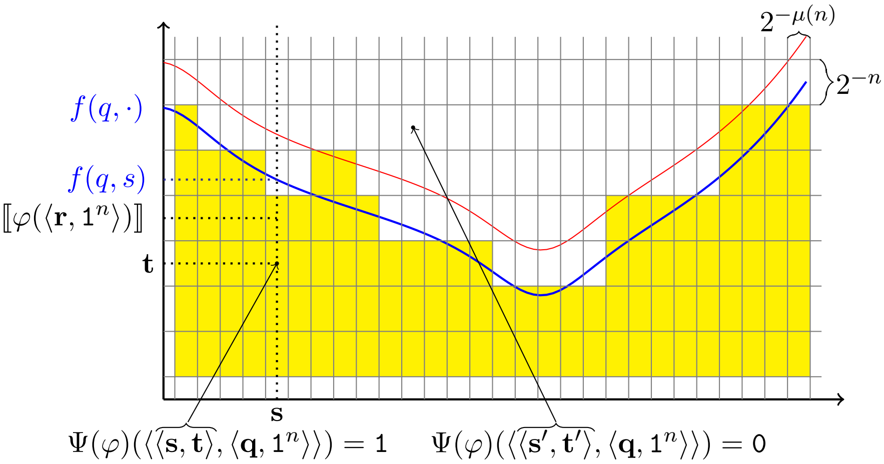


Figure 6.2: The boxes assigned value 1 by H are colored. If the center of the box is above the upper line, it is required to be white, below the blue required to be colored. If the center is between the lines, both is allowed.

and therefore

$$|g(x) - g(x')| \leq \int_{[0,1]^{d'}} |f(x,y) - f(x',y)|\mathrm{d}y \leq \int_{[0,1]^{d'}} 2^{-n}\mathrm{d}y = 2^{-n}.$$

$\mathrm{INT} \leq_P (\mathrm{INT}_d^{d'})^{\mathbb{N}}$: Note that $\mathrm{INT}_1^1 \leq_P \mathrm{INT}_d^{d'} \leq_P (\mathrm{INT}_d^{d'})^{\mathbb{N}}$. Therefore, it is enough to specify a reduction $\mathrm{INT} \leq_P \mathrm{INT}_1^1$. Let the pre-processor map a function f to the function $g(x) := f(\min\{x_1, x_2\})$. Then

$$(\mathrm{INT}_1^1\, g)(y) = \int_0^y f(t)dt + f(y) \cdot (1 - y).$$

Therefore, the post-processor can easily obtain the integral.

$\mathrm{INT}\,|_{L_{1,0}} \leq_P \mathrm{INT}$: is trivial.

$\#\mathrm{EXIST} \leq_P \mathrm{INT}\,|_{L_{1,0}}$: Let the pre-processor H produce from a string function φ a name of the function $h_\varphi \in L_{1,0}$ depicted in Figure 6.3. This can be done in polynomial-time: The function is Lipschitz 1 by construction so a modulus of continuity is easy to get hold of, and for evaluation in a dyadic point exactly one query has to been made and then the function value can explicitly be computed. The post-processor reads the values of the antiderivative of this function in two points to compute the integral over the interval corresponding to the input string $\mathbf{a}$, and reads the number of non-zero values of φ off from the binary expansion of the result. ∎

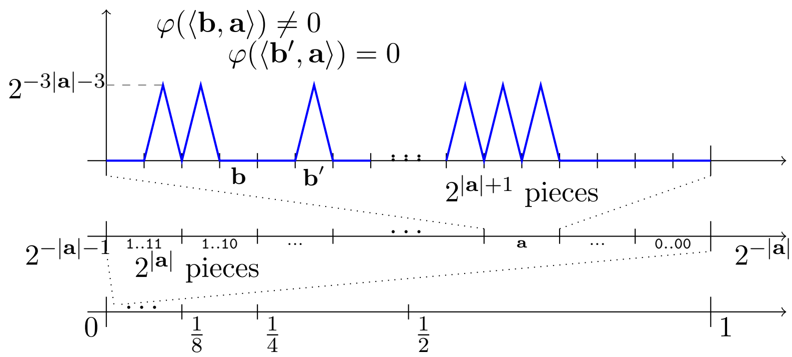


Figure 6.3: The function h_φ.

The above remains true if the integration operator is further restricted to the smooth functions. The triangle bump functions h_φ in the proof can be replaced by smooth bump functions.

It follows that the integration operator also has the properties we listed in and before Lemma 6.1.15 for the operator #EXIST.

Corollary 6.1.18. *The following hold for the integration operator:*

- $\mathrm{INT} \equiv_P \mathrm{INT} \times \mathrm{INT} \equiv_P \mathrm{INT}^{\mathbb{N}}$.
- INT *is not polynomial-time computable but polynomial-space computable.*
- INT *maps polynomial time computable functions to polynomial time computable functions if and only if $\mathcal{FP} = \#\mathcal{P}$.*

It is also possible to give a direct proof where the pre-processor scales multiple input functions and stitches them together to one continuous function and the post processor chooses the right intervals to integrate over.

Another corollary that follows from Example 6.1.5 and the previous is:

Corollary 6.1.19. $\mathrm{INT}_{\mathbf{u}}^{\mathbb{N}} \equiv_P \mathrm{INT}$.

6.1.4 Multiplication operators

This section considers multiplication operators on spaces of functions. The pointwise product of two continuous functions is well defined and a continuous function again. Moreover, the operator mapping a pair of continuous functions to their product is polynomial-time computable with respect to the standard representation. Multiplication of functions from L^p-spaces is more problematic as these spaces are not closed under multiplication. However, all L^p-spaces are closed under multiplication with essentially bounded functions. It turns out that even multiplying with a very simple fixed function is as difficult as integrating a continuous function and thus not possible in polynomial time.

Theorem 6.1.20. *Equip* $\mathrm{L}^1([0,1])$ *with the singular representation* ξ_s*. Then the following are polynomial-time Weihrauch equivalent:*

- *The operator* $x\cdot : \mathrm{L}^1([0,1]) \to \mathrm{L}^1([0,1])$ *mapping a function* f *to the function* $x \cdot f$*.*
- *The integration operator* $\mathrm{INT} : \mathcal{C}([0,1]) \to \mathcal{C}([0,1])$*.*

Proof. Fix some $f \in \mathrm{L}^1$ and set $F(x) := \int_0^x f \mathrm{d}\lambda$. By partial integration

$$\int_a^b x f \mathrm{d}\lambda = bF(b) - aF(a) - \int_a^b F \mathrm{d}\lambda. \tag{6.1}$$

To polynomial-time Weihrauch reduce the multiplication with the identity to the integration operator, first note that a singularity modulus of f is also a singularity modulus of $x \cdot f$.

Define the pre-processor H as a realizer of the function mapping f to F. This operator is polynomial-time computable due to the definitions of the representations on L^1 and the continuous functions respectively. Applying the integration operator results in a name of an antiderivative of F. Using this and (6.1), the post-processor can easily compute the integrals over the function $x \cdot f$.

For the reduction in the other direction use Theorem 6.1.17 to replace the integration operator by its restriction to the Lipschitz 1 functions that vanish in zero. These functions are in particular absolutely continuous, therefore let the pre-processor be a realizer of the operator sending a given continuous function F to the integrable function f such that $F(x) = \int_0^x f \mathrm{d}\lambda$. The modulus of continuity of F is a singularity modulus of this function, thus this operator is polynomial-time computable. Applying the multiplication operator results in an L^1-name of $x \cdot f$. Using (6.1) the integrals over this function can be used by the post-processor to compute the values of an antiderivative of F in dyadic points. A modulus of continuity of the antiderivative is given by $n \mapsto n + \lceil \mathrm{lb}(\|F\|_\infty) \rceil$. Since a bound on the supremum norm of F can be computed from its name in polynomial time, a modulus of continuity can be produced by the postprocessor. ■

Let Ω be a measurable bounded set. Recall that the dual space of $\mathrm{L}^p(\Omega)$ is $\mathrm{L}^q(\Omega)$, where $q = \left(1 - \frac{1}{p}\right)^{-1}$. A function $g \in \mathrm{L}^q(\Omega)$ is identified with the functional

$$\phi_g : \mathrm{L}^p(\Omega) \to \mathbb{R}, \quad f \mapsto \int fg \mathrm{d}\lambda.$$

That this functional is well defined, i.e. $f \cdot g \in \mathrm{L}^1$ can for instance be seen using Hölder's inequality from Theorem 4.2.1. In particular the following multiplication operator is well defined and worth investigations:

$$\mathrm{mult}_{p,\Omega} : \mathrm{L}^p(\Omega) \times \mathrm{L}^q(\Omega) \to \mathrm{L}^1(\Omega), \quad (f,g) \mapsto f \cdot g$$

Theorem 6.1.21. *Let* Ω *be measurable and bounded with non-empty interior. Then* INT *is polynomial-time Weihrauch reducible to* $\mathrm{mult}_{p,\Omega}$*.*

Proof. First consider the case $\Omega = [0,1]$. Let φ be a name of a function $F \in \mathcal{C}([0,1])$. Since a bound on the values of the function can be computed in polynomial-time we may assume that the function takes values in the unit interval. Recall the functions

$$\Psi :\subseteq \Sigma^{**} \to \Sigma^{**} \quad \text{and} \quad h_{(\cdot)} : \Sigma^{**} \to \mathcal{C}([0,1]), \quad \psi \mapsto h_\psi$$

from the proof of Theorem 6.1.17 (visualized in Figures 6.2 and 6.3). Recall that the integrals of F can be computed from $\Psi(\varphi)$ by counting the nonzero values and that the nonzero values of ψ can be counted by integrating h_ψ. Therefore, the integrals of F can be obtained from the integrals of $h_{\Psi(\varphi)}$. By definition h_ψ is a Lipschitz one function. Thus, its weak derivative h'_ψ is bounded and contained in L^p for any p. The function $(h'_\psi)^2$ can easily be described: it arises from h_ψ by replacing each of the hats with a characteristic function of a small interval. In particular the integrals of h_ψ can be read from the integrals of $(h'_\psi)^2$.

Thus, to get access to the integrals over F we may feed L^p- resp. L^q-names of $h'_{\Psi(\varphi)}$ to each of the inputs of $\mathrm{mult}_{p,\Omega}$. The approximations the integrals of $h'_{\Psi(\varphi)}$, that is the values of $h_{\Psi(\varphi)}$ can be obtained from φ in polynomial time. The assignment $n \mapsto 4(\lceil p \rceil n + 4)$ is an L^p-modulus of any h'_ψ. Feeding these names to $\mathrm{mult}_{p,\Omega}$ results in an L^1-name of $(h'_{\Psi(\varphi)})^2$. Since the post-processor is granted time that is allowed to depend on the modulus of continuity of the input function F, it can read the integrals of F off from the integrals of this function in polynomial time.

For an arbitrary set Ω first there exists a small cube that is included in the domain. Extend the functions from above independently of the additional variables and scale them to fit within the cube. The argument from above can be repeated. ∎

The technique of proof can be used to prove a variation of Theorem 6.1.20. It does not imply the former result since it does not use the singular representation for L^1.

Corollary 6.1.22. *The following are polynomial-time Weihrauch equivalent:*

- *The operator* $x\cdot : \mathrm{L}^p([0,1]) \to \mathrm{L}^p([0,1])$ *mapping a function f to the function $x \cdot f$.*
- *The integration operator* $\mathrm{INT} : \mathcal{C}([0,1]) \to \mathcal{C}([0,1])$.

Proof. The proof can basically be copied from Theorem 6.1.20 with the following modifications: In the first implication compute an L^p-modulus of $x \cdot f$ from an L^p-modulus of f. In the second implication replace F by the function $h_{\Psi(\varphi)}$ (as in the proof of Theorem 6.1.21) to be able to obtain an L^p-modulus of the weak derivative in an straightforward way. ∎

This Corollary in turn implies Theorem 6.1.21 for $\Omega = [0,1]$ as the function x is polynomial-time computable in any L^p. However, the proof of the theorem is more direct and does not use partial integration.

6.2 Poisson's Equation

This section discusses a more elaborate example of an operator that is polynomial-time Weihrauch equivalent to the integration operator. Namely a solution operator of a simple, yet very important partial differential equation: Poisson's Equation. Here we recollect some facts and background about Laplace's and Poisson's Equation and adapt and improve known results on the regularity of their solutions. Only the case of dimensions $d > 1$ is dealt with. The next chapter uses the content of this chapter to show that the solution operator is polynomial-time Weihrauch equivalent to the integration operator from (INT) on page 102. Recall that the **Laplace operator** Δ on an open set $\Omega \subseteq \mathbb{R}^d$ is defined by

$$\Delta : C^2(\Omega) \to C(\Omega), \quad u \mapsto \sum_{i=1}^{d} D_{ii} u = \sum_{i=1}^{d} \frac{\partial^2 u}{\partial x_i^2}.$$

The equation $\Delta u = f$ is called **Poisson's Equation**, the special case $f = 0$ is called **Laplace's Equation**.

The **Dirichlet Problem** for these equations is to find, given functions $g : \partial\Omega \to \mathbb{R}$ and $f : B \to \mathbb{R}$, a function $u : \overline{\Omega} \to \mathbb{R}$ satisfying the equation on Ω and restricting to g on the boundary. I.e. the Dirichlet Problem for Poisson's Equation is to find a function u fulfilling

$$\Delta u = f \text{ in } \Omega, \quad u|_{\partial\Omega} = g. \tag{DPP}$$

The Dirichlet Problem is called **homogeneous** if $g = 0$. By linearity of the Laplace operator finding a solution of (DPP) can be divided into two problems: The Dirichlet Problem for Laplace's Equation and the homogeneous Dirichlet Problem for Poisson's Equation.

The Dirichlet Problem for Laplace's Equation and Poisson's Equation regularly turns up in practical problems like finding the gravitational potential to a given mass density or the electrical potential to a given charge density. The solution theory is well developed and this chapter makes extensive use of the results already known from analysis.

6.2.1 The solution operators

Fix some dimension $d > 1$. From now on consider as domain Ω the d-dimensional open unit ball B.

For the Dirichlet Problem for Laplace's Equation

Consider the Dirichlet Problem for Laplace's Equation on the unit ball B in $\mathbb{R}^d$:

$$\Delta u = 0 \text{ in } B, \quad u|_{\partial B} = g. \tag{DPL}$$

where $g : \partial B \to \mathbb{R}$ is some function on the boundary of B.

It is known that a solution u of the above equation exists and is unique whenever g is continuous. Thus there is a solution operator

$$\mathrm{sol}_L : C(\partial B) \to \mathcal{C}(\overline{B}) \cap \mathcal{C}^2(B), \quad g \mapsto u.$$

That is, sol_L is the unique operator fulfilling

$$\Delta \, \mathrm{sol}_L(g) = 0 \text{ in } B, \quad \mathrm{sol}_L(g)|_{\partial B} = g.$$

It is possible to give a more explicit description of this operator. Let σ denote the surface measure on the d-dimensional unit sphere.

Theorem 6.2.1 (Poisson integral). *For $x \in B$ the formula*

$$\mathrm{sol}_L(g)(x) = \int_{\partial B} K(x,y) \cdot g(y) \mathrm{d}\sigma(y), \tag{PI}$$

holds, where

$$K(x,y) = \frac{1 - \|x\|^2}{d \cdot \lambda(B) \cdot \|x-y\|^d}$$

is the ***Poisson Kernel*** *(see Figure 6.4).*

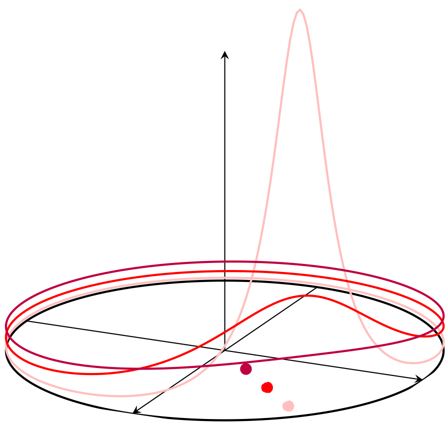


Figure 6.4: The Poisson kernel for different values of x plotted over y. ($d = 2$).

A proof can for example be found in [GT01, Theorem 2.6] and is reevaluated for the proof of Lemma 6.2.5 below. Functions fulfilling Laplace's Equation on an open set are known as harmonic functions and well investigated. In particular it is known that these functions are real analytic on their domain.

For the homogeneous Dirichlet Problem for Poisson's Equation

We turn to the homogeneous Dirichlet Problem for Poisson's Equation on the unit ball:

$$\Delta u = f \text{ in } B, \quad u|_{\partial B} = 0. \tag{HDP}$$

Note that for this equation to make sense, f need not be continuous. However, to talk about computability and complexity of the solution operator, f should be considered an element of a represented space. The most general reasonable choice for such a space would be L^1. However, a closed integral formula like the Poisson Integral from Theorem 6.2.1 for sol_L is only available under further assumptions about the function f. A possible restriction is $f \in \mathrm{L}^\infty$. Since L^∞, as it is not separable, does not allow a reasonable representation we further restrict to $f \in \mathcal{C}(\overline{B})$. Recall that this space is equipped with the standard representation from Definition 6.1.10.

In contrast to the situation for Laplace's Equation it is known that there are continuous functions f such that no twice continuously differentiable solution to Poisson's Equation exists. This is known as **Petrini's example** and can be overcome by considering the solution to be an element of a Sobolev space. A reader not familiar with this concept may simply understand **weak solution corresponding to** f to mean the function according to the following solution formula (cf. for instance [GT01, §2.4+2.5]):

Theorem 6.2.2 (The Green's potential). *The solution operator* $\mathrm{sol}_P^h : \mathcal{C}(\overline{B}) \to \mathcal{C}(\overline{B})$ *of homogeneous Dirichlet Problem for Poisson's Equation is given by*

$$\mathrm{sol}_P^h(f)(x) := \int_B G(y,x) \cdot f(y)\mathrm{d}y \tag{GP}$$

where

$$G(y,x) = \tilde{\Gamma}(\|x-y\|) - \tilde{\Gamma}\left(\|x\| \left\|y - \frac{x}{\|x\|^2}\right\|\right), \tag{G}$$

is the ***Green's function*** *and*

$$\tilde{\Gamma}(r) = \begin{cases} -\frac{1}{2\pi}\ln(r) & \text{if } d = 2, \\ \frac{1}{d\cdot(d-2)\cdot\lambda(B)\cdot r^{d-2}} & \text{if } d > 2 \end{cases}.$$

In Figure 6.5 the Green's function G is depicted for the case $d = 2$, and in dependence on y for a fixed x.

These integrals, taken from potential theory, make sense for any integrable and bounded function f. A reader familiar with potential theory may recognize the radially symmetric extension $\Gamma(x) := \tilde{\Gamma}(\|x\|)$ of $\tilde{\Gamma}$ as the **fundamental solution** of the d-dimensional Laplacian, and see that **Green's function** is obtained from it by the method of image charges. Again, the reader not familiar with the concepts may simply consider these names.

While $\mathrm{sol}_P^h(f) \in C^1(B)$ (see Section 6.2.2), the second derivatives need not exist. Also note that the right hand side of Equation (GP), is well defined whenever $f \in \mathrm{L}^\infty$.

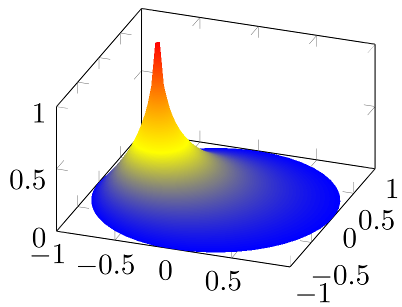


Figure 6.5: The Green's function for $x = \left(-\frac{1}{2}, 0\right)$ plotted over y. ($d = 2$).

Theorem 6.2.3. *Whenever f is bounded and integrable and the corresponding weak solution is twice continuously differentiable, it is a solution of the homogeneous Dirichlet Problem for Poisson's Equation* (HDP) *on page 110.*

A proof can for instance be found in [GT01].

6.2.2 Obtaining the moduli

There seems to be a straightforward way to construct a Weihrauch reduction that computes one of the solution operators sol_L or sol_P^h from the integration operator by using the concrete integral formulas from Theorems 6.2.1 and 6.2.2. Note however, that both the Poisson Kernel an the Green's function have singularities and thus the integration operator can not be applied directly. This can be overcome by truncating the singularities and obtaining a sequence of approximate solutions converging to the real solution. Recall from Theorem 6.1.17 that instead of INT we may reduce to the operator $(\mathrm{INT}_d^{d'})^{\mathbb{N}}$. Thereby it is possible to obtain approximations to the values of a solution on dyadic points in the unit ball. However, we encounter a situation very similar to Example 6.1.9 where fast convergence can only be proven on growing subsets that exhaust the unit ball. Thus, the modulus of continuity has to be acquired separately. This is what is done in this section.

For Laplace's Equation

Note that, while being analytic on B, the solution $\mathrm{sol}_L(g)$ of the Dirichlet Problem for Laplace's Equation can not expected to be differentiable on $\overline{B}$ if the boundary data g is not differentiable. Recall that we use the euclidean norm $\|\cdot\|$ on $\mathbb{R}^d$. To control the behavior of the solution in proximity of the boundary employ the following well-known gradient estimate:

Theorem 6.2.4 (Gradient estimate). *For any solution u of* (DPL) *on page 109*

$$\|Du(x)\| \leq \frac{d \cdot \|g\|_\infty}{\mathrm{dist}(x, \partial B)}.$$

Here Du denotes the gradient, that is the vector containing the partial derivatives $D_i u$ and $\mathrm{dist}(x, \partial B) := \inf\{\|x - y\| \mid y \in \partial B\}$ is the usual distance function. A proof can be found in [GT01, Theorem 2.10].

The next Lemma extracts a modulus of continuity for $\mathrm{sol}_L(g)$ from one of g:

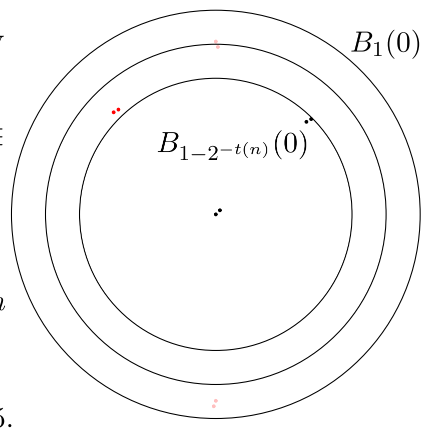


Figure 6.6: The case distinction of Lemma 6.2.5: Case 1, Case 2 and both cases.

Lemma 6.2.5. *If μ is a modulus of continuity of $g \in \mathcal{C}(\partial B)$, then the function*

$$n \mapsto d\mu(n+2) + 2n + C$$

is a modulus of continuity of the unique solution $\mathrm{sol}_L(g)$ *of* (DPL) *on page 109, where*

$$C = \left\lceil \mathrm{lb}\left(\max\left\{\frac{2\|g\|_\infty}{d\lambda(B)}, 1\right\} \max\{d\,\|g\|_\infty, 1\}\right)\right\rceil + d + 5.$$

The proof closely follows the standard proof of Theorem 6.2.1 as it can for instance be found in [Bre11].

Proof. Call the function from the statement ν. To verify that ν is a modulus of continuity of $u := \mathrm{sol}_L(g)$ choose some arbitrary $x, y \in \overline{B}$ such that $\|x-y\| \le 2^{-\nu(n)}$ and distinguish the cases that both points are close to the boundary and that both points lie well within B. For this set

$$\begin{aligned} t(n) &:= d\mu(n+2) + n + \mathrm{lb}\left(\max\left\{\frac{2\|g\|_\infty}{d\cdot\lambda(B)}, 1\right\}\right) + d + 4 \\ &\le \nu(n) - \mathrm{lb}(\max\{d\|g\|_\infty, 1\}) - n - 1 \qquad (6.2) \\ &\le \nu(n) - 1. \qquad (6.3) \end{aligned}$$

Note that (6.3) and $\|x-y\| \le 2^{-\nu(n)}$ together imply that either $\|x\|, \|y\| \le 1 - 2^{-t(n)}$ or $\|x\|, \|y\| \ge 1 - 2^{-t(n)+1}$ or both.

Case 1 ($\|x\|, \|y\| \le 1 - 2^{-t(n)}$): Apply Theorem 6.2.4 and use (6.2) to obtain

$$\begin{aligned} |u(x) - u(y)| &\le \sup\left\{\|(Du)(z)\| \,\middle|\, z \in B_{1-2^{-t(n)}}(0)\right\} \cdot \|x-y\| \\ &\le \frac{d\cdot\|g\|_\infty}{2^{-t(n)}}\|x-y\| \le d\cdot\|g\|_\infty 2^{-\nu(n)+t(n)} < 2^{-n}. \end{aligned}$$

Case 2 ($\|x\|, \|y\| \ge 1 - 2^{-t(n)+1}$): Consider the element $z := \frac{x+y}{\|x+y\|} \in \partial B$ and observe that

$$\begin{aligned} \|x-z\| &= \left\|x - \frac{x+y}{\|x+y\|}\right\| = \left\|\frac{x-y}{2} + \frac{x+y}{2} - \frac{x+y}{\|x+y\|}\right\| \\ &\le 2^{-\nu(n)-1} + \frac{2-\|x+y\|}{2} \le 2^{-\nu(n)-1} + 2^{-t(n)+1} + 2^{-\nu(n)-1} \\ &\le 2^{-t(n)+2} \le 2^{-\mu(n+2)-1}, \end{aligned}$$

and therefore for any $t \in \partial B$ with $\|t-z\| \ge 2^{-\mu(n+2)}$

$$\|x-t\| = \|x-z-(t-z)\| \ge |\|t-z\| - \|x-z\|| \ge 2^{-\mu(n+2)-1}. \qquad (6.4)$$

Since the unique solution of (DPL) from page 109 with constant boundary condition $u|_{\partial B} \equiv 1$ is given by the constant function $u \equiv 1$, it must hold that

$$\int_{\partial B} K(x,t)\mathrm{d}\sigma(t) = 1 \qquad (6.5)$$

for any $x \in B$. Therefore, and since $u|_{\partial B} = g$ and thus $u(z) = g(z)$,

$$\begin{aligned} |u(x) - u(z)| &= \left|\int_{\partial B} K(x,t)\cdot\big(g(t)-g(z)\big)\mathrm{d}\sigma(t)\right| \\ &\le \int_{\partial B} K(x,t)|g(t)-g(z)|\mathrm{d}\sigma(t) \\ &= \int_{\partial B\cap B_{2^{-\mu(n+2)}}(z)} K(x,t)|g(t)-g(z)|\mathrm{d}\sigma(t) \\ &\quad + \int_{\partial B\cap B_{2^{-\mu(n+2)}}(z)^c} K(x,t)|g(t)-g(z)|\mathrm{d}\sigma(t). \end{aligned}$$

The first of these integrals can be estimated by using that μ is a modulus of continuity for g and (6.5). Use the definition of the Poisson Kernel K for the second term, (6.4) and estimate the remaining integral

$$\begin{aligned} |u(x) - u(z)| &< 2^{-n-2} + \frac{2\|g\|_\infty}{d \cdot \lambda(B)} \frac{1 - \|x\|^2}{2^{-d\mu(n+2)-d}} \\ &\leq 2^{-n-2} + \frac{2\|g\|_\infty}{d \cdot \lambda(B)} 2^{-t(n)+2+d\mu(n+2)+d} \leq 2^{-n-1}. \end{aligned}$$

Since the same reasoning works with x replaced by y, one finally gets

$$|u(x) - u(y)| \leq |u(x) - u(z)| + |u(y) - u(z)| < 2^{-n}.$$

This completes the proof. ■

For Poisson's Equation

It often makes sense to divide the weak solution w into the two summands appearing in (G) on page 111:

$$\mathrm{sol}_P^h(f)(x) = \underbrace{\int_B \Gamma(x-y) \cdot f(y) \mathrm{d}y}_{=:v(x)} - \underbrace{\int_B \tilde{\Gamma}\left(\|x\| \cdot \left\| y - \frac{x}{\|x\|^2} \right\| \right) \cdot f(y) \mathrm{d}y}_{=:w_0(x)}, \tag{N}$$

where v is also called the **Newtonian potential**. Note that the defining formula makes sense on the whole space. A proof that the Newtonian potential v is continuously differentiable whenever $f \in \mathrm{L}^\infty$ can be found in [GT01, Lemma 4.1]. Our goal is to explicitly state the dependence of the modulus of continuity of $\mathrm{sol}_P^h(f)$ on the function f, this is an easy consequence of the cited lemma:

Lemma 6.2.6. *Whenever $f \in \mathrm{L}^\infty$, then the function*

$$n \mapsto n + \lceil \mathrm{lb}(\max\{\|f\|_\infty, 1\}) \rceil + 1$$

is a modulus of continuity of the Newtonian potential v from Equation (N).

Proof. From [GT01, Lemma 4.1] it is known that the directional derivatives of the Newton potential v can be computed pulling the directional derivative under the integral. Explicit calculations show that $\|f\|_\infty$ is a bound on the maximal value of all directional derivatives an thus a Lipschitz constant. This Lipschitz constant translates to a linear modulus of continuity (compare to Example 3.1.12). ■

While the Newton potential need not be twice differentiable, it is twice weakly differentiable and fulfills $\Delta v = f$ in the sense of weak derivatives. Therefore, from (N) it can be seen that $w_0 := \mathrm{sol}_P^h(f) - v$ is the solution of the Dirichlet Problem for Laplace's Equation with boundary condition $v|_{\partial B}$. It is straight forward to see that $\|v\|_\infty \leq 2\|f\|_\infty / (d\lambda(B))$. Since moduli of continuity are preserved under restriction of the domain, applying Lemma 6.2.5 leads to a linear modulus of continuity of w_0. This is basically the proof of the following:

Proposition 6.2.7. *Whenever $f \in \mathrm{L}^\infty$ the function*

$$n \mapsto d \cdot \mu(n+2) + 3n + C$$

with

$$C := \left\lceil \mathrm{lb}\left(\max\left\{\frac{2\|f\|_\infty^2}{d^2\lambda(B)^2}, 1\right\} \max\left\{\frac{2\,\|f\|_\infty}{\lambda(B)}, 1\right\}\right)\right\rceil + \lceil \mathrm{lb}(\max\{\|f\|_\infty, 1)\rceil + d + 6$$

is a modulus of continuity of $\mathrm{sol}_P^h(f)$*.*

6.2.3 Regularity of solutions

From analysis it is known that a lot more can be said about the regularity of the solution in the case where f is Hölder continuous. The solution $\mathrm{sol}_P^h(f)$ is Hölder continuous with the same Hölder constant as f (unless the Hölder constant is 1 in this case the solution need not be Lipschitz).

Remark 6.2.8. Note that from this it follows that the results from the previous section, while sufficient for our purposes, are not optimal: By Example 2.1.2 the Hölder constant corresponds to the slope of the modulus of continuity, thus the multiplication of the modulus by the dimension cannot be necessary.

Furthermore it is known that if f is Hölder continuous the Newtonian potential (and therefore also $\mathrm{sol}_P^h(f)$) is twice continuously differentiable. The Hölder continuous functions correspond to the functions with linear modulus of continuity. This section proves that this condition can be relaxed to polynomial moduli of continuity. In particular the solution for polynomial-time computable right hand side are classical solutions.

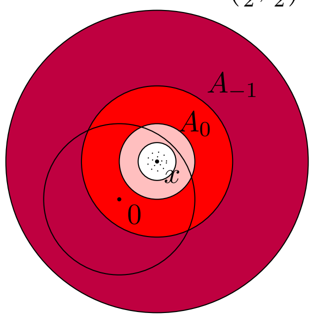
We start with a simple Lemma:

Lemma 6.2.9. *The series*

$$\sum_{m=1}^{\infty} P(m) \cdot 2^{-m}$$

converges absolutely for any polynomial P.

Figure 6.7: The sets A_n for $d = 2$ and $x = \left(\frac{1}{2}, \frac{1}{2}\right)$.



Proof. Use l'Hôspital's rule to see that the function $h : [0;\infty) \to \mathbb{R},\ x \mapsto P(x)2^{-\frac{x}{2}}$ is bounded. Thus

$$\begin{aligned}\sum\nolimits_{m=n}^{\infty} |P(m)\cdot 2^{-m}| &= \sum\nolimits_{m=n}^{\infty} |P(m)\cdot 2^{-\frac{m}{2}}| \cdot 2^{-\frac{m}{2}} \\ &\leq \|h\|_\infty \cdot \sum\nolimits_{m=0}^{\infty} (1/\sqrt{2})^m < \infty.\end{aligned}$$

This proves the assertion. ∎

Theorem 6.2.10. *If f has a polynomial modulus of continuity, the weak solution $\mathrm{sol}_P^h(f)$ is twice continuously differentiable in B. If L is a Lipschitz constant of f, the second derivatives of the solution are bounded by $2\mathrm{lb}(L) + \|f\|_\infty / d$.*

PROOF (PROOF OF THE DIFFERENTIABILITY.). The function $w := \mathrm{sol}^h_P(f) - v$ (see (N) on page 114) is the solution of a Dirichlet Problem for Laplace's Equation and as such analytic on B.

Thus, it suffices to show that the Newtonian potential

$$v : B \to \mathbb{R}, \quad x \mapsto \int_B \Gamma(x-y) \cdot f(y) \mathrm{d}y$$

is twice continuously differentiable. It is well known that this function is once continuously differentiable [GT01, Lemma 4.1].

By an observation by Morera [Mor87], the Newtonian potential is twice differentiable, whenever the integral

$$\int_B \frac{|f(x)-f(y)|}{\|x-y\|^d} \mathrm{d}y$$

converges for any $x \in B$. In this case the second partial derivatives are given by

$$\frac{\partial^2 \mathrm{sol}^h_P(f)}{\partial x_i \partial x_j} = \int_B (f(y)-f(x)) \frac{\partial^2 G}{\partial x_i \partial x_j}(x,y)\mathrm{d}y - \frac{1}{d} f(x)\delta_{i,j} \tag{6.6}$$

(where $\delta_{i,i} = 1$ and $\delta_{i,j} = 0$ for $i \neq j$). To see that the integral is finite if there is a polynomial modulus of continuity divide the unit ball into spheres of finite thickness around x (compare Figure 6.7)

$$B = \bigcup_{n=-1}^{\infty} (B \cap \underbrace{(B_{2^{-\mu(n)}}(x) \setminus B_{2^{-\mu(n+1)}}(x))}_{=:A_n}),$$

(with the convention $\mu(-1) = -1$) and estimate it by an infinite sum:

$$\begin{aligned}\int_B \frac{|f(x)-f(y)|}{\|x-y\|^d} \mathrm{d}y &\leq \sum_{n=0}^{\infty} 2^{-n} \cdot \int_{A_n} \frac{1}{\|x-y\|^d} \mathrm{d}y \\ &= \ln(2)d \cdot \lambda(B) \cdot \sum\nolimits_{n=0}^{\infty} \big(\mu(n+1) - \mu(n)\big) \cdot 2^{-n}.\end{aligned}$$

This sum is finite by Lemma 6.2.9. If L is an Lipschitz constant of f, then $n \mapsto n + \lceil \mathrm{lb}(L) \rceil$ is a modulus of continuity of f and the above infinite sequence reduces to $2\mathrm{lb}(L)$. This inequality can be used to estimate the second derivatives by the concrete formula from Equation (6.6). ∎

6.3 Poisson's Equation and the integration operator

The general Dirichlet Problem for Poisson's Equation on the unit ball B

$$\Delta u = f \text{ in } B, \quad u|_{\partial B} = g$$

has a unique solution for each $f \in \mathrm{L}^\infty\big(\overline{B}\big)$ and $g \in \mathcal{C}(\partial B)$. For our purposes it is not important to understand what a weak solution is but only to acknowledge that there is an operator

$$\mathrm{sol}_P : \mathcal{C}(\overline{B}) \times \mathcal{C}(\partial B) \to \mathcal{C}(\overline{B})$$

that should be considered the solution operator of the general Dirichlet Problem for Poisson's Equation.

Building on the last section, this section provides a proof of the following:

Theorem 6.3.1. *The solution operator* sol_P *of the general Dirichlet Problem for Poisson's Equation is polynomial-time Weihrauch equivalent to the integration operator from* (INT) *on page 102.*

6.3.1 Integrating Poisson's Equation

This section proves that the solution operator of the Dirichlet Problem for Poisson's Equation is polynomial-time Weihrauch reducible to the integration operator INT:

Theorem 6.3.2. $\mathrm{sol}_L \leq_P \mathrm{INT}$, $\mathrm{sol}_P^h \leq_P \mathrm{INT}$ *and* $\mathrm{sol}_P \leq_P \mathrm{INT}$.

The first two statements are separately proven in Proposition 6.3.4 and Proposition 6.3.8 and combined to a proof of the above on page 122. For the first two cases the approach is very similar: To obtain the moduli use Lemma 6.2.5 resp. Proposition 6.2.7. Then truncate the integrands of the concrete solution formulas from Section 6.2.1, transform to radial coordinates and finaly apply Theorem 6.1.17 to find approximations to the values on dyadic element by one application of the integration operator. The algorithms producing these sequences can be used to compute the needed approximations by noting that the distance of a dyadic point of the unit ball from the boundary does not shrink too fast as the size of the encoding increases. Finally the results for sol_L and sol_P^h can be combined to prove $\mathrm{sol}_P \leq_P \mathrm{INT}$.

Laplace's Equation

Recall that the solution $\mathrm{sol}_L(g)$ of the Dirichlet Problem for Laplace's Equation with boundary condition $g : \partial B \to \mathbb{R}$ can be expressed by the explicit formula from Theorem 6.2.1:

$$\mathrm{sol}_L(g)(x) = \int_{\partial B} K(x,y) g(y) \mathrm{d}\sigma(y), \tag{PI}$$

where σ is the surface measure on the unit sphere and

$$K(x,y) = \frac{1 - \|x\|^2}{d \cdot \lambda(B) \|x-y\|^d}. \tag{K}$$

Consider the sequence of functions

$$K_m : [-1;1]^d \times [-1;1]^d \to \mathbb{R}, \quad (x,y) \mapsto \begin{cases} \min\left\{\frac{1-\|x\|^2}{\|x-y\|^d}, \frac{1-\|x\|^2}{(m+1)^d}\right\} & \text{if } x \neq y \\ \frac{1-\|x\|^2}{(m+1)^d} & \text{if } x = y \end{cases}$$

(cf. Figure 6.8) and set

$$w_m(x) := \int_{\partial B} K_m(x,y) \cdot g(y) \mathrm{d}\sigma(y).$$

Since the truncation only effects the values on x close to the boundary, the functions w_m are identical to $\mathrm{sol}_L(g)$ on most of the unit ball:

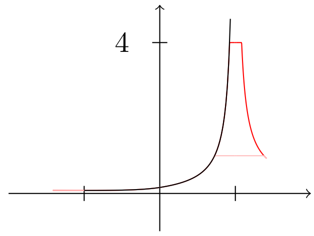


Figure 6.8: Cross-sections of Poisson kernel and truncations for $d = 2$.

Lemma 6.3.3. *If* $x \in B_{1-\frac{1}{m+1}}(0)$ *then*

$$w_m(x) = \mathrm{sol}_L(g)(x).$$

Proof. The requirement on x implies that for any $y \in \partial B$ the inequality $\|x-y\|^d \geq (1-\|x\|)^d \geq (m+1)^{-d}$ holds. Thus, for such x it follows that $K_m(x,y) = K(x,y)$ and also

$$w_m(x) = \int_{\partial B} K_m(x,y) g(y) \mathrm{d}\sigma(y) = \int_{\partial B} K(x,y) \cdot g(y) \mathrm{d}\sigma(y) = \mathrm{sol}_L(g)(x).$$

The following proposition is the first third of Theorem 6.3.2.

Proposition 6.3.4. *The solution operator* $\mathrm{sol}_L : \mathcal{C}(\partial B) \to \mathcal{C}(\overline{B})$ *of the Dirichlet Problem for Laplace's Equation is polynomial-time Weihrauch reducible to* INT.

Proof. By Theorem 6.1.17 the integration operator INT is polynomial-time Weihrauch equivalent to $(\mathrm{INT}_d^{d-1})^{\mathbb{N}}$, that is the operator that takes a sequence of $2d-1$ variate functions and returns the sequence of d variate functions where the other $d-1$ variables have been integrated over the unit interval. So reduce to this operator instead.

First describe the pre-processor of the Weihrauch reduction: It first multiplies the input function with the truncated Poisson kernel and then transforms into spherical coordinates. Thus turning the needed integration over ∂B into an integration over a $d-1$ dimensional square.

Because the functions

$$h_m : [-1;1]^d \times [-1;1]^d \to \mathbb{R}, \quad (x,y) \mapsto \begin{cases} \min\left\{\frac{1}{\|x-y\|^d}, \frac{1}{(m+1)^d}\right\} & \text{if } x \neq y \\ \frac{1}{(m+1)^d} & \text{if } x = y \end{cases}$$

form a polynomial-time computable sequence, and $K_m = (1-\|x\|^2)h_m$ also $(K_m)_{m\in\mathbb{N}}$ forms a polynomial-time computable sequence. The multiplication with the truncated Poisson kernels can therefore be done in polynomial time by Lemma 6.1.13.

Recall that the transformation to spherical coordinates in d dimensions is given by

$$\Phi(r,\theta_1,\ldots,\theta_{d-1})_i = \begin{cases} r \cdot \cos(\theta_i) \cdot \prod_{j=1}^{i-1} \sin(\theta_j) & \text{if } i < d-1 \\ r \cdot \sin(\theta_{d-1}) \prod_{j=1}^{i-2} \sin(\theta_j) & \text{if } i = d-1 \\ r \cdot \cos(\theta_{d-1}) \prod_{j=1}^{i-2} \sin(\theta_j) & \text{if } i = d \end{cases}$$

(cf. for instance [Blu60]) with Jacobian determinant

$$|D\Phi|(r,\theta_1,\dots,\theta_{d-1}) = r^{d-1}\cdot\prod_{j=1}^{d-2}\sin^{d-1-j}(\theta_j).$$

All these functions are polynomial-time computable and for

$$\tilde{g} : \theta := (\theta_1,\dots,\theta_{d-1}) \mapsto g\big(\Phi(1,\theta_1,\dots,\theta_{d-1})\big)$$

polynomial-time computability of the operation $g\mapsto\tilde{g}$ follows from the polynomial-time computability of composition. Thus, again by Lemma 6.1.13, the pre-processor that maps the function g to the sequence of functions

$$I_m : (x,\theta)\mapsto K_m\big(x,\Phi(1,\theta)\big)\cdot\tilde{g}(\theta)\cdot|D\Phi|(1,\theta)$$

is polynomial-time computable.

Next describe the post-processor of the Weihrauch reduction: Note, that the post-processor is also given direct access to the name of the input function g. Therefore it can compute values of a modulus of continuity of u according to Lemma 6.2.5. The post-processor is given a name ϕ of a sequence of functions and input $\langle [\![q]\!], 1^n\rangle$. Let k be the maximum of the denominators of q and return an encoding of $\phi(\langle\langle [\![q]\!], \mathtt{1}^n\rangle, 2k^2\rangle)$ padded to the length of the modulus of continuity.

To see that this is indeed a Weihrauch reduction, note by the substitution law

$$w_m(x) = \int_{\partial B} K_m(x,y)\cdot g(y)\mathrm{d}\sigma(y) = \int_{[0;2\pi]\times[0;\pi]^{d-2}} I_m(x,\theta)\mathrm{d}\theta.$$

Thus, applying $(\mathrm{INT}_d^{d-1})^{\mathbb{N}}$ to the value of the pre-processor returns the sequence $(w_m)_{m\in\mathbb{N}}$.

It remains to check that the post-processor returns valid approximations to the solution. By Lemma 6.3.3 w_m coincides with $\mathrm{sol}_L(g)$ for all $x\in B_{1-1/(m+1)}(0)$. Thus, it remains to check that each dyadic $q\in B$ with denominators at most k has distance at least $1/(2k^2)$ to the boundary. To see this note that $\|q\|^2\in[0;1)$ is dyadic and has denominator at most k^2. This together with

$$\|q\| = \sqrt{1+\|q\|^2-1} \leq 1-\frac{1-\|q\|^2}{2}\in[0;1)$$

implies that q is at least $1/(2k^2)$ away from the boundary. ∎

Poisson's Equation

We turn to the second part of Theorem 6.3.2, that is to the solution operator sol_P^h for the homogeneous Dirichlet Problem for Poisson's Equation.

Recall the explicit solution formula from Theorem 6.2.2:

$$\mathrm{sol}_P^h(x) = \int_B G(y,x)\cdot f(y)\mathrm{d}y$$

where G is the Green's function defined in (G).

To avoid the singularities of the Green's function, truncate the fundamental solution. Set

$$\tilde{\Gamma}_m : [0; d+2] \to \mathbb{R}, r \mapsto \begin{cases} \tilde{\Gamma}(r) & \text{if } r \geq \frac{1}{m+1}, \\ \tilde{\Gamma}(\frac{1}{m+1}) & \text{otherwise.} \end{cases}$$

(cf. Figure 6.9a). And note:

Lemma 6.3.5. *The function sequence* $\big(\tilde{\Gamma}_m\big)_{m\in\mathbb{N}}$ *is polynomial-time computable.*

Proof. It is easy to see that $\tilde{\Gamma}_m$ are Lipschitz continuous with Lipschitz constants

$$L_m = 2^{(d-1)m - \mathrm{lb}(d\cdot\lambda(B))}.$$

Since the dimension d is fixed, we can choose an integer upper bound C of $-\mathrm{lb}\big(d\cdot\lambda(B)\big)$ and the function $\mu(n) := (d-1)n + C$ will be a modulus of continuity for $(\tilde{\Gamma}_m)_{m\in\mathbb{N}}$.

It is straightforward to give algorithms computing the functions on dyadic arguments (for the case $d = 2$ see Example 2.1.5). ■

Define a sequence $G_m : [-1,1]^d \times [-1,1]^d \to \mathbb{R}$ by

$$G_m(x,y) := \begin{cases} \tilde{\Gamma}_m(\|x-y\|) - \tilde{\Gamma}_m\left(\|y\| \left\| x - \frac{y}{\|y\|^2} \right\|\right) & \text{if } y \neq 0 \\ \tilde{\Gamma}_m(\|x\|) - \tilde{\Gamma}_m(1) & \text{if } y = 0 \end{cases}$$

(see Figure 6.9b). Note that

$$\|x-y\| \leq \|x\| + \|y\| \leq 2\sqrt{d} \leq d+2$$

and

$$\left\| \|y\| \left(x - \frac{y}{\|y\|^2} \right) \right\| \leq \|y\|\|x\| + 1 \leq d+2$$

and therefore G_m is well defined from the function sequence $\big(\tilde{\Gamma}_m\big)_{m\in\mathbb{N}}$ above.

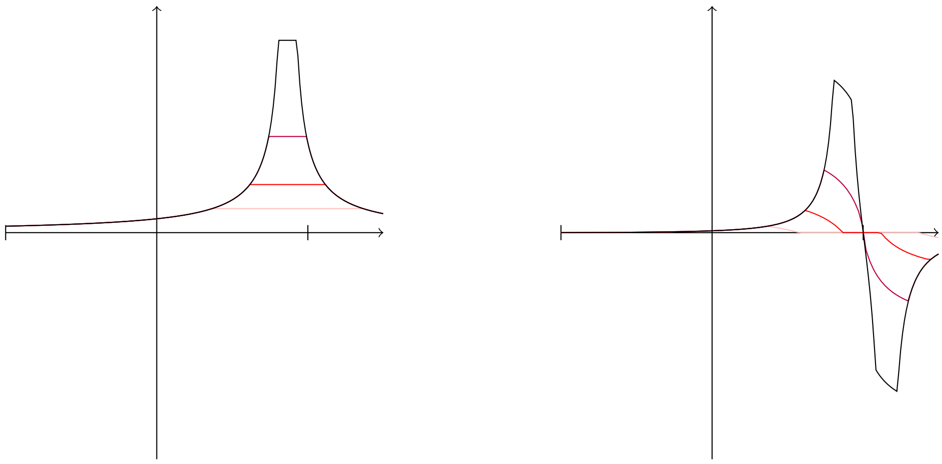
Lemma 6.3.6. *The sequence* $(G_m)_{m\in\mathbb{N}}$ *is polynomial-time computable.*

Proof. It is easy to see that the functions

$$(x,y) \mapsto \|x-y\| \quad \text{and} \quad (x,y) \mapsto \begin{cases} \left\| \left(\|y\|x - \frac{y}{\|y\|} \right) \right\| & \text{if } y \neq 0 \\ 1 & \text{if } y = 0 \end{cases}$$

are polynomial-time computable. For instance for the second function this can be verified as follows: Since the function is Lipschitz continuous with Lipschitz constant 2, it has a linear modulus of continuity. Since all involved (component) functions are computable on $B \setminus B_{2^{-N}}(0)$ in time polynomial in the output precision and N we can proceed as follows: Given some argument q and a precision requirement n check whether $\|q\|^2 \leq 2^{-2n+2}$. If it is, then compute approximations with the desired precision. If it is not, then return 1. Since we have the modulus of continuity, one can check that this leads to valid approximations in any case.

The modulus of continuity and the algorithm to compute the functions on dyadic arguments can now be easily obtained from those of these functions and $\tilde{\Gamma}_m$. ■



(a) Cross-sections of truncations of the fundamental solution.

(b) Cross-sections of truncations of the Green's function.

Figure 6.9: Cross-sections for $d = 3$ and $y = (1, 1, 1)/2$ plotted over x.

Consider the sequence formed by the functions

$$w_m : B \to \mathbb{R}, \quad x \mapsto \int_B G_m(y, x) \cdot f(y)\mathrm{d}y.$$

As should be expected, these are approximations to the solution $\mathrm{sol}_P^h(f)$:

Lemma 6.3.7. *Whenever* $x \in B_{1-\frac{1}{m+1}}(0)$, *then*

$$|w_m(x) - \mathrm{sol}_P^h(x)| \leq \frac{\|f\|_\infty}{m+1}.$$

Proof. Inserting the definitions of w_m and G_m leads to

$$\begin{aligned}
|w_m(x) - \mathrm{sol}_P^h(x)| &= \left| \int_B (G_m(y, x) - G(y, x)) \cdot f(y)\mathrm{d}y \right| \\
&\leq \left| \int_B \left(\tilde{\Gamma}_m(\|x - y\|) - \tilde{\Gamma}(\|x - y\|) \right) \cdot f(y)\mathrm{d}y \right| \\
&+ \left| \int_B (\tilde{\Gamma}_m - \tilde{\Gamma}) \left(\|y\| \left\| x - \frac{y}{\|y\|^2} \right\| \right) \cdot f(y)\mathrm{d}y \right|.
\end{aligned}$$

Note that $\tilde{\Gamma}$ and $\tilde{\Gamma}_m$ do only differ on $[0, \frac{1}{m+1})$. By assumption $\|x\| \leq 1 - \frac{1}{m+1}$. This implies

$$\|y\| \cdot \left\| x - \frac{y}{\|y\|^2} \right\| \geq |\underbrace{\|y\| \cdot \|x\|}_{\leq 1} - 1| \geq \frac{1}{m+1},$$

thus the second integral is equal to zero. The first integral on the other hand can be estimated by transforming to spherical coordinates around x:

$$|w_m(x) - \mathrm{sol}_P^h(x)| \leq \|f\|_\infty \cdot d \cdot \lambda(B) \cdot \int_0^{\frac{1}{m+1}} \tilde{\Gamma}(r) \cdot r^{d-1} \mathrm{d}r \tag{6.7}$$

The integration of $r^{d-1}\tilde{\Gamma}$ can be carried out explicitly by distinction of the cases $d = 2$ and $d > 2$ and leads to:

$$\int_0^{\frac{1}{m+1}} r^{d-1} \cdot \tilde{\Gamma}(r)\mathrm{d}r = \begin{cases} \frac{\mathrm{lb}(m+1)+1}{d \cdot \lambda(B)(m+1)^2} & \text{if } d = 2 \\ \frac{1}{2d(d-2)\lambda(B)(m+1)^2} & \text{if } d > 2 \end{cases}$$

And since $\frac{\mathrm{lb}(m+1)+1}{m+1} \leq 1$, in both cases

$$\int_0^{\frac{1}{m+1}} r^{d-1} \cdot \tilde{\Gamma}(r)\mathrm{d}r \leq \frac{1}{d \cdot \lambda(B)(m+1)}.$$

Inserting this into (6.7) results in the desired inequality. ∎

Proposition 6.3.8. *The solution operator* $\mathrm{sol}_P^h : \mathcal{C}(\overline{B}) \to \mathcal{C}(\overline{B})$ *of the homogeneous Dirichlet Problem for Poisson's Equation is polynomial-time Weihrauch reducible to* INT.

Proof. The proof is very similar to the proof of Theorem 6.3.2 and thus kept brief. Theorem 6.1.17 allows to reduce to $(\mathrm{INT}_d^d)^{\mathbb{N}}$ instead of INT. The preprocessor multiplies with the sequence G_m and transforms to spherical coordinates. The former operation is polynomial time due to Lemma 6.3.6, a proof that the latter operator is polynomial-time computable can be found in the proof of Theorem 6.3.2.

As in that proof, applying the extended integration operator leads to a name of the sequence $(w_n)_{n\in\mathbb{N}}$. The post-processor can be chosen similar to the post-processor in the proof of Theorem 6.3.2, where for the proof of correctness Lemma 6.3.3 is replaced by Lemma 6.3.7. ∎

This enables us to prove the main result of this section:

Proof (of Theorem 6.3.2). The first two claims $\mathrm{sol}_L \leq_P \mathrm{INT}$ and $\mathrm{sol}_P^h \leq_P \mathrm{INT}$ were proven in Proposition 6.3.4 resp. Proposition 6.3.8. It remains to combine them to a proof of $\mathrm{sol}_P \leq_P \mathrm{INT}$.

Note that for by linearity of the Laplace operator

$$\Delta(\mathrm{sol}_P^h(f) + \mathrm{sol}_L(g)) = \Delta\,\mathrm{sol}_P^h(f) + \mathrm{sol}_L(g) = f$$

and

$$(\mathrm{sol}_P^h(f) + \mathrm{sol}_L(g))|_{\partial B} = \mathrm{sol}_P^h(f)|_{\partial B} + \mathrm{sol}_L(g)|_{\partial B} = g.$$

Thus, $\mathrm{sol}_P(f,g) = \mathrm{sol}_P^h(f) + \mathrm{sol}_L(g)$. Since the addition operator is polynomial-time computable and recalling that $\mathrm{INT} \equiv_P \mathrm{INT} \times \mathrm{INT}$ by Corollary 6.1.18 obtain

$$\mathrm{sol}_P \leq_P \mathrm{sol}_P^h \times \mathrm{sol}_L \leq_P \mathrm{INT} \times \mathrm{INT} \leq_P \mathrm{INT}.$$

This proves the assertion. ∎

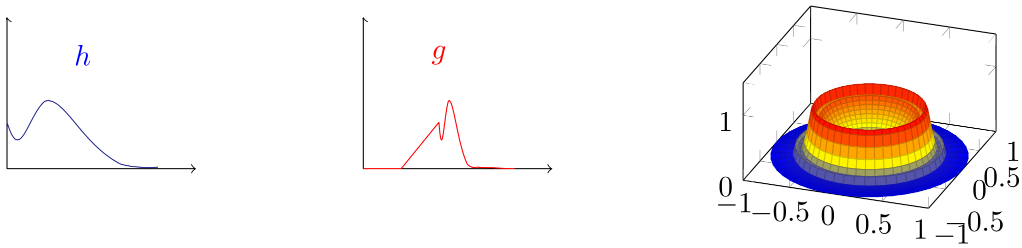


Figure 6.10: The functions g and f for a sample function h in the case $d = 2$.

6.3.2 Hardness of Poisson's Equation

The last section proved that solving the Dirichlet Problem for Poisson's Equation is at most as hard as integration. This section provides the other direction and shows that it is exactly as hard, this is the statement of Theorem 6.3.1. We actually prove the following stronger statement:

Proposition 6.3.9. $\mathrm{INT} \leq_P \mathrm{sol}_P^h$.

Proof. By Theorem 6.1.17 it suffices to reduce the restriction of INT to the set $L_{1,0}$ of Lipschitz 1 functions that vanish in zero. Reduce the restriction of INT to the bigger set of Lipschitz 1 functions bounded by one instead.

Thus, let h be a Lipschitz 1 function bounded by 1. Define the pre-processor of the Weihrauch reduction as follows: Set

$$g(x) := \begin{cases} 0 & \text{if } x \leq \frac{1}{4}, \\ 4h(0) \cdot (x - \frac{1}{4}) & \text{if } \frac{1}{4} < x \leq \frac{1}{2}, \\ h\big(4(x - \frac{1}{2})\big) & \text{if } \frac{1}{2} < x \leq \frac{3}{4}, \\ 4h(1) \cdot (1 - x) & \text{else} \end{cases}$$

And let the pre-processor be the operator mapping h to the function

$$f : B \to \mathbb{R}, \quad y \mapsto \begin{cases} \frac{g(\|y\|)}{\|y\|^{d-1}} & \text{if } y \neq 0, \\ 0 & \text{if } y = 0 \end{cases},$$

(cf. Figure 6.10). The function f is radially symmetric. Since the corresponding solution $\mathrm{sol}_P^h(f)$ is unique and the Laplace operator is also invariant under rotations, the solution is also radially symmetric. In radial coordinates Poisson's Equation takes the form

$$\frac{\partial \left(r^{d-1} \frac{\partial\, \mathrm{sol}_P^h(f)}{\partial r} \right)}{\partial r} = g. \tag{6.8}$$

With $\mathrm{sol}_P^h(f)$ also its radial derivative is radially symmetric. Thus, the function

$$u(\|y\|) := \frac{\partial\, \mathrm{sol}_P^h(f)}{\partial r}(y)$$

is well defined. From h being Lipschitz 1 and bounded by 1 it follows that f is also bounded by 1 and Lipschitz. Thus the regularity result from Theorem 6.2.10 provides a concrete bound on the second derivatives of f. It is a well known result that the derivative of a function can be computed in polynomial time from the values of the function if a bound on the second derivative is known. It follows that u is computable in polynomial time from $\mathrm{sol}_P^h(f)$ and h.

Integrating both sides of Equation (6.8) from 0 to r results in

$$\int_0^r g(t)\mathrm{d}t = r^{d-1}u(r).$$

And therefore, due to the definition of g,

$$\int_0^x h(t)\mathrm{d}t = \int_0^{\frac{1}{2}+\frac{x}{4}} g(t)\mathrm{d}t - \frac{h(0)}{2} = \left(\frac{1}{2}+\frac{x}{4}\right)^{d-1} u\left(\frac{1}{2}+\frac{x}{4}\right) - \frac{h(0)}{2}.$$

With u also the right hand side can be computed from $\mathrm{sol}_P^h(f)$ and h in polynomial time and can be used as post-processor of the Weihrauch reduction. ■

Proof (of Theorem 6.3.1). By Theorem 6.3.2 $\mathrm{sol}_P \leq_P \mathrm{INT}$. The other direction follows from Proposition 6.3.9 together with the obvious fact that $\mathrm{sol}_P^h \leq_P \mathrm{sol}_P$. ■

7 Conclusion

Before we list things that can and should still be done, recall that a list of the achievements of the thesis was given on page 7 of the introduction.

A collection of starting points for further research projects and loose ends can be found within the remarks throughout the thesis:

- The first remark on page 27 concerns the singular representation. It says that the singularity modulus looks like it is more appropriate for locally integrable functions than for integrable functions. It is, however, not really appropriate for locally integrable functions as these can have singularities that get worse and worse when leaving any bounded set. This could, at least for the one dimensional case easily be fixed by introducing another parameter. locally integrable functions turn up in the theory of distributions and are for this reason important for partial differential equations.

- On page 33 the second remark says that it is not clear what the final topology of the singular representation is. Now, if the singular representation is indeed the restriction of a representation of a space of locally integrable functions to functions with a common bounded domain, then maybe one should take a look at what topologies people from analysis use on the locally integrable functions.

- The next remark on page 80 says that deep results from coding theory may allow to improve the results. Actually results from coding theory where cited in earlier versions of this thesis, however, due to the bad accessibility and the relative minor payoff they were removed again. These connections could prove helpful in other situations, for instance for proving similar results for Sobolev spaces.

- On page 89 the remark hints that the running time bound on the norm of L^p can be used to produce bounds on the Fréchet-Kolmogorov sets for more complicated domains. Nearly all sources known to the author discuss the Fréchet-Kolmogorov Theorem and its quantitative versions at least for hyper cubes. It is worth further investigations if similar techniques like in the one dimensional case can be applied to eliminate the iteration of the size parameter.

- The next remark on page 90 gives a vague intuition for why the design choices made in the definition of representations of the Sobolev spaces might be the right ones. If these choices are indeed justified it should be possible to make the content of this remark rigorous. I.e. there should be an indexing of the compact subsets of Sobolev spaces that uses the L^p-modulus of the highest derivative as parameter and leads to sizes exponential in the parameter.

- The remark on page 94 says that there is a hidden exponential dependence on a parameter that can be removed by adding some discrete information, namely bounds on the norms of the lower derivatives. To see if this is reasonable to do, some applications or implementations are necessary.

- The last remark on page 115 addresses one of many small opportunities for improvements: The modulus extracted is not optimal and it may be possible to improve it in a straightforward way. Another more obvious gap is left by the quantitative version of the Fréchet-Kolmogorov Theorem 4.3.2: One could try to minimize the shifts done. In particular the $n+2$ in the upper bound can maybe improved to $n+1$ by using that the proof of Lemma 5.1.7 provides slightly better approximation properties. For the great scheme of the thesis, however, these improvements seem of little importance.

Of course there are several further points that are to general to be mentioned in a remark:

- To discuss partial differential equations it is indispensable to extend the approach from Section 5.2 to more dimensions. One of the goals when starting out with this thesis was to find a rigorous framework for tools like the finite elements method. The first step has been done but there are several more to do.

- However, already one dimensional Sobolev spaces have some applications and some of these have been discussed in the framework of computable analysis. In particular the results from [BY06] seem like an ideal starting point.

- The results of Section 6.3 are restricted to the unit ball. It would be desirable to be able to prove similar results for more general domains. In particular the reduction to integration part. For this it is necessary to get hold of the Green's function. Some considerations in this direction have been done by other authors.

- Polynomial-time Weihrauch reductions have been discussed surprisingly little. They seem like a promising tool for both classifying problems with respect to their difficulty as well as improving the acceptance of real complexity theory: While it might be impossible to construct an algorithm that integrates a function in polynomial time, it is possible to apply the integration algorithms that are used in numerics to other problems in an efficient way. As long as the Weihrauch reduction preserves the set of functions the algorithm works fast on, this should lead to a fast algorithm. For instance for the results of Chapter 6 the author would guess that the reduction $(\mathrm{INT}_d^{d'})^{\mathbb{N}} \leq_P \mathrm{INT}$ does not preserve these functions, while the reductions of the solution operators $\mathrm{sol}_L \leq_P (\mathrm{INT}_d^{d'})^{\mathbb{N}}$ and $\mathrm{sol}_P \leq_P (\mathrm{INT}_d^{d'})^{\mathbb{N}}$ probably do.

Bibliography

[ACN07] Klaus Aehlig, Stephen Cook, and Phuong Nguyen. "Relativizing small complexity classes and their theories". In: *Computer science logic*. Vol. 4646. Lecture Notes in Comput. Sci. Springer, Berlin, 2007, pp. 374–388.

[Bee85] Michael J. Beeson. *Foundations of constructive mathematics*. Vol. 6. Ergebnisse der Mathematik und ihrer Grenzgebiete (3) [Results in Mathematics and Related Areas (3)]. Metamathematical studies. Springer-Verlag, Berlin, 1985, pp. xxiii+466.

[Bis67] Errett Bishop. *Foundations of constructive analysis*. McGraw-Hill Book Co., New York-Toronto, Ont.-London, 1967, pp. xiii+370.

[BB85] Errett Bishop and Douglas Bridges. *Constructive analysis*. Vol. 279. Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]. Springer-Verlag, Berlin, 1985, pp. xii+477.

[Blu60] L. E. Blumenson. "Classroom Notes: A Derivation of n-Dimensional Spherical Coordinates". In: *Amer. Math. Monthly* 67.1 (1960), pp. 63–66.

[BBP12] Vasco Brattka, Matthew de Brecht, and Arno Pauly. "Closed choice and a uniform low basis theorem". In: *Ann. Pure Appl. Logic* 163.8 (2012), pp. 986–1008.

[BG11a] Vasco Brattka and Guido Gherardi. "Effective choice and boundedness principles in computable analysis". In: *Bull. Symbolic Logic* 17.1 (2011), pp. 73–117.

[BG11b] Vasco Brattka and Guido Gherardi. "Weihrauch degrees, omniscience principles and weak computability". In: *J. Symbolic Logic* 76.1 (2011), pp. 143–176.

[BGH15] Vasco Brattka, Guido Gherardi, and Rupert Hölzl. "Probabilistic computability and choice". In: *Inform. and Comput.* 242 (2015), pp. 249–286.

[BH02] Vasco Brattka and Peter Hertling. "Topological properties of real number representations". In: *Theoret. Comput. Sci.* 284.2 (2002). Computability and complexity in analysis (Castle Dagstuhl, 1999), pp. 241–257.

[BY06] Vasco Brattka and Atsushi Yoshikawa. "Towards computability of elliptic boundary value problems in variational formulation". In: *J. Complexity* 22.6 (2006), pp. 858–880.

[Bre11] Haim Brezis. *Functional analysis, Sobolev spaces and partial differential equations*. Universitext. Springer, New York, 2011, pp. xiv+599.

[BS86] Douglas K. Brown and Stephen G. Simpson. "Which set existence axioms are needed to prove the separable Hahn-Banach theorem?" In: *Ann. Pure Appl. Logic* 31.2-3 (1986). Special issue: second Southeast Asian logic conference (Bangkok, 1984), pp. 123–144.

[Bus88] Jonathan F. Buss. "Relativized alternation and space-bounded computation". In: *J. Comput. System Sci.* 36.3 (1988). Structure in complexity theory Conference (Berkeley, CA, 1986), pp. 351–378.

[BK02] Samuel R. Buss and Bruce M. Kapron. "Resource-bounded continuity and sequentiality for type-two functionals". In: *ACM Trans. Comput. Log.* 3.3 (2002). Special issue on logic in computer science (Santa Barbara, CA, 2000), pp. 402–417.

[Fri84] Harvey Friedman. "The computational complexity of maximization and integration". In: *Adv. in Math.* 53.1 (1984), pp. 80–98.

[GT01] David Gilbarg and Neil S. Trudinger. *Elliptic partial differential equations of second order*. Classics in Mathematics. Reprint of the 1998 edition. Berlin: Springer-Verlag, 2001, pp. xiv+517.

[Grz55] A. Grzegorczyk. "Computable functionals". In: *Fund. Math.* 42 (1955), pp. 168–202.

[HP13] Kojiro Higuchi and Arno Pauly. "The degree structure of Weihrauch reducibility". In: *Log. Methods Comput. Sci.* 9.2 (2013), 2:02, 17.

[KC96] B. M. Kapron and S. A. Cook. "A new characterization of type-2 feasibility". In: *SIAM J. Comput.* 25.1 (1996), pp. 117–132.

[Kaw11] Akitoshi Kawamura. "Computational Complexity in Analysis and Geometry". PhD thesis. University of Toronto, 2011.

[KC10] Akitoshi Kawamura and Stephen Cook. "Complexity theory for operators in analysis". In: *STOC'10—Proceedings of the 2010 ACM International Symposium on Theory of Computing*. ACM, New York, 2010, pp. 495–502.

[KO14] Akitoshi Kawamura and Hiroyuki Ota. "Small complexity classes for computable analysis". In: *Mathematical foundations of computer science 2014. Part II*. Vol. 8635. Lecture Notes in Comput. Sci. Springer, Heidelberg, 2014, pp. 432–444.

[KP14] Akitoshi Kawamura and Arno Pauly. "Function spaces for second-order polynomial time". In: *Language, life, limits*. Vol. 8493. Lecture Notes in Comput. Sci. Springer, Cham, 2014, pp. 245–254.

[KSZ16a] Akitoshi Kawamura, Florian Steinberg, and Martin Ziegler. *Complexity Theory of (Functions on) Compact Metric Spaces*. accepted for LICS2016 Conference. 2016.

[KSZ16b] Akitoshi Kawamura, Florian Steinberg, and Martin Ziegler. "On the computational complexity of the Dirichlet Problem for Poisson's Equation". In: *Mathematical Structures in Computer Science* FirstView (July 2016), pp. 1–29.

[KSZ16c] Akitoshi Kawamura, Florian Steinberg, and Martin Ziegler. "Towards Computational complexity theory on Advanced Function Spaces in Analysis". In: *Pursuit of the Universal: 12th Conference on Computability in Europe, CiE 2016, Paris, France, June 27 - July 1, 2016, Proceedings*. Ed. by Arnold Beckmann, Laurent Bienvenu, and Nataša Jonoska. Cham: Springer International Publishing, 2016, pp. 142–152.

[Kaw+15] Akitoshi Kawamura et al. "Computational benefit of smoothness: Parameterized bit-complexity of numerical operators on analytic functions and Gevrey's hierarchy". In: *J. Complexity* 31.5 (2015), pp. 689–714.

[Kle52] Stephen Cole Kleene. *Introduction to metamathematics*. D. Van Nostrand Co., Inc., New York, N. Y., 1952, pp. x+550.

[Ko91] Ker-I Ko. *Complexity theory of real functions*. Progress in Theoretical Computer Science. Birkhäuser Boston, Inc., Boston, MA, 1991, pp. x+309.

[KF82] Ker-I Ko and Harvey Friedman. "Computational complexity of real functions". In: *Theoret. Comput. Sci.* 20.3 (1982), pp. 323–352.

[Koh08] U. Kohlenbach. *Applied proof theory: proof interpretations and their use in mathematics*. Springer Monographs in Mathematics. Springer-Verlag, Berlin, 2008, pp. xx+532.

[Koh96] Ulrich Kohlenbach. "Mathematically strong subsystems of analysis with low rate of growth of provably recursive functionals". In: *Arch. Math. Logic* 36.1 (1996), pp. 31–71.

[Koh05] Ulrich Kohlenbach. "Some computational aspects of metric fixed-point theory". In: *Nonlinear Anal.* 61.5 (2005), pp. 823–837.

[KLN15] Ulrich Kohlenbach, Laurentiu Luestean, and Adriana Nicolae. "Quantitative results on Fejér monotone sequences". 2015.

[KT59] A. N. Kolmogorov and V. M. Tihomirov. "ε-entropy and ε-capacity of sets in function spaces". In: *Uspehi Mat. Nauk* 14.2 (86) (1959), pp. 3–86.

[KS05] Daren Kunkle and Matthias Schröder. "Some examples of non-metrizable spaces allowing a simple type-2 complexity theory". In: *Proceedings of the 6th Workshop on Computability and Complexity in Analysis (CCA 2004)*. Vol. 120. Electron. Notes Theor. Comput. Sci. Elsevier, Amsterdam, 2005, pp. 111–123.

[LLM01] S. Labhalla, H. Lombardi, and E. Moutai. "Espaces métriques rationnellement présentés et complexité: le cas de l'espace des fonctions réelles uniformément continues sur un intervalle compact". In: *Theoret. Comput. Sci.* 250.1-2 (2001), pp. 265–332.

[Lam06] Branimir Lambov. "The basic feasible functionals in computable analysis". In: *J. Complexity* 22.6 (2006), pp. 909–917.

[Lon05] John R. Longley. "Notions of computability at higher types. I". In: *Logic Colloquium 2000*. Vol. 19. Lect. Notes Log. Assoc. Symbol. Logic, Urbana, IL, 2005, pp. 32–142.

[Lor66] G. G. Lorentz. "Metric entropy and approximation". In: *Bull. Amer. Math. Soc.* 72.6 (Nov. 1966), pp. 903–937.

[Meh76] Kurt Mehlhorn. "Polynomial and abstract subrecursive classes". In: *J. Comput. System Sci.* 12.2 (1976). Sixth Annual ACM Symposium on the Theory of Computing (Seattle, Wash., 1974), pp. 147–178.

[Mor87] G. Morera. "Sulle derivate seconde della funzione potenziale di spazio." Italian. In: *Ist. Lombardo, Rend., II. Ser.* 20 (1887), pp. 302–310.

[PS16] Arno Pauly and Florian Steinberg. "Representations of Analytic Functions and Weihrauch Degrees". In: *Computer Science – Theory and Applications: 11th International Computer Science Symposium in Russia, CSR 2016, St. Petersburg, Russia, June 9-13, 2016, Proceedings*. Ed. by S. Alexander Kulikov and J. Gerhard Woeginger. Cham: Springer International Publishing, 2016, pp. 367–381.

[PER89] Marian B. Pour-El and J. Ian Richards. *Computability in analysis and physics*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1989, pp. xii+206.

[Rud87] Walter Rudin. *Real and complex analysis*. Third. McGraw-Hill Book Co., New York, 1987, pp. xiv+416.

[Sch02] Matthias Schröder. "Admissible Representations for Continuous Computations". PhD thesis. FernUniversität Hagen, 2002.

[Sch04] Matthias Schröder. "Spaces allowing type-2 complexity theory revisited". In: *MLQ Math. Log. Q.* 50.4-5 (2004), pp. 443–459.

[SSZ16] Matthias Schröder, Florian Steinberg, and Martin Ziegler. "Average-Case Bit-Complexity Theory of Real Functions". In: *Mathematical Aspects of Computer and Information Sciences: 6th International Conference, MACIS 2015, Berlin, Germany, November 11-13, 2015, Revised Selected Papers*. Ed. by S. Ilias Kotsireas, M. Siegfried Rump, and K. Chee Yap. Cham: Springer International Publishing, 2016, pp. 505–519.

[Sud01] Madhu Sudan. "Coding theory: tutorial & survey". In: *42nd IEEE Symposium on Foundations of Computer Science (Las Vegas, NV, 2001)*. IEEE Computer Soc., Los Alamitos, CA, 2001, pp. 36–53.

[Tim94] A. F. Timan. *Theory of approximation of functions of a real variable*. Translated from the Russian by J. Berry, Translation edited and with a preface by J. Cossar, Reprint of the 1963 English translation. Dover Publications, Inc., New York, 1994, pp. viii+631.

[Tod91] Seinosuke Toda. “PP is as hard as the polynomial-time hierarchy”. In: *SIAM J. Comput.* 20.5 (1991), pp. 865–877.

[Tro92] A. S. Troelstra. “Comparing the theory of representations and constructive mathematics”. In: *Computer science logic (Berne, 1991).* Vol. 626. Lecture Notes in Comput. Sci. Springer, Berlin, 1992, pp. 382–395.

[Tro69] A. S. Troelstra. *Principles of intuitionism.* Lectures presented at the Summer Conference on Intuitionism and Proof Theory (1968) at SUNY at Buffalo, N.Y. Lecture Notes in Mathematics, Vol. 95. Springer-Verlag, Berlin-New York, 1969, pp. ii+111.

[TD88a] A. S. Troelstra and D. van Dalen. *Constructivism in mathematics. Vol. I.* Vol. 121. Studies in Logic and the Foundations of Mathematics. An introduction. North-Holland Publishing Co., Amsterdam, 1988, pp. xx+342+XIV.

[TD88b] A. S. Troelstra and D. van Dalen. *Constructivism in mathematics. Vol. II.* Vol. 123. Studies in Logic and the Foundations of Mathematics. An introduction. North-Holland Publishing Co., Amsterdam, 1988, i–xviii and 345–880 and I–LII.

[Tur36] A. M. Turing. “On Computable Numbers, with an Application to the Entscheidungsproblem”. In: *Proc. London Math. Soc.* S2-42.1 (1936), pp. 230–265.

[Tur37] A. M. Turing. “On Computable Numbers, with an Application to the Entscheidungsproblem. A Correction”. In: *Proc. London Math. Soc.* S2-43.6 (1937), p. 544.

[Wei00] Klaus Weihrauch. *Computable analysis.* Texts in Theoretical Computer Science. An EATCS Series. An introduction. Springer-Verlag, Berlin, 2000, pp. x+285.

[Wei03] Klaus Weihrauch. “Computational complexity on computable metric spaces”. In: *MLQ Math. Log. Q.* 49.1 (2003), pp. 3–21.

[WZ07] Klaus Weihrauch and Ning Zhong. “Computable analysis of the abstract Cauchy problem in a Banach space and its applications. I”. In: *MLQ Math. Log. Q.* 53.4-5 (2007), pp. 511–531.

[WZ02] Klaus Weihrauch and Ning Zhong. “Is Wave Propagation Computable or Can Wave Computers Beat the Turing Machine?” In: *Proc. London Mathematical Society* 85.2 (2002), pp. 312–332.

[Wer00] Dirk Werner. *Funktionalanalysis.* extended. Springer-Verlag, Berlin, 2000, pp. xii+501.

[WKK09] Ernst Wienholtz, Hubert Kalf, and Thomas Kriecherbauer. *Elliptische Differentialgleichungen zweiter Ordnung.* Eine Einführung mit historischen Bemerkungen. [An introduction with historical remarks]. Dordrecht: Springer, 2009, pp. xii+401.

[Wil88] Christopher B. Wilson. "A measure of relativized space which is faithful with respect to depth". In: *J. Comput. System Sci.* 36.3 (1988). Structure in complexity theory Conference (Berkeley, CA, 1986), pp. 303–312.

[Zho99] Ning Zhong. "Computability structure of the Sobolev spaces and its applications". In: *Theoret. Comput. Sci.* 219.1-2 (1999). Computability and complexity in analysis (Castle Dagstuhl, 1997), pp. 487–510.

[ZW03] Ning Zhong and Klaus Weihrauch. "Computability theory of generalized functions". In: *J. ACM* 50.4 (2003), pp. 469–505.

[ZZ99] Ning Zhong and Bing-Yu Zhang. "L^p-computability". In: *MLQ Math. Log. Q.* 45.4 (1999), pp. 449–456.

[ZB04] Martin Ziegler and Vasco Brattka. "Computability in linear algebra". In: *Theoret. Comput. Sci.* 326.1-3 (2004), pp. 187–211.

Wissenschaftlicher Werdegang

Der Autor begann seinen wissenschaftlichen Werdegang an der Wöhlerschule, wo er im Jahre 2005 sein Abitur mit dem 'DPG-Buchpreis 2005' für Physik abschloss. Direkt im Anschluss fing er an an der Technischen Universität Darmstadt Physik zu studieren. Im Laufe der ersten vier Semester erweiterte der Autor dies zunächst zu einem dualen Studium und wechselte letztendlich vollständig in die Mathematik. Seine Diplomarbeit mit dem Titel 'Faltungsoperatoren auf endlichen Quantengruppen als Beispiele vollständig positiver Operatoren' fertigte er unter Aufsicht von Prof. Kümmerer an. Nach einer halbjährigen Anstellung als wissenschaftlicher Mitarbeiter für das Projekt 'A datatype for analytic functions in iRRAM' begann der Autor betreut von Prof. Martin Ziegler zu promovieren. Diese Promotion wurde über ein DFG Stipendium finanziert das dem Autor von dem IRTG 1529 'mathematische Fluid Dynamik' verliehen wurde. Dieses Stipendium beinhaltete einen halbjährigen Aufenthalt in Japan. Kurz nach diesem Aufenthalt verließ Prof. Martin Ziegler die Technische Universität Darmstadt und trat eine Stelle in Korea an. Dies und die funktionierende Kooperation mit Akitoshi Kawamura und Holger Thies führten zu einem weiteren längeren Aufenthalt des Autors in Japan. Die Betreuung in Darmstadt wurde von Prof. Kohlenbach übernommen.