École doctorale IAEM Lorraine

Aspects Topologiques des Représentations en Analyse Calculable

(Topological Aspects of Representations in Computable Analysis)

Soutenance le 9 janvier 2023

pour l'obtention d'une

Habilitation à Diriger des Recherches de l'Université de Lorraine

(mention informatique, section 27)

par

Mathieu Hoyrup

Composition du jury

Président :	Julien Cervelle	PR, Université Paris-Est Créteil
Rapporteurs :	Olivier Bournez	PR, École Polytechnique
	Vasco Brattka	PR, Universität der Bundeswehr München
	Elvira Mayordomo	PR, Universidad de Zagaroza
Examinateurs :	Olivier Finkel	CR CNRS, IMJ-PRG
	Emmanuel Jeandel	PR, Université de Lorraine

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thesul.

Résumé

L'analyse calculable permet de formaliser le traitement algorithmique d'objets mathématiques infinis. La théorie repose sur une représentation symbolique des objets, dont le choix détermine les capacités de calcul de la machine, notamment sa difficulté à résoudre chaque problème donné. La friction entre le caractère discret du calcul et la nature continue des objets est capturée par la topologie, qui exprime l'idée d'approximation finie d'objets infinis.

Nous étudions en profondeur les multiples interactions entre calcul et topologie, cherchant à analyser l'information qui peut être extraite algorithmiquement d'une représentation. Je me penche plus particulièrement sur la comparaison entre deux représentations d'une même famille d'objets, sur les liens détaillés entre complexité algorithmique et topologique des problèmes, ainsi que sur les relations entre représentations finies et infinies.

Abstract

Computable analysis provides a formalization of algorithmic computations over infinite mathematical objects. The central notion of this theory is the symbolic representation of objects, which determines the computation power of the machine, and has a direct impact on the difficulty to solve any given problem. The friction between the discrete nature of computations and the continuous nature of mathematical objects is captured by topology, which expresses the idea of finite approximations of infinite objects.

We thoroughly study the multiple interactions between computations and topology, analysing the information that can be algorithmically extracted from a representation. In particular, we focus on the comparison between two representations of a single family of objects, on the precise relationship between algorithmic and topological complexity of problems, and on the relationship between finite and infinite representations.

Remerciements

Je remercie tout d'abord Olivier Bournez, Vasco Brattka et Elvira Mayordomo d'avoir accepté d'être rapporteur(e)s, ainsi que Julien Cervelle, Olivier Finkel et Emmanuel Jeandel pour leur participation au jury.

Merci aux membres de Mocqua et de son ancêtre Carte pour la bonne ambiance d'équipe, merci aux assistantes pour leur aide précieuse. Merci à l'ensemble du personnel qui contribue à la qualité du cadre de travail. Merci à la communauté de calculabilité pour son dynamisme et son ouverture d'esprit. Merci à mes collaborateurs et étudiants, ce fut un plaisir de travailler ensemble.

Merci à ma famille pour tout le reste. Merci à mes parents et à ma sœur pour la continuité, merci à Anne-Lise, Paul et Gaspard pour la densité, merci à Caroline pour la complétude.

Contents

Part I

Overview

Chapter 1

Introduction

My research takes place in the field of computable analysis. It is the theoretical study of computations involving infinite objects such as real numbers, compact subsets of Euclidean spaces, functions over the natural numbers or any objects arising from mathematical analysis. This active research area, although theoretical in nature, gives information about the possibility of practical computations, as it is based on the computation model of Turing machines performing discrete computations in finite time but with arbitrary precision, and guarantees on the correctness of the result. A typical negative result states that some problem is not computable, which in practical terms means that more information needs to be provided on the input to solve the problem. A typical positive result states that some problem is computable, which then raises the question of the complexity of that problem.

Most of the modern research in computable analysis uses the model of oracle or type-two Turing machines, allowing infinite objects to be represented by infinite sequences of bits or natural numbers. A representation of a class of objects then determines the computing abilities of the machine, having a direct impact on the computability of problems involving these objects. Therefore, one of the main questions in this area is to understand how the representation affects computations, and the general goal of my research is to improve our understanding of this question.

Mathematical objects usually have multiple facets and carry many different types of information. Large parts of computability theory and computable analysis are about comparing different pieces of information and how they can be computably derived from each other. Typical questions are: given an object represented in some way, is it possible to compute some piece of information about that object? Given two representations of the same objects, is it possible to computably translate one into the other?

Computations and representations of mathematical objects can be seen as structures that are added to the ordinary structures of mathematics, such as topologies or partial orders, and that interact with them in many ways. The driving force of this interaction is the friction between the discrete, finite nature of a computing device and the infinite nature of mathematical objects. The general direction underlying my research work is to exploit the relationship between computations and topology, with several goals in mind:

- The first goal is to gain a conceptual understanding of the precise relationship between computability and topology, which are two different ways of analyzing mathematical objects by the information available about them,
- The second goal is pedagogical. In computability theory and computable analysis, many problems have a similar flavour and are solved using similar techniques. One usually needs

some experience to absorb these techniques and to know how and when to apply them, by integrating a catalogue of examples before being able to reproduce them.

Identifying mathematically explicit and precise features that are common to these constructions and make them possible gives a shortcut in this learning process, and also gives an explanation of why the construction is possible; the main goal of science is not just to prove new results, but to explain why they hold.

- The third goal is pragmatic. We want to develop techniques to help carrying out difficult constructions, typically by providing a result performing that construction in an abstract setting, that can be applied in any particular situation by checking that the conditions are met. A good result identifies simple natural conditions, rather than *ad hoc* assumptions that would be tailored towards the construction; indeed, we want the result to make our lives easier, so both understanding and checking the conditions should be much easier than doing the construction by hand.

1.1 Context

Representations are implicitly used in most of the literature on computable mathematics, from Turing [Tur36] to Rice [Ric54], Grzegorczyk [Grz57] or Lacombe [Lac59] to name a few. The systematic study of representations was started by Weihrauch and Kreitz in [KW85]. The theory gives a central role to topology, by defining a representation to be *admissible* if it makes the computable functions the effective analog of continuous functions. Weihrauch's book [Wei00] settles the core of the theory of representations on countably-based topological space, which are ubiquitous in mathematics, simple to analyze algorithmically and for which representations behave very smoothly. In his PhD [Sch02a], Schröder extended the theory beyond countably-based spaces, identifying at the same time the class of topological spaces having an admissible representation, namely the T_0 quotients of countably-based spaces, and highlighting in this general setting that computability is related to sequential continuity rather than continuity. Pauly synthesized and developed in [Pau16] a different approach, showing how topological notions can be derived from the representation, building on work by Schröder [Sch02a], Escardó [Esc04], Taylor [Tay11]. Many other works have contributed to the development of the theory of representations, by Brattka, Hertling, Schröder, Ziegler, de Brecht, Kihara and many others [BH02, Sch04, Bra05, Zie12, dB13, KP14].

The *Handbook on Computability and Complexity in Analysis* [BH21] presents an overview of the state of the art and of the most recents advances in Computable Analysis.

There exist other approaches to computable analysis, which are not discussed in this document. The most famous ones are the Russian school of constructive mathematics ([Šan68] for instance), Bishop's constructive analysis [Bis67], domain theory [Eda97], the Blum-Shub-Smale model [BCSS12].

1.2 The general questions

Below I present in simple terms the general questions that drive my research work. They are usually informal, sometimes vague, but will lead to precisely formulated questions calling for a mathematical answer. They are motivated by the need to better understand concepts, objects, techniques, to relate different notions, and to provide simple and useful tools to easily derive difficult results. Sometimes the motivation is only that the question is natural or easy to formulate but has no simple answer, which is evidence that our concepts and understanding are not in line with the mathematical phenomena and that new ideas are needed.

What information is given by a representation?

Given a class of objects and a representation of them, what information about the objects is given by that representation? Let us give a few examples:

- A computable function $f : \mathbb{N} \to \mathbb{N}$ can be represented by a finite program,
- A continuous higher-order functional of type $F : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ can be represented by a list of tuples $(n_0, \dots, n_k, n)$ such that for every $f : \mathbb{N} \to \mathbb{N}$ whose values are $f(0) = n_0, \dots, f(k) = n_k$, one has $F(f) = n$,
- A polynomial $P \in \mathbb{R}[X]$ can be represented by giving an upper bound on its degree together with its coefficients, using a standard representation of real numbers.

In each case, we want to understand what information can be extracted from the representation. In order to give mathematical answers, we can reformulate this vague problem into more precise questions:

- What properties of the objects are decidable or semidecidable using these representations?
- More generally, given a fixed complexity level, what properties have that complexity? Complexity will be measured using descriptive set theory.

Topology is behind many constructions.

Practitioners of computability theory and computable analysis know that topology is at play behind many constructions. However, only few results make the role of topology explicit. Having such results is interesting from several perspectives:

- Conceptual: they build a precise bridge between topology and computability,
- Pedagogical: they give a conceptual understanding of a technique, by identifying in which situations it can be applied,
- Pragmatic: they provide a general result capturing the most technical aspects of the construction, that can be applied in various situations by simply checking that the assumptions are met.

For instance, in computability theory one often needs to resort to priority methods in order to build objects satisfying infinitely many properties at the same time. Although many frameworks have been developed to present these arguments in a unified way, there is to our knowledge no general result that identifies "simple" conditions on a family of properties, and proves the existence of an object satisfying these properties. Of course, "simplicity" is subjective, but the result should take care of most of the complexity of the construction by making it much easier to understand and check these conditions than to carry out the construction. A way to reach this goal is to look for conditions that can be expressed using already existing concepts, notably topological ones.

Some of the results presented in this manuscript, are answers to this question, but this project is still ongoing. In particular it is not clear whether the most advanced constructions can be recasted using this approach, but I would like to push this idea as far as possible.

Computability vs continuity.

It is an empirical fact that most natural functions $f : X \to Y$ occurring in mathematical analysis fall into one of the following categories:

- f is computable, (a single algorithm transforms each input x into $f(x)$),
- f is computable, but not uniformly (each x can be transformed into $f(x)$, but with a different algorithm for each x),
- f sends some computable input x to a non-computable output $f(x)$.

For instance, the following three examples fall in the respective categories: the sine function, the function sending a real number, represented by rational approximations, to its binary expansion, and the differentiation operator (restricted to C^1-functions so that the output is continuous hence can be represented).

These three cases do not cover all the possibilities, and counter-examples can be built. However, these examples are artificial, i.e. built on purpose, and to our knowledge no "natural" example exists. A possible explanation is that natural examples often come with some additional structure that rules out pathological behaviors. Therefore, we would like to identify which additional structures or assumptions make this happen.

A famous example was given by Pour-El and Richards, who showed how a vector space structure dramatically restricts the possible behaviors. They proved that for linear operators f on Banach spaces satisfying certain mild computability assumptions, either f is computable or maps a computable input to a non-computable output. Therefore, for this class of functions, already the first and third conditions cover all the cases.

We want to identify large classes of functions, beyond the linear operators of Pour-El and Richards, for which only the three cases mentioned above are possible. One of the main goals is that it can help showing that a function falls into the third category, which is often difficult to prove by hand.

1.3 Content

This manuscript is separated in Parts I and II. Part I provides an informal overview of Part II, briefly presenting some of the problems motivating my research works, as well as the answers that we obtained to these problems. Part II contains the more detailed presentation of my work, and is organized as follows:

- In Chapter 5, we study the following question: Given two representations of the same objects, do they induce the same computable objects? We focus on representations that are induced by countably-based topologies. The results involve computability theory and Baire category. Some of the results are published in [Hoy14], other results appear here for the first time.
- In Chapter 6, we propose a notion of genericity that is compatible with weak forms of computability. The purpose is to give a way to study the properties of "typical" objects, among the weakly computable ones. In particular, it identifies conditions on properties implying the existence of weakly computable objects satisfying these properties. The results involve computability, Baire category and non-Hausdorff topologies. The relevant publications are [Hoy14, Hoy17].

- A representation of objects has a direct impact on the difficulty of decision problems on these objects. A way to understand the information content of a representation is to study the complexity of decision problems, because having more information as input decreases the complexity of problems. In Chapter 7, we study the relationship between the algorithmic descriptive complexity of problems and their topological complexity. The results were partially obtained in collaboration with Antonin Callard during his L3 internship and are published in [CH20, Hoy20b].

- In Chapter 8, we investigate the problem of describing the semidecidable properties in a given represented space. When the space is a topological space with an admissible representation, the semidecidable properties are the effective analogs of the open sets. Therefore, we study the base-complexity of the space which measures the difficulty of describing the open sets. The results appear in [Hoy22].

- In Chapter 9, we study the relationship between finite and infinite representations. More precisely, we study the difference between representations by finite programs or by infinite oracles. In particular, we give a characterization of the (semi)decidable properties of primitive recursive functions, when they are represented by finite primitive recursive schemes. Part of the results were obtained with Cristóbal Rojas and are published in [HR15, Hoy16a, HR17].

1.4 My other works

This manuscript summarizes the results that I obtained on the topological aspects of representations. I have also worked on several other topics, that are not included here. Let me briefly summarize these works.

I have been working on algorithmic randomness, as a sequel to my PhD thesis. Together with Cristóbal Rojas, we have developed a notion of probabilistic computability that is compatible with the pointwise approach of algorithmic randomness and called *layerwise computability* [HR09a, HR09b, BHS17]. I have written a book chapter on this topic in [Hoy20a]. I have also worked on further connections between algorithmic randomness, dynamical systems and the computable aspects of measure theory, working with Laurent Bienvenu, Daniel Coronel, Alexander Frank, Peter Gács, Stefano Galatolo, Cristóbal Rojas, Alexander Shen and Klaus Weihrauch [BGH$^+$11, GHR11b, Hoy11, HRW11, BDH$^+$12, Hoy12, HRW12, GHR12, Hoy13, CFHR22]. Together with Jason Rute, we have written a chapter on computable measure theory and algorithmic randomness in the Handbook of Computability and Complexity in Analysis [HR21].

I have worked on the complexity aspects of computable analysis, exploring the framework developed by Kawamura and Cook based on higher-order complexity theory. This work was done with my PhD student Hugo Férée and several colleagues, Walid Gomaa, Emmanuel Hainry and Romain Péchoux [FHHP10, FGH13, FGH14, FHHP15]. I have also worked on the complexity of convex sets with Alonso Herrera during his internship.

I have also studied problems on more isolated topics: fixed-point theorem in subrecursive languages with Guillaume Bonfante, Mohamed El-Aqqad and Benjamin Greenbaum [BEGH15]; the problem of extending computable real functions with Walid Gomaa [HG17]; the study of semicomputability for simple objects such as triangles, disks or polynomials, with Diego Nava Saucedo and Donald M. Stull [HSS18, HS19]; computability on quasi-Polish spaces with Cristóbal Rojas, Victor Selivanov and Donald M. Stull [HRSS19]; the Scott topology over spaces of open sets with Émile Larroque during his M1 internship; Weihrauch reducilibity and descriptive complexity with Hervé Sabrié during his L3 internship; computability of Polish spaces with Takayuki Kihara and Victor Selivanov [HKS20]; computability of compact sets with my PhD student Djamel Eddine Amir [AH22].

Chapter 2

Problems

Part II is divided in 5 chapters, each one presenting the development of a specific topic. The 5 sections of this chapter are brief overviews of those 5 chapters, with for each one a short presentation of the corresponding problem as well as an informal statement of our answer to the problem.

The purpose of this chapter is to be accessible to the non-expert reader. The problems are deliberately presented outside of any context, in particular with little reference to the literature. These information will be given in the corresponding chapters.

2.1 Comparing two representations

A mathematical object usually has many facets and can be represented in various ways, by providing some piece of information about the object. A natural problem is then to understand whether two representations give different information. The term "information" is vague, but more precise questions can be formulated, using topology or computability:

- Is there a continuous procedure transforming a representation into the other?
- Is there a uniformly computable one?
- Is there an object that is computable w.r.t. one representation but not the other?

The topological and computability-theoretic formulations are related: for instance, a uniform computable translation must be continuous. However, the third question has no topological counterpart in general, and we want to fill this gap. The goal of relating this computability-theoretic question to topology is twofold: to improve our general understanding, and to provide a general result showing the existence of such objects by simply analyzing the topological properties of the representations.

In order to relate computability and topology, we need to restrict our attention to objects living in topological spaces with their natural representations. It leads to the following problem:

Problem

Given two topologies on a space, when do they induce different sets of computable objects?

Let us give a few examples of objects having several natural representations:

- A set of natural numbers $A \subseteq \mathbb{N}$ can be represented either by its characteristic function, or by an enumeration of A in an arbitrary order,

- A real number $x \in [0,1]$ can be represented either as a limit of a sequence of rational numbers converging at a given rate, or by giving the binary expansion of x,
- An absolutely continuous probability measure μ over $[0,1]$ can be represented either by giving the weights of the rational intervals, or by giving its Radon-Nikodym derivative, i.e. the unique function $f \in L^1([0,1])$ such that $\mu(A) = \int_A f \,\mathrm{d}\lambda$ where λ is the Lebesgue measure; here, f is described as an L^1-limit of rational polynomials for instance,
- A compact Polish space (X,τ) can be represented either by choosing a dense sequence and giving a complete metric inducing the topology and which is computable on that sequence, or by giving additionally the compact information, i.e. by enumerating the finite covers of X by basic open balls,
- A circle $C \subseteq [0,1]^2$ can be represented by giving its center and radius, or by a sequence of finite sets converging to C in the Hausdorff metric, or by a dense sequence in C, or by an enumeration of open balls covering its complement,
- A disk $D \subseteq [0,1]^2$ can be represented in the same ways as a circle.

2.1.1 Results: Generically weaker topology

We study this problem in Chapter 5.

We work on a set X endowed with two countably-based topologies τ and τ'. Each topology induces a particular representation of the points of X, in which a point $x \in X$ is described by listing its basic neighborhoods in that topology. We then say for short that x is τ-computable if it is computable w.r.t. the standard representation associated with τ, and similarly for τ'. Our problem is then to find conditions on τ, τ' implying the existence of a point that is τ'-computable but not τ-computable.

We give a partial answer in the case when τ is Polish and τ' is weaker than τ. The reason why we take τ Polish is that our answer is based on Baire category, which is a topological way to define a notion of *small* set, similar in spirit to the notion of sets of measure 0, but mathematically very different.

We then introduce a notion expressing in a way that τ' is *much weaker* than τ. Precisely, we say that τ' is **generically weaker** than τ if every subset of X on which τ' and τ are equivalent must be small. We introduce an effective version of this notion and prove the following:

Answer

If τ' is generically weaker than τ in an effective way, then there exists a τ'-computable point that is not τ-computable.

The construction of such a point is an application of the priority method with finite injury, a technique invented independently by Freidberg and Muchnik in the 50s to solve Post's problem, i.e. to build a c.e. set that is not computable but does not compute the halting problem.

It is interesting to note that most of the constructions of counter-examples in computable analysis are done by a more or less direct encoding of the halting problem (for instance, Pour-El Richards non-computable solution to the wave equation). Here, we are in a situation where computable analysis needs more advanced computability-theoretic arguments.

At first sight, the notion of a generically weaker topology may seem to be a strong condition. However, using results from Descriptive Set Theory, we show that it is essentially optimal to prove our result. More precisely, it is optimal when the existence of τ'-computable but not τ-computable points holds relative to any oracle.

We apply our result to a computability problem in the analysis of dynamical systems, namely the non-computability of the ergodic decomposition, which was the original motivation for developing all this work. We also present an extension of the result involving three topologies.

Some of the results are published in [Hoy14, Hoy17], others are new and appear for the first time in this manuscript.

2.2 Generic semicomputable objects

A common situation in mathematics is to prove the existence of objects satisfying many prescribed properties at the same time. Building such objects by hand can be rather difficult, and a powerful technique to build such objects for free is provided by Baire category. It helps building objects satisfying countably many properties by finding the appropriate topology making these properties prevalent, and the Baire category theorem then implies that their intersection is prevalent as well. Therefore it not only shows the existence of objects satisfying these properties, but that *typical* objects satisfy them. We mention Jones' excellent overview of such applications of Baire category in mathematical analysis [Jon99].

Computability theory and computable analysis are crowded with constructions of objects that are "semicomputable" in some sense[1], and satisfy countably many properties. We want to be able to apply Baire category arguments in this setting, in order to avoid repeating similar constructions again and again, and to obtain existence results in a much simpler way. However, ordinary Baire category arguments will not apply in this case, because the property of being semicomputable is usually small in the sense of Baire category. We want to make sense of questions like:

Problem

- What are the properties of a typical computably enumerable (c.e.) subset of $\mathbb{N}$?
- What are the properties of a typical left-c.e. real?
- What are the properties of a typical Π^0_1 disk in $\mathbb{R}^2$?
- Etc.

2.2.1 Results: Upper-genericity

In Chapter 6, we develop an effective version of Baire category that is compatible with semicomputability. It enables one to better understand the properties of semicomputable objects by studying the generic ones, rather than by building objects by hand.

Like in Chapter 5, we work on a set endowed with two topologies τ and τ'. We assume again that τ is Polish so that Baire category can be applied, and we assume that τ' is weaker than τ. We then define a notion of genericity w.r.t. τ and τ', and prove that it is indeed compatible with being τ'-computable.

[1] The word "semicomputable" is not a particular notion, but gathers all sorts of weak computability properties, like c.e. set, left-c.e. reals, Π^0_1-sets, etc.

The idea is to adapt the notion of 1-generic points. We recall that a point x is 1-generic (w.r.t. τ) if for every effective τ-open set U, if x is a limit w.r.t. τ of a sequence $x_n \in U$, then $x \in U$.

We define a weaker notion that involves τ' via its *specialization pre-order* $\leq_{\tau'}$: for two points x, y, define $x \leq_{\tau'} y$ if every τ'-open set containing x also contains y. We then say that x is **upper-generic** (w.r.t. τ, τ') if for every effective τ-open set U, if x is a limit w.r.t. τ of a sequence $x_n \in U$ *satisfying* $x \leq_{\tau'} x_n$, then $x \in U$.

We then prove that this notion of genericity is compatible with being τ'-computable, and therefore provides a notion of "typical" τ'-computable point:

Answer

Under mild computability assumptions, if τ is Polish and τ' is weaker than τ, then there exist upper-generic τ'-computable points.

As in the previous chapter, the construction uses the priority method with finite injury invented by Friedberg and Muchnik, and it captures many results involving this method, including Friedberg and Muchnik's original construction. Our general result can then be easily applied in many situations to build an object with prescribed properties, without having to carry out the construction again and again.

Finding more powerful results that would capture more advanced constructions is a challenging problem, left for future research.

The results are published in [Hoy14, Hoy17]. We apply them in other works [HG17, HSS18, HS19].

2.3 What does a representation allow to compute?

Given a particular way to represent a class of objects, we want to understand what information about the objects can be derived from this representation. A way to make this problem more precise is to investigate how difficult it is to test various properties using this representation.

A very interesting case is given by the polynomials with real coefficients. They are one of the most fundamental mathematical objects, but when it comes to choosing a representation in the sense of computable analysis, the choice is not neutral and leads to unusual phenomena, which we explain now. A little thinking leads to choosing the following representation: a real polynomial $P \in \mathbb{R}[X]$ is given by:

- An upper bound n on its degree,
- Usual representations of its coefficients $p_0, \ldots, p_n$, so that $P = p_0 + p_1 X + \ldots + p_n X^n$.

It can be argued that this representation is better than other natural candidates. For instance, requiring the exact degree of the polynomial would make addition non-computable because one cannot decide equality between real numbers.

This representation is natural and simple, but has unexpected singularities. To see this, let us first discuss the good properties of the representation of real numbers, that fail for the representation of real polynomials. The standard representation of real numbers (or of any countably-based topological space), for instance by Cauchy sequences of rational numbers converging at a given speed, is strongly connected to the topology on the space: there is a precise correspondence between the algorithmic complexity of a property and its topological complexity. For instance, for a subset $A \subseteq \mathbb{R}$,

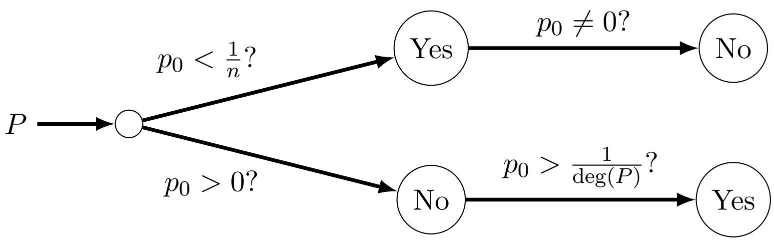
- A is semidecidable if and only if A is a computably enumerable (c.e.) open set,
- A is decidable with one mind-change if and only if both A and its complement can be expressed as differences of two c.e. open sets,
- A is decidable with n mind-changes if and only if both A and its complement can be expressed as differences of $n+1$ c.e. open sets,
- Etc.

It turns out that in the space $\mathbb{R}[X]$ with the representation described above and the topology induced by this representation, the algorithmic and topological complexity of subsets diverge. Moreover, the disagreement is witnessed by a very simple and concrete example, and not as the result of an artificial construction. This example is the set

$$A = \left\{ P \in \mathbb{R}[X] : p_0 = 0 \text{ or } p_0 > \frac{1}{\deg(P)} \right\}.$$

Let us quickly show that this set can be decided with one mind-change, but cannot be expressed as a difference of two open sets.

Algorithm. The following algorithm takes a polynomial $P = p_0 + \ldots + p_n X^n$ as input and decides whether $P \in A$ outputting Yes or No, possibly with one mind-change:



Note that all the inequalities are semidecidable. In particular, the inequality $p_0 > \frac{1}{\deg(P)}$ is semidecidable because it is equivalent to the existence of $k \in \mathbb{N}$ such that $p_k > 0$ and $p_0 > \frac{1}{k}$.

Topological complexity. One can see that A is not a difference of open set by considering the following converging double-sequence:

$$\frac{1}{n} + \frac{1}{p} X^{n+1} \xrightarrow[p\to\infty]{} \frac{1}{n} \xrightarrow[n\to\infty]{} 0.$$

Each application of the limit operator changes the location of the terms with respect to A:

$$\frac{1}{n} + \frac{1}{p} X^{n+1} \in A, \qquad \frac{1}{n} \notin A, \qquad 0 \in A.$$

It implies that A cannot be a difference of two open sets (in the same way as if a set B is open, a sequence that is outside B cannot converge to an element of B).

This example shows that the usual correspondence between computability and topology breaks down, so we want to investigate this disagreement and to understand why it happens.

Problem

What is the origin of the disagreement between algorithmic complexity and topological complexity? For which spaces and which complexity levels do they disagree?

2.3.1 Results: Descriptive complexity on represented spaces

Our answers to this problem are presented in Chapter 7.

The notions of complexity that are implicitly used in the previous discussion are provided by Descriptive Set Theory (DST). This theory is about the complexity of describing subsets of a topological space, starting from open sets and applying finite or countable boolean operations. Therefore, it provides a measure of **topological complexity** of sets. It comes with hierarchies such as the Borel hierarchy ($\utilde{\mathbf{\Sigma}}^0_\alpha, \utilde{\mathbf{\Pi}}^0_\alpha, \utilde{\mathbf{\Delta}}^0_\alpha$ for countable ordinals α) or the Hausdorff difference hierarchy ($\utilde{\mathbf{D}}_\alpha(\utilde{\mathbf{\Sigma}}^0_\beta)$ for countable ordinals α, β), and with a notion of reducibility between them by continuous functions, called Wadge reducibility.

Descriptive Set Theory only makes sense in topological spaces. However, it is possible to transfer concepts of DST from the Baire space to any represented space (X, δ_X) via the representation (even in the absence of a topology on X): the complexity of a set $A \subseteq X$ can be defined as the complexity of its preimage under the representation, namely $\delta_X^{-1}(A)$. As this preimage is the symbolic representation of the set A, we call this notion of complexity the **symbolic complexity** of the set A. Symbolic complexity is closely related to the algorithmic complexity illustrated above, because algorithms do not directly operate on objects, but on their symbolic representations.

In a topological space with an admissible representation, we therefore have two competing measures of complexity for a subset, namely topological and symbolic complexity. A representation is admissible if it is in some sense compatible with the topology, so one may think that symbolic and topological complexity coincide in that case. It is partially true: they indeed coincide in any countably-based topological space with an admissible representation, as proved by de Brecht and Yamamoto [dBY09, dB13]. However, in this chapter we study other admissibly represented topological spaces, such as the space of polynomials, and show that most of the time symbolic and topological complexity disagree. We derive precise comparisons between them in several classes of spaces, and relate their disagreement to the topological properties of the space.

Our results also explain why symbolic and topological complexities disagree: admissible representations are related to the *sequential* rather than the topological aspects of the space. This phenomenon was already known and mostly discovered by Schröder [Sch02a, Sch02b]. In topological spaces that are not countably-based, sequential and topological notions often differ. For instance, the sequential closure does not always coincide with the topological closure, and topological subspaces or topological products of sequential spaces are not always sequential. All our separation results between symbolic and topological complexity exploit these differences.

Answer

The disagreement between symbolic and topological complexity originates from the difference between sequential and topological notions.

This work was partially done with Antonin Callard during his L3 internship, and published in [CH20, Hoy20b].

2.4 What does a representation allow to compute?

Computable analysis meets higher-order programming, because they both give ways of computing with infinite objects, in particular higher-order functionals over $\mathbb{N}$. For instance, if the input to a higher-order program is a function of type $F : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$, i.e. a second-order function, how should F be presented to the program?

We only consider functions $F : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ that are continuous, i.e. such that for each $f : \mathbb{N} \to \mathbb{N}$, the value $F(f)$ only depends on a finite number of values of f. There are several models to define the interaction between a higher-order program and its input, for instance λ-calculus, Basic Feasible Functionals (BFF), game semantics, representations, Kleene-Kreisel functionals, etc.

Here we work with the Kleene-Kreisel model, which can be equivalently obtained using representations, and also from some version of game semantics. The access to an input $F : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ is done via a dialog between two players, Input and Program, where Input knows F and Program wants to evaluate F at some function $f : \mathbb{N} \to \mathbb{N}$. Program starts asking for the value of $F(f)$. As f is itself a function, it is not given at once but via a subdialog: Input asks Program for values $f(n)$ at specific numbers $n \in \mathbb{N}$. After a finite number of interactions, Input has obtained sufficient information about f and can finally answer Program's original request, i.e. the value of $F(f)$.

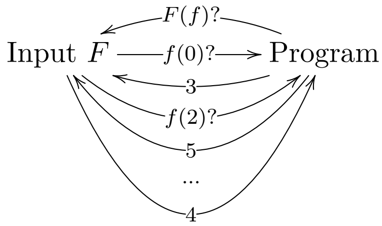


Figure 2.1: A dialog:
Program asks Input: what is the value of $F(f)$?
Input then asks Program for values of f;
Input eventually answers: $F(f) = 4$.

Given this access to F, what information does Program have about F? What are the properties of F that Program can decide or semidecide? In addition to the values taken by F, this dialog also gives information about how much of the input f is needed to evaluate $F(f)$, which enables one to decide or semidecide non-trivial properties about F. Let us list a few examples:

1. Whether F is not constant is semidecidable,
2. Let $f : \mathbb{N} \to \mathbb{N}$ be some computable function. Whether $F(f) = 0$ is decidable,
3. For each $n \in \mathbb{N}$, let $f_n : \mathbb{N} \to \mathbb{N}$ be some computable function, Whether $F(\lambda n.F(f_n)) = 0$ is decidable,
4. Let $f : \mathbb{N} \to \mathbb{N}$ be some computable function. Whether F is constantly 0 on $\{g : \mathbb{N} \to \mathbb{N} : g \leq f\}$ is decidable,
5. Let $F_0 : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ be computable. Whether $F \neq F_0$ is semidecidable,
6. etc.

Is it possible to establish a complete list of all the semidecidable properties, or a list which is sufficient to express all the semidecidable properties as combinations of these basic properties?

For comparison, when the input is a function $f : \mathbb{N} \to \mathbb{N}$, the semidecidable properties are generated by the basic properties "$f(n) = p$", in the sense that the class of semidecidable properties of such functions $f : \mathbb{N} \to \mathbb{N}$ is the smallest class of properties containing these basic properties and closed under finite intersections and computable unions. It gives a complete and concrete description of the decidable properties of first-order functions.

Describing the class semidecidable properties of functions $F : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ is a difficult problem which is currently unsolved. For want of anything better, one can first try to identify the difficulty of this problem, which is already challenging.

Problem

How difficult is it to list the semidecidable properties of continuous functionals of type $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$?

2.4.1 Results: Base-complexity of topological spaces

We study this problem in Chapter 8.

The difficult part is to give a lower bound. We do this by applying the diagonal argument. However we first need to extend the scope of this technique to situations where it does not immediately apply. The goal of this chapter is to present these results, in the more general context of base-complexity of topological spaces.

Admissibly represented topological spaces form a class of topological spaces with very good categorical properties. It is in particular a cartesian closed category, i.e. it has exponentials: if X, Y are two spaces in this class, then the space $\mathcal{C}(X, Y)$ of continuous functions from X to Y can be endowed with a natural admissible representation and a well-behaved topology. The good behavior is that continuous functions can be curried and uncurried: if $f : Z \times X \to Y$ is continuous then so is $f : Z \to \mathcal{C}(X, Y)$, and vice versa (note that the product space $Z \times X$ is not endowed with the product topology, but with its sequentialization).

The easiest way to describe the topology on $\mathcal{C}(X, Y)$ is to define a representation on $\mathcal{C}(X, Y)$, which can be naturally obtained from representations on X and Y, and then take the final topology of that representation. It is then very easy to form function spaces. The Kleene-Kreisel functionals, which are higher-order functionals over the natural numbers, can be obtained using this inductive construction:

- $\mathbb{N}\langle 0\rangle = \mathbb{N}$,
- $\mathbb{N}\langle n+1\rangle = \mathcal{C}(\mathbb{N}\langle n\rangle, \mathbb{N})$,

and it can be extended to $\mathbb{N}\langle\alpha\rangle$ for countable ordinals α.

The exponentiation in the category of admissibly represented spaces provides a topology for each such space. However, the definition of the topology as the final topology of the representation is implicit: it does not tell us what its open sets look like or how they can be described. Somehow, we have to deal with topological spaces for which we do not know what the open sets are (although they are formally well-defined), which makes their analysis difficult. Note that the open sets are closely related to the semidecidable properties discussed previously, because the semidecidable properties are nothing else than the open sets having a computable description.

Our goal is to better understand the open sets in these topological spaces. While finding a "concrete" description of them looks out of reach, we can precisely measure the minimal complexity of any such description. It is called the **base-complexity** of the topological space and was introduced and studied by de Brecht, Schröder and Selivanov in [dBSS16]. They defined the base-complexity hierarchy $\mathrm{Base}(\boldsymbol{\Sigma}^1_\alpha)$ and left the following problems open:

Problem

- What is the base-complexity of $\mathbb{N}\langle\alpha\rangle$? One has $\mathbb{N}\langle\alpha\rangle \in \mathrm{Base}(\boldsymbol{\Sigma}^1_{\alpha+1})$, is it optimal?
- One has $\mathrm{Base}(\boldsymbol{\Sigma}^1_{\alpha}) \subsetneq \mathrm{Base}(\boldsymbol{\Sigma}^1_{\alpha+3})$, is it optimal?

We solve the two questions:

Answer

- The base-complexity of $\mathbb{N}\langle\alpha\rangle$ is exactly $\boldsymbol{\Sigma}^1_{\alpha+1}$ (for $\alpha \geq 2$),
- The base-complexity hierarchy is proper, with $\mathrm{Base}(\boldsymbol{\Sigma}^1_{\alpha}) \subsetneq \mathrm{Base}(\boldsymbol{\Sigma}^1_{\alpha+1})$.

The proofs are unsurprisingly based on a diagonal argument. However, the diagonal argument cannot be directly applied, so we had to develop techniques to extend the scope of that argument. A particularly interesting observation is that although the results only involve single-valued functions, the proofs require using multi-valued functions. Therefore, multi-valued continuous functions are able to capture essential properties of topological spaces that continuous single-valued functions cannot. The results are published in [Hoy22].

2.5 Finite vs infinite representation

In computable analysis, infinite mathematical objects are represented by infinite sequences of symbols or numbers. However, most of the concrete objects that we deal with in mathematics have a finite description, which can be a name, a formula or a definition: for instance, the number $\pi = 4\sum_{n=0}^{\infty}\frac{(-1)^n}{2n+1}$, the exponential function $e^x = \sum_{n=0}^{\infty}\frac{x^n}{n!}$, etc. Most of the time this finite description can be translated into a finite program that computes the relevant information about the object (decimals of π, evaluation of e^x at any precision, etc.).

In terms of computability, is there a difference between having an infinite description of an object (the sequence of decimals of π), or a finite program that is able to produce such an infinite description? Does a finite description give strictly more information in general? What properties are decidable from finite descriptions? Is it possible to describe all of them? For instance, if the objects are computable functions on the natural numbers, the question can be formulated as follows.

Problem 1

If a computable function $f : \mathbb{N} \to \mathbb{N}$ is given as a finite program p, what information can be extracted from the code of p, beyond the values of f?

We are only concerned with information that is intrinsic to f, i.e. independent of the particular program p.

Let us examine another example of objects having finite descriptions: the class of primitive recursive functions, which can be described using programs in the LOOP language (a programming language with `for` loops but no `while` loop).

It turns out that finite descriptions indeed give strictly more information than infinite ones, which is witnessed by a rather simple property of primitive recursive functions that can be decided from finite descriptions, but not from infinite ones. Say that $f : \mathbb{N} \to \mathbb{N}$ is **compressible** if for every n, there exists a LOOP program of length $\leq n$ that computes the values of f on inputs $0, \dots, n$, but not necessarily on other inputs[2]. Intuitively, f is compressible if each prefix of f can be described in a short way.

Given a LOOP program p computing f, the following algorithm decides whether f is compressible:

- Measure the length k of p,
- For each $n < k$, decide whether there exists a program of length $\leq n$ computing the values of f on inputs $0, \dots, n$.

Note that the test in the second step is indeed decidable, because LOOP programs always halt so we can simply test all the programs of length at most n, one after the other. The definition of compressibility involves an infinite universal quantification over n, but knowing the length of p reduces it to a finite quantification, making the property decidable.

However if f is given by an oracle, i.e. if one only has access to the values of f, then the infinite quantification over n makes its compressibility undecidable. Indeed, no finite prefix of a compressible function f can guarantee its compressibility.

So we have an example of a property that is decidable from finite descriptions, but not from infinite ones. We want to know what are the other decidable properties, and to find a complete description of all the decidable properties.

Problem 2

What are the decidable properties of primitive recursive functions when they are given by LOOP programs?

2.5.1 Results: Markov computability vs Type-two computability

We investigate these problems in Chapter 9.

The possibility of using either infinite or finite representations to model computations with mathematical objects has given rise to two schools of computable mathematics. The Polish school, led by Grzegorczyk, is at the origin of the modern presentation of computable analysis formalized using type-two Turing machines and representations by Kreitz and Weihrauch [KW85]. The Russian school, led by Markov, uses ordinary type-one Turing machines and numberings of objects by finite indices.

The comparison between Markov and type-two computability has been an active research topic in the 50s. Although in many situations the two models are equivalent, they do differ. Friedberg was the first one to find an example witnessing the difference, from which the compressibility property presented above is inspired. However, when the two models are different, it is not clear what the difference is and how far the two models are from each other, and what additional information is contained in a finite representation, compared to an infinite one. Our partial answer to this question is that all the additional information is contained in the *size* of the finite representation. More precisely,

[2]There might be no compressible function because there is no program of length smaller than 10 for instance. In that case, the definition can be modified by requiring the compressibility condition for every $n \geq 10$ only.

Answer to Problem 1

Having the code of a program for a computable function $f : \mathbb{N} \to \mathbb{N}$ is equivalent to having:

- The values of f,
- Any upper bound on the size of any program for f.

By "equivalent", we mean that if an algorithmic task that can be performed from any program, then it can be performed from the values of f and an upper bound on the size of a program. In particular, no interesting information can be found in the code of the program, only in its size.

When restricting to subclasses of computable functions, we can go further and obtain a complete characterization of their decidable properties.

Answer to Problem 2

The semidecidable properties of primitive recursive functions given by LOOP programs are generated by the following properties:

- $f(n) = p$, where $n, p \in \mathbb{N}$,
- f is h-compressible, where h is computable non-decreasing unbounded function.

Here, h-compressibility is a straightforward generalization of the compressibility property presented above, requiring the existence of programs of lengths $h(n)$ computing the values of f on inputs $0, \ldots, n$.

We have obtained many other results on this topic, some of them are presented in Chapter 9. This work was partially done in collaboration with Cristóbal Rojas and published in [HR15, Hoy16a, HR17, Hoy22]. We have also written a popularization article on this topic in Images des Mathématiques [Hoy15].

Chapter 3

Publications

Here is a complete list of my publications, grouped by topic. All the articles are available from my website `https://members.loria.fr/MHoyrup/`.

Computable aspects of dynamical systems

- **Dynamical systems: stability and simulability.** M. Hoyrup. *Mathematical Structures in Computer Science* 2007 [Hoy07].
- **Computability and the morphological complexity of some dynamics on continuous domains.** M. Hoyrup, A. Kolçak and G. Longo. *Theoretical Computer Science* 2008 [HKL08].
- **A constructive Borel-Cantelli lemma. Constructing orbits with required statistical properties.** S. Galatolo, M. Hoyrup and C. Rojas. *Theoretical Computer Science* 2009 [GHR09b].
- **Computing the speed of convergence of ergodic averages and pseudorandom points in computable dynamical systems.** S. Galatolo, M. Hoyrup and C. Rojas. CCA 2010 [GHR10a].
- **Dynamics and abstract computability: computing invariant measures.** S. Galatolo, M. Hoyrup and C. Rojas. *Discrete and Continuous Dynamical Systems - A* 2011 [GHR11b].
- **Statistical properties of dynamical systems - simulation and abstract computation.** S. Galatolo, M. Hoyrup, and C. Rojas. *Chaos, Solitons and Fractals* 2012 [GHR12].
- **Computability of the ergodic decomposition.** M. Hoyrup. *Annals of Pure and Applied Logic* 2013 [Hoy13].
- **Realizing semicomputable simplices by computable dynamical systems.** D. Coronel, A. Frank, M. Hoyrup and C. Rojas. *Theoretical Computer Science* 2022 [CFHR22].

Algorithmic randomness

- **Randomness on computable probability spaces - A dynamical point of view.** P. Gács, M. Hoyrup and C. Rojas. STACS 2009 [GHR09a].

- **Computability of probability measures and Martin-Löf randomness over metric spaces.** M. Hoyrup and C. Rojas. *Information and Computation* 2009 [HR09c].
- **An application of Martin-Löf randomness to effective probability theory.** M. Hoyrup and C. Rojas. CiE 2009 [HR09a].
- **Applications of effective probability theory to Martin-Löf randomness.** M. Hoyrup and C. Rojas. ICALP 2009 [HR09b].
- **Effective symbolic dynamics, random points, statistical behavior, complexity and entropy.** S. Galatolo, M. Hoyrup, and C. Rojas. *Information and Computation* 2010 [GHR10b].
- **Algorithmic tests and randomness with respect to a class of measures**. L. Bienvenu, P. Gács, M. Hoyrup, C. Rojas, and A. Shen. *Proceedings of the Steklov Institute of Mathematics* 2011, [BGH+11].
- **Randomness on computable probability spaces - a dynamical point of view.** P. Gács, M. Hoyrup, and C. Rojas. *Theory of Computing Systems* 2011 [GHR11a].
- **Randomness and the ergodic decomposition.** M. Hoyrup. CiE 2011 [Hoy11].
- **A constructive version of Birkhoff's ergodic theorem for Martin-Löf random points.** L. Bienvenu, A. R. Day, M. Hoyrup, I. Mezhirov and A. Shen. *Information and Computation* 2012 [BDH+12].
- **The dimension of ergodic random sequences.** M. Hoyrup. STACS 2012 [Hoy12].
- **Layerwise Computability and Image Randomness.** L. Bienvenu, M. Hoyrup, and A. Shen. Theory of Computing Systems 2017 [BHS17].

Computable analysis

- **Computability of the Radon-Nikodym derivative.** M. Hoyrup, C. Rojas and K. Weihrauch. CiE 2011 [HRW11].
- **Computability of the Radon-Nikodym derivative.** M. Hoyrup, C. Rojas and K. Weihrauch. *Computability* 2012 [HRW12].
- **Irreversible computable functions.** M. Hoyrup. STACS 2014 [Hoy14].
- **Genericity of weakly computable objects.** M. Hoyrup. *Theory of Computing Systems* 2017 [Hoy17].
- **On the extension of computable real functions.** M. Hoyrup and W. Gomaa. LICS 2017 [HG17]
- **Semicomputable Geometry.** M. Hoyrup, D. Nava Saucedo and D. M. Stull. ICALP 2018 [HSS18]
- **Semicomputable Points in Euclidean Spaces.** M. Hoyrup and D. M. Stull. MFCS 2019 [HS19].

- **Computability on Quasi-Polish Spaces.** M. Hoyrup, C. Rojas, V. L. Selivanov and D. M. Stull. DCFS 2019 [HRSS19].
- **Descriptive Complexity on Non-Polish Spaces.** A. Callard and M. Hoyrup. STACS 2020 [CH20].
- **Descriptive Complexity on Non-Polish Spaces II.** M. Hoyrup. ICALP 2020 [Hoy20b].
- **The fixed-point property for represented spaces.** M. Hoyrup. *Annals of Pure and Applied Logic* 2022 [Hoy22].
- **Computability of finite simplicial complexes.** D.E. Amir and M. Hoyrup. ICALP 2022 [AH22].

Markov computability

- **On the information carried by programs about the objects they compute.** M. Hoyrup and C. Rojas. STACS 2015 [HR15].
- **The decidable properties of subrecursive functions.** M. Hoyrup. ICALP 2016 [Hoy16a].
- **On the information carried by programs about the objects they compute.** *Theory of Computing Systems* 2017 [HR17].

Complexity in analysis

- **Interpretation of stream programs: Characterizing type-2 polynomial time complexity.** H. Férée, E. Hainry, M. Hoyrup, and R. Péchoux. ISAAC 2010 [FHHP10].
- **On the query complexity of real functionals**. H. Férée, W. Gomaa, and M. Hoyrup. LICS 2013 [FGH13].
- **Higher-order complexity in analysis.** H. Férée and M. Hoyrup. CCA 2013 [FH13].
- **Analytical properties of resource-bounded real functionals.** H. Férée, W. Gomaa, and M. Hoyrup. *Journal of Complexity*, 2014 [FGH14].
- **Characterizing polynomial time complexity of stream programs using interpretations.** H. Férée, E. Hainry, M. Hoyrup, and R. Péchoux. *Theoretical Computer Science*, 2015 [FHHP15].

Miscellaneous

- **Rewriting logic and probabilities.** O. Bournez and M. Hoyrup. RTA 2003 [BH03].
- **Immune systems in computer virology.** G. Bonfante, M. El-Aqqad, B. Greenbaum, and M. Hoyrup. CiE 2015 [BEGH15]

Popularization

- **Une brève introduction à la théorie de l'aléatoire.** L. Bienvenu and M. Hoyrup. *Gazette des mathématiciens* 2010 [BH10].
- **Que calcule cet algorithme ?** M. Hoyrup. *Images des mathématiques* 2015 [Hoy15]

Book chapters

- **Algorithmic Randomness and Layerwise Computability**, M. Hoyrup, in *Algorithmic Randomness: Progress and Prospects*, Lectures Notes in Logic, Cambridge University Press (2020) [Hoy20a].

- **Computable Measure Theory and Algorithmic Randomness**, M. Hoyrup and J. Rute, in V. Brattka and P. Hertling, editors, *Handbook of Computability and Complexity in Analysis*, Springer, 2021 [HR21].

Invited talks

- **Effective probability theory.** Workshop "New Interactions between Analysis, Topology and Computation", 2009.

- **Algorithmic randomness and the ergodic decomposition.** AMS/ASL Meeting, special session *Logic and Analysis*, 2011.

- **Computably analysis and algorithmic randomness.** CCA 2011.

- **On the inversion of computable functions.** CCR 2012.

- **The information carried by programs about the objects they compute.** CCC 2015.

- **The typical constructible object.** CiE 2016, *Special session on Computable and Constructive Analysis* [Hoy16b].

- **Topological analysis of representations.** CiE 2018, *Special session on Continuous Computation* [Hoy18].

- **Semicomputable geometry.** CCR 2018.

- **Descriptive complexity on represented spaces.** CCA 2020.

- **The fixed-point property for represented spaces.** Computability Theory and Applications, online seminar, 2021.

- **Descriptive complexity of topological invariants.** WDCM 2021.

- **Realizing semicomputable simplices by computable dynamical systems.** Minisymposium at 2021 Joint Meeting of the DMV-ÖMG.

Part II

Detailed results

Chapter 4

Background

In this chapter, we give the most essential definitions that are needed to understand the next chapters.

We assume familiarity with basics of general topology, such as the notions of topological space, metric space, compact set, continuous function, homeomorphism, etc. We refer to the introductory textbook [HY88] where all these definitions can be found.

We also assume familiarity with basics of classical computability theory, i.e. with the notions of Turing machine, decidable or computable subset of $\mathbb{N}$, semidecidable or c.e. subset of $\mathbb{N}$ and computable partial function $f :\subseteq \mathbb{N} \to \mathbb{N}$.

Computability theory and computable analysis use and extend these notions to define computability of functions between other spaces than $\mathbb{N}$, typically uncountable spaces. Good references on computable analysis are [Wei00, BHW08, Pau16, Sch21].

4.1 The Baire space

The first step in extending computations from $\mathbb{N}$ to other spaces is to define the notion of a computable function on the Baire space.

The **Baire space** is the set $\mathcal{N}$ of infinite sequences of natural numbers, endowed with the topology generated by the cylinders: if $u \in \mathbb{N}^*$ is a finite sequence of natural numbers, then $[u] \subseteq \mathcal{N}$ is the set of infinite sequences extending u. Elements of $\mathcal{N}$ can also be seen as functions from $\mathbb{N}$ to $\mathbb{N}$.

We will work with oracle Turing machines, also called type-two Turing machines, and will simply call them Turing machines. Such a machine works with an oracle $p \in \mathcal{N}$ and an input $n \in \mathbb{N}$ and possibly outputs a natural number $m \in \mathbb{N}$. We then write $M^p(n) = m$. We say that a partial function $F :\subseteq \mathcal{N} \to \mathcal{N}$ is **computable** if there exists a Turing machine M such that for every $p \in \mathrm{dom}(F)$ and every $n \in \mathbb{N}$, $M^p(n) = q(n)$.

Extending computability from $\mathbb{N}$ to $\mathcal{N}$ comes with a new and fundamental phenomenon: computable functions from $\mathcal{N}$ to $\mathcal{N}$ are continuous. It comes from the fact that the output of an oracle Turing machine is produced in finite time, so it can only depend on a finite portion of the oracle. Thus, oracle Turing machines have two types of limitations:

- Logical limitations coming from their finite nature, making for instance the halting problem unsolvable,
- Topological limitations coming from the finite (although unbounded) access they have to an infinite oracle.

4.2 Represented spaces

Representations make it possible to extend computability from the Baire space to many spaces (virtually all spaces having the cardinality of the continuum).

Definition 4.2.1. Let X be a set. A **representation** of X is a surjective partial function $\delta_X :\subseteq \mathcal{N} \to X$. The pair (X, δ_X) is called a **represented space**.

Note that the Baire space could be equivalently replaced with the Cantor space of infinite binary sequences, because these two spaces computably embed into each other. For an element $x \in X$, any $p \in \mathcal{N}$ such that $\delta_X(p) = x$ is called a **name** of x. A point $x \in X$ is **computable** if it has a computable name. If (Y, δ_Y) is another represented space and $f :\subseteq X \to Y$ is a partial function, then a **realizer** of f is any partial function $F :\subseteq \mathcal{N} \to \mathcal{N}$ sending every name of every point $x \in \mathrm{dom}(f)$ to a name of $f(x)$, i.e. making the following diagram commute:

$$\begin{tikzcd} X \arrow[r, "f"] & Y \\ \mathcal{N} \arrow[u, "\delta_X"] \arrow[r, "F"] & \mathcal{N} \arrow[u, "\delta_Y"'] \end{tikzcd}$$

We say that f is **computable** if it has a computable realizer. A subset $A \subseteq X$ is **decidable** if its characteristic function $\chi_A : X \to \{0, 1\}$ is computable, which means that there exists a Turing machine that, given a name of any $x \in X$, halts and outputs 1 if $x \in A$ and 0 if $x \notin A$. The set A is **semidecidable** if the Turing machine halts if and only if $x \in A$. Equivalently, A is semidecidable if and only if there exists a set $U \subseteq \mathcal{N}$ such that $\delta_X^{-1}(A) = U \cap \mathrm{dom}(\delta_X)$ and U is **effectively open**, i.e. $U = \bigcup_{\sigma \in E} [\sigma]$ for some c.e. set $E \subseteq \mathbb{N}^*$.

4.3 Admissibly represented spaces

We saw that computable functions on the Baire space are continuous. Moreover, a continuous function on the Baire space is continuous if and only if it is computable relative to some oracle (available to the machine in addition to the input oracle).

It implies that computable functions between represented spaces must also be continuous for the appropriate topologies. Precisely, a representation $\delta_X :\subseteq \mathcal{N} \to X$ induces a topology called its **final topology**, whose open sets are the sets $U \subseteq X$ such that $\delta_X^{-1}(U)$ is an open subset of $\mathrm{dom}(\delta_X)$ (i.e. the intersection of an open subset of $\mathcal{N}$ with $\mathrm{dom}(\delta_X)$). It is an easy exercise to show that every computable function between represented spaces is continuous w.r.t. the final topologies of the representations.

One is naturally led to the following question: is it true that a function between represented spaces is continuous (w.r.t. the final topologies of the representations) if and only if it is computable relative to some oracle, or equivalently if it has a continuous realizer? The answer is negative in general, but is positive in many natural spaces, so we put this property into a definition.

Definition 4.3.1. A representation δ_X is **admissible** if every continuous partial function $f :\subseteq \mathcal{N} \to X$ has a continuous realizer.

Here, the space $\mathcal{N}$ can be seen as a represented space with the obvious representation given by the identity. If δ_X is admissible, then for every represented space (Y, δ_Y), a function $f : X \to Y$ is continuous if and only if it has a continuous realizer.

The next problem is to understand which topological spaces have an admissible representation. The answer was given by Schröder [Sch02a].

Theorem 4.3.1. *A topological space (X, τ) has an admissible representation if and only if it is T_0 and is a quotient of a countably-based space.*

We recall that a topological space (X, τ_X) is a quotient of a topological space (Y, τ_Y) if there exists a surjective function $f : Y \to X$ such that τ_X is the final topology of f, i.e. $U \subseteq X$ is τ_X-open iff $f^{-1}(U)$ is τ_Y-open.

4.4 Countably-based spaces

A particularly simple class of admissibly represented topological spaces is given by the countably-based T_0-spaces, developed in details in [Wei00]. A countably-based T_0-space (X, τ) with a numbered basis $(B_i)_{i \in \mathbb{N}}$ has a very concrete admissible representation called the **standard representation**, which encodes a point $x \in X$ by any enumeration of the set $\{i \in \mathbb{N} : x \in B_i\}$. More precisely, $p \in \mathcal{N}$ is a name of x if for all $i \in \mathbb{N}$,

$$x \in B_i \iff \text{there exists } n \in \mathbb{N} \text{ such that } p(n) = i + 1.$$

This representation is designed so that from a name of x, one can computably enumerate the set $\{i \in \mathbb{N} : x \in B_i\}$ but not decide this set in general.

An **effective countably-based T_0-space** is a countably-based T_0-space with a numbered basis $(B_i)_{i \in \mathbb{N}}$ such that intersection between open sets is effective: there exists a c.e. set $E \subseteq \mathbb{N}^3$ such that $B_i \cap B_j = \bigcup_{(i,j,k) \in E} B_k$ for all $i, j \in \mathbb{N}$.

Further relations between computability and topology can be formulated in such spaces. A subset $A \subseteq X$ is semidecidable if and only if it is **effectively open**, i.e. $A = \bigcup_{i \in E} B_i$ for some c.e. set $E \subseteq \mathbb{N}$. A function $f : X \to Y$ is computable if and only if it is **effectively continuous**, i.e. the pre-images $f^{-1}(B_i^Y)$ of basic open sets of Y are effective open subsets of X, uniformly in i.

We will often work in restricted classes of countably-based spaces, namely separable metric spaces, for which we need a computable structure. A **computable metric space** is a metric space (X, d) coming with a dense sequence $(s_i)_{i \in \mathbb{N}}$ such that the real numbers $d(s_i, s_j)$ are uniformly computable. A **computable Polish space** is an effective countably-based T_0-space that is computably homeomorphic to a complete computable metric space.

Chapter 5

Comparing two topologies

Contents

5.1 Introduction

Mathematical objects can usually be represented in several natural ways. Our general goal is to compare the "information" that different representations provide about the same objects. This problem can be precisely formulated in several ways, using topology or computability. We are interested in understanding the interactions between these different approaches, and in the following specific question: when do two representations induce different notions of computable points?

With the aim of comparing topology and computability, we will focus on admissible representations of topological spaces, more precisely the standard representations associated to countably-based T_0-topological spaces. They cover the most common spaces coming from mathematical analysis and are central in the theory of represented spaces, for instance they are the main object of study in [Wei00].

Our goal is then to reduce the problem of comparing the computable points of two representations to the comparison between their underlying topologies. The idea is that for natural representations of objects, topology already reflects most of their computability properties.

A famous example of such a reduction of computability to topology is provided by the work of Pour-El and Richards appearing in their book [PER89]. They show that for a linear function $f : X \to Y$ between Banach spaces, and under mild computability assumptions, the following are equivalent:

- f is continuous,
- f is computable,
- f sends each computable input x to a computable output $f(x)$.

In particular, if f is not continuous then their result automatically shows the existence of a computable input x whose image $f(x)$ is not computable. They applied this result to show that the wave equation has non-computable solutions. This result is important for several reasons. It gives a theoretic explanation of the empirical equivalence between computability and continuity. It is very handy: it captures a certain type of computability-theoretic constructions, which need not be repeated in each situation.

There have been other approaches to the understanding of functions that do not preserve computability, or to computably compare the information content of representations. Brattka [Bra99] proposed algebraic conditions implying that a function does not preserve computability, applicable for instance to the function sending a set to its topological boundary. The study of non-uniformly computable functions and piecewise continuous functions also gives information about this problem [BP10, Zie12, KP14, PdB12]. Kihara and Pauly [KP14] showed how computability-theoretic degrees can be seen as the degrees of points in topological spaces, and together with Ng [KNP19] exploited this correspondence to separate degrees by topological arguments, or conversely to compare topological spaces by analyzing the degrees of their points. There are many other articles about the comparison between different representations of objects, for instance real numbers [BH02] or subsets of Euclidean spaces or metric spaces [BW99, Mil02, Zie02, BP03, Ilj11, IS18].

The results presented in this chapter are reformulations and developments of the results published in [Hoy13, HR17].

5.1.1 Reduction to topology

Let X be a set endowed with two effective countably-based topologies τ, τ'. We assume that τ' is effectively weaker than τ, i.e. that the basic τ'-open sets are uniformly effectively τ-open. Equivalently, the identity $\mathrm{id} : (X, \tau) \to (X, \tau')$ is computable, so in particular τ-computable points are τ'-computable.

We want to understand when τ'-computable are or are not τ-computable. Most common examples fall into one of these three categories:

- $\mathrm{id} : (X, \tau') \to (X, \tau)$ is computable,
- $\mathrm{id} : (X, \tau') \to (X, \tau)$ is non-uniformly computable, i.e. there exists a countable decomposition $X = \bigcup_{n\in\mathbb{N}} X_n$ such that each restriction id_{X_n} is computable,
- There exists $x \in X$ which is τ'-computable but not τ-computable.

The second condition can be equivalently formulated as follows: for every x, there exists a Turing machine converting any τ'-name of x into a τ-name of x.

In the first two cases, τ'-computability implies τ-computability. Note that these three cases do not cover all the possibilities, however concrete examples usually belong to one of them. Our goal is to improve our understanding of the third case and develop a technique to prove it.

As often, it is very informative to approach a computability question by first relaxing it to a topological question, for which three cases do cover all the possibilities. A function f is σ-continuous if its domain can be decomposed as a countable union of sets, such that the restriction of f to each set is continuous. The three cases are:

- $\mathrm{id} : (X, \tau') \to (X, \tau)$ is continuous,
- $\mathrm{id} : (X, \tau') \to (X, \tau)$ is σ-continuous but not continuous,
- $\mathrm{id} : (X, \tau') \to (X, \tau)$ is not σ-continuous.

Answering the topological question first is usually easier, and helps solving the computability question. We will present a technique, applicable when τ is Polish, that essentially reduces the computability question to a topological one. We introduce a notion expressing that id is not σ-continuous in an effective way (Definition 5.3.1) and show that it implies the existence of a τ'-computable point that is not τ-computable (Theorem 5.3.2). This notion is topological with a touch of computability and we show that its purely topological counterpart is equivalent to $\mathrm{id} : (X, \tau') \to (X, \tau)$ not being σ-continuous (Theorem 5.2.1).

The idea of relating σ-homeomorphisms between topological spaces and the computability-theoretic degrees of their points was thoroughly investigated by Kihara, Pauly and Ng [KP14, KNP19].

5.2 Generically weaker topology

Let (X, τ) be a computable Polish space and let τ' be a weaker countably-based topology on X. Our first goal is to understand when $\mathrm{id} : (X, \tau') \to (X, \tau)$ is not σ-continuous; our next goal will be to understand when some τ'-computable point is not τ-computable.

We assume that τ' is *effectively* weaker than τ, i.e. that $\mathrm{id} : (X, \tau) \to (X, \tau')$ is computable; in other words, τ' comes with a numbered basis $(B'_i)_{i\in\mathbb{N}}$ such that each B'_i is an effective τ-open set, uniformly in i.

Say that $\mathrm{id} : (X, \tau') \to (X, \tau)$ is **σ-continuous** if X can be decomposed as a countable union $X = \bigcup_{n\in\mathbb{N}} X_n$ such that the restriction of id to each X_n is continuous. A common reason why such a decomposition is not possible is that any set $C \subseteq X$ on which id is continuous is *meager* w.r.t. τ: it implies that X cannot be decomposed into a countable union of such sets, because (X, τ) is a Polish space so it is not a countable union of meager sets by the Baire category theorem. We will see that this sufficient condition, which seems stronger at first, is actually necessary, up to restriction to some subspace (Theorem 5.2.1).

If C is a subset of X, then a topology on X induces a topology on C, obtained by intersecting the open sets with C. We say that τ and τ' *agree* on C if they induce the same topology on C, which is the same as saying that the restriction of $\mathrm{id} : (X, \tau') \to (X, \tau)$ to C is continuous.

Definition 5.2.1. We say that τ' is **generically weaker** than τ if every $C \subseteq X$ on which τ and τ' agree is meager in (X, τ).

As discussed above, if τ' is generically weaker than τ then $\mathrm{id} : (X, \tau') \to (X, \tau)$ is not σ-continuous so it is not non-uniformly computable. In other words,

Proposition 5.2.1. *If τ' is generically weaker than τ then there exists $x \in X$ such that no Turing machine translates τ'-names of x into τ-names of x.*

Moreover, the result relativizes: for every oracle A, there exist such an x that defeats all machines having A as oracle.

That τ' is generically weaker than τ is a sufficient condition to make $\mathrm{id} : (X, \tau') \to (X, \tau)$ not σ-continuous. It is not a necessary condition, because the discontinuity of id may happen only in a small set. We show that it is almost necessary: if $\mathrm{id} : (X, \tau') \to (X, \tau)$ is not σ-continuous, then it is possible to restrict to a subset on which τ' is generically weaker than τ. It is a consequence of a result by Solecki [Sol98], Pawlikowski and Sabok [PS09, PS12].

If $Y \subseteq X$, then let τ_Y and τ'_Y be the subspace topologies inherited from τ and τ' respectively.

Theorem 5.2.1. *Let (X, τ) be Polish and $\tau' \subseteq \tau$ be countably-based. The following are equivalent:*

- $\mathrm{id} : (X, \tau') \to (X, \tau)$ *is not σ-continuous,*
- *There exists $Y \subseteq X$ such that τ_Y is Polish and τ'_Y is generically weaker than τ_Y.*

We will prove this result in Section 5.4. We now investigate our notion of generically weaker topology.

5.2.1 Characterization

Checking that τ' is generically weaker than τ may be difficult, using the raw definition. We give a characterization that is easier to check in practice, and which will lead to an effective version.

The following notions witness that some τ-open sets are far from being τ'-open.

Definition 5.2.2 (Witness)**.** Let B be a non-empty τ-open set.

- An **X-witness** is a non-empty τ-open set $U \subseteq X$ that does not contain any non-empty τ'-open set,
- A **B-witness** is a non-empty τ-open set $U \subseteq B$ that does not contain any non-empty $V \cap B$ where V is τ'-open.

In other words, U is a B-witness if in the subspace B, U has empty τ'-interior. Said differently, U is a B-witness if every τ'-open set intersecting B also intersects $B \setminus U$, i.e. if $B \setminus U$ is τ'-dense in B.

The following algorithmic intuition is helpful: if U is a B-witness, then for any point $x \in U$, it is not possible to know in finite time that x belongs to U if we are given a name of x in the topology τ', even with the extra information that $x \in B$.

It is also helpful to have a formulation in terms of converging sequences, illustrated in Figure 5.1.

Proposition 5.2.2. *A non-empty open set $U \subseteq B$ is a B-witness if and only if every $x \in U$ is a limit, in the topology τ', of a sequence $x_n \in B \setminus U$.*

We now state and prove the main result of this section, which is a characterization of generically weaker topologies.

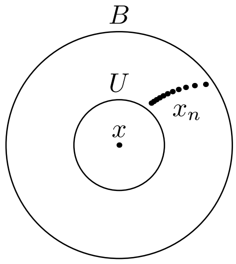


Figure 5.1: Illustration of Proposition 5.2.2: x_n converge to x in the topology τ'

Proposition 5.2.3 (Characterization of generically weaker topologies)**.** *τ' is generically weaker than τ if and only if every non-empty τ-open set B has a B-witness.*

In practice, it is usually easier to start showing the existence of an X-witness, and then adapting the proof to any subspace $B \in \tau$ in order to obtain a B-witness.

Proof. Assume that some non-empty $B \in \tau$ has no B-witness. We first show that for every non-empty τ-open set $U \subseteq B$, there exists $V \in \tau'$ such that $B \cap \mathrm{cl}_\tau(V) = B \cap \mathrm{cl}_\tau(U)$. We express U as the union of all the τ-basic open sets $U_i \subseteq U$. Each U_i is not a B-witness, so there exists $V_i \in \tau'$ such that $\emptyset \neq B \cap V_i \subseteq U_i$. Let $V = \bigcup_i V_i$. One has $B \cap V \subseteq U$, and $B \cap U \subseteq \mathrm{cl}_\tau(V)$ as V intersects each $B \cap U_i$. As a result, $B \cap \mathrm{cl}_\tau(V) = B \cap \mathrm{cl}_\tau(U)$.

Let $(U_n)_{n\in\mathbb{N}}$ be an enumeration of the basic τ-open sets contained in B and let $(V_n)_{n\in\mathbb{N}}$ be τ'-open sets such that $B \cap \mathrm{cl}_\tau(V_n) = B \cap \mathrm{cl}_\tau(U_n)$. Let $C = \bigcap_n (U_n \triangle V_n)^c$. By definition of C, τ' and τ agree on C. C is τ-co-meager in B, because each $U_n \triangle V_n$ is nowhere τ-dense in B, as $B \cap \mathrm{cl}_\tau(U_n \triangle V_n) \subseteq B \cap (\partial U_n \cup \partial V_n)$. Therefore, τ' is not generically weaker than τ.

Conversely, assume that each non-empty $B \in \tau$ has a B-witness. Let $C \subseteq X$ be such that τ' and τ agree on C, and let us show that C is nowhere dense. Given a non-empty τ-open set B, we need to find a non-empty τ-open set $W \subseteq B$ disjoint from C. Let U be a B-witness and B' be a non-empty τ-open set such that $\overline{B'} \subseteq U$. If B' is disjoint from C then take $W = B'$. If B' intersects C, then let $V \in \tau'$ be such that $B' \cap C = V \cap C$. V intersects U so V intersects $B \setminus U$ hence $B \setminus \overline{B'}$. Therefore $W = V \cap B \setminus \overline{B'}$ is non-empty and disjoint from C. □

We finally observe that in Definition 5.2.1, one can replace the condition of being "meager" by the apparently stronger condition of being "nowhere dense", giving the same notion.

Proposition 5.2.4 (Meager vs nowhere dense)**.** *Let $\tau' \subseteq \tau$ be generically weaker than τ. If τ' and τ agree on $C \subseteq X$, then C is nowhere dense in (X, τ).*

Proof. The idea is that we can always assume that C is a G_δ-set w.r.t. τ, and that a G_δ-set is meager if and only if it is nowhere dense.

More precisely, if τ and τ' agree on C, then let V_n be τ'-open sets such that $B_n \cap C = V_n \cap C$ for each n. The set $C' = \bigcap_n (B_n \triangle V_n)^c$ is a G_δ-set w.r.t. τ containing C, and τ' agrees with τ on C' because $B_n \cap C' = V_n \cap C'$ for each n. As a result, C' is meager. As C' is a G_δ-set, it is nowhere dense. As $C \subseteq C'$, C is nowhere dense as well. □

5.2.2 Examples

We give several examples of generically weaker topologies. They also illustrate how Proposition 5.2.3 makes it easy to show that a topology is generically weaker than another one.

Example 5.2.1 (Cantor vs Scott). Let (X, τ) be the Cantor space with the Cantor topology generated by the cylinders, and τ' be the Scott topology generated by the sets $\{x \in X : x_n = 1\}$, where $n \in \mathbb{N}$ and x_n is the bit of x at position n. It is easy to see that τ' is generically weaker than τ.

First, the cylinder $[0]$ contains no non-empty Scott open set so it is an X-witness. More generally, for any cylinder $[u]$, the cylinder $[u0]$ contains no Scott open set intersected with $[u]$, so $[u0]$ is a $[u]$-witness. ◁

Example 5.2.2 (Uniform norm vs L^1 norm). Let (X, τ) be the space $\mathcal{C}[0, 1]$ of continuous real functions with the topology of the uniform norm, and τ' be the L^1-topology. We show that τ' is generically weaker than τ.

First, the ball $B_\infty(0, 1) = \{f : \|f\|_\infty < 1\}$ is an X-witness, i.e. it contains no non-empty L^1-open set: for each $f \in B_\infty(0, 1)$, the functions $f_n = f + 2e^{-nx}$ converge to f in L^1-norm but do not belong to $B_\infty(0, 1)$. This argument can be applied inside any ball $B := B_\infty(g, r)$: the ball $U := B_\infty(g, r/3)$ is a B-witness, because if $f \in U$, then the functions $f_n = f + (2r/3)e^{-nx}$ converge to f in L^1-norm and belong to $B \setminus U$. ◁

Example 5.2.3 (Norms). More generally, let $\|.\|_1$ and $\|.\|_2$ be norms over a vector space (we assume that $\|.\|_1$ is separable and complete). If $\|.\|_2$ is strictly weaker than $\|.\|_1$ then the induced topology τ_2 is generically weaker than the induced topology τ_1, using the same argument as in the previous example: $B_1(x, r/3)$ is a $B_1(x, r)$-witness.

Note how the vector space structure has a strong impact on how norm-based topologies compare: if $\|.\|_1$ is weaker than $\|.\|_2$, then either they are equivalent, or $\|.\|_1$ is generically weaker than $\|.\|_2$, and there is no intermediate case. It is at the core of Pour-El and Richards first main theorem [PER89]. ◁

Example 5.2.4 (A non-example). Let (X, τ) be the space of real numbers with the Euclidean topology, let $f(x) = x^2$ and τ' the initial topology of f, consisting of the open subsets of $\mathbb{R}$ that are symmetric around 0. Although τ' is strictly weaker than τ, τ' is not generically weaker than τ.

Indeed, there is no $(0, +\infty)$-witness, because on $(0, +\infty)$, τ and τ' induce the same topology. Note that there exists an X-witness: the open interval $(1, 2)$ contains no non-empty τ'-open set. ◁

5.3 Effective version

We saw that if τ' is generically weaker than τ then there exist $x \in X$ whose τ-names cannot be computed from τ'-names. We now want the stronger property that x is τ'-computable but not τ-computable. In other words, we want to build such an x effectively. To achieve this, we need an effective version of the notion of a generically weaker topology, which we formulate using the characterization given by Proposition 5.2.3.

Main Definition 5.3.1: Effectively generically weaker

Say that τ' is **effectively generically weaker** than τ if every basic τ-open set B has a B-witness $U_B \subseteq B$ that can be computed from B.

Note that U_B can be assumed to be a basic τ-open set, and the computation takes an index of B as input and outputs an index of U_B.

In all concrete cases, the proof that τ' is generically weaker than τ gives explicit witnesses and actually shows that τ' is *effectively* generically weaker than τ. All the examples given in the previous section are effective. The following example is particularly interesting.

Example 5.3.1 (Norms). Example 5.2.3 is effective: if $\|.\|_1$ and $\|.\|_2$ are computable norms over a computable vector space, in the sense of Pour-El Richards [PER89], and $\|.\|_2$ is strictly weaker than $\|.\|_1$, then $\|.\|_2$ is effectively generically weaker than $\|.\|_1$ because witnesses are straightforward to compute. ◁

We now state the main result of this chapter, giving relatively simple conditions implying that τ and τ' do not induce the same computable points.

Main Theorem 5.3.2: τ'-computable but not τ-computable

Let (X, τ) be a computable Polish space and τ' an effectively weaker countably-based topology. If τ' is effectively generically weaker than τ then there exists a τ'-computable point that is not τ-computable.

Proof idea. The detailed proof can be found in [Hoy14] using a different formulation. Here we give the intuition. Consider a game between a Player and infinitely many Opponents, where Player describes a point x in the topology τ' and each Opponent tries to describe x in the topology τ. We prove that Player has a computable winning strategy. Therefore, if each Opponent is a Turing machine, then Player computes x in τ' while every Opponent fails to compute x in τ.

In a game against one Opponent only, the winning strategy of Player is simple. Let $U \in \tau$ be an X-witness. Player computes (in the topology τ') some x such that $x \in U$ if and only if Opponent never outputs U, which makes Opponent fail to compute x in the topology τ. To achieve this, Player starts describing some $x_0 \in U$ by enumerating $N'(x)$. If Opponent eventually guesses that x_0 belongs to U, i.e. enumerates U, then change x_0 to some x_1 outside U, but still in the current τ'-neighborhood of x_0, and continue enumerating the basic τ'-neighborhoods of x_1. Such an x_1 exists as the current τ'-neighborhood of x_0 is not contained in U by choice of U. In that case, Opponent does not describe $x = x_1$ as she enumerates U which does not contain x. If Opponent never guesses that $x \in U$ then $x = x_0$ belongs to U and again Opponent fails.

The strategy against infinitely many Opponents is almost the same, but we have to arrange the single strategies using the priority method with finite injury. The strategy S_e against Opponent number e is restrained by the strategies $S_0, \dots, S_{e-1}$, i.e. S_e has to keep x inside some τ-open set B. Player applies the strategy described using a B-witness U_B. U_B is in turn a restrain for the strategies against Opponents $e' > e$. □

Example 5.3.2 (Norms). We continue with Example 5.2.3 and Example 5.3.1. Let $\|.\|_1$ and $\|.\|_2$ be complete computable norms over a computable vector space, in the sense of Pour-El Richards [PER89]. If $\|.\|_2$ is strictly weaker than $\|.\|_1$, then $\|.\|_2$ is effectively generically weaker than $\|.\|_1$, so by Theorem 5.3.2 there exists a point x that is computable in norm $\|.\|_2$ but not in the norm $\|.\|_1$. It is a particular case of Pour-El Richards First Main Theorem [PER89]. Pour-El and Richards' full result can actually be recovered by working in the Polish space formed by the graph of the linear operator, details are explained in [Hoy14]. ◁

If x is a point provided by Theorem 5.3.2, then τ-names of x contain strictly more information than τ'-names of x. The proof of Theorem 5.3.2 can be adapted to show what type of non-computable information can be encoded in τ-names, that cannot be obtained from τ'-

names. Namely, computing τ-names from τ'-names is at least as hard as computing a limit, or equivalently the Turing jump.

Theorem 5.3.3 (Weihrauch reduction of lim is necessary)**.** *If τ' is effectively generically weaker than τ, then* lim *is Weihrauch reducible to* $\mathrm{id} : (X, \tau') \to (X, \tau)$.

We do not know whether a strong Weihrauch reduction can be obtained.

5.3.1 Application: non-computability of the ergodic decomposition

We originally proved Theorem 5.3.2 in order to analyse the non-computability of the ergodic decomposition, that we briefly explain now.

Dynamical systems naturally induce interesting geometrical objects. A dynamical system is a pair (X, T) where X is a compact metric space and $T : X \to X$ a continuous map. The orbit of a point $x \in X$ is the sequence starting at $x_0 = x$ and obtained by iterating T, i.e. $x_{n+1} = T(x)$. The main goal of the theory of dynamical systems is to understand how the orbits of points behave. In particular, ergodic theory is the study of the statistical distributions of orbits, which are expressed by T-invariant probability measures.

To each dynamical system (X, T) is associated its set of invariant probability measures. It is a rich geometrical object called a *Choquet simplex*: it is a compact convex set, and every point of that set can be uniquely decomposed as a convex combination of its extremal points, called the *ergodic* measures. Therefore, it is an infinite-dimensional analog of simplices like triangles or tetrahedra.

We are interested in the computability of the ergodic decomposition: given an invariant probability measure, is it possible to compute its decomposition as a convex combination of ergodic measures? Note that the decomposition is usually an integral over *continuously* many ergodic measures, so let us consider the simplest case of a *finite* decomposition:

Problem

Let μ be a computable invariant measure which is the average of two ergodic measures, $\mu = \frac{\mu_1+\mu_2}{2}$. Are μ_1, μ_2 computable?

We have been puzzled by this question for a few years, but could eventually give a negative answer, presented in [Hoy13, Hoy14]. The dynamical system is the shift map operating on the Cantor space of infinite binary sequences.

Answer

There exist two non-computable ergodic shift-invariant measures μ_1, μ_2 whose average $\frac{\mu_1+\mu_2}{2}$ is computable.

It is an interesting example from mathematical analysis for which a direct construction does not seem to be possible, and for which computability-theoretic techniques like the priority method seem to be needed. We developed the content of this chapter to simplify the construction by formulating it in an abstract setting, leading to Theorem 5.3.2.

Let us see how to apply Theorem 5.3.2 to solve this problem. This problem is really about comparing two representations on a single set. The set X is the set of pairs of ergodic shift-invariant measures. The representations δ, δ' are defined as follows. A δ-name of a pair (μ_1, μ_2) is simply a pair of names of μ_1 and μ_2. A δ'-name of (μ_1, μ_2) is a name of their average $\frac{\mu_1+\mu_2}{2}$.

Note that δ' is indeed well-defined, i.e. is a function, as the average uniquely determines the pair due the uniqueness of the ergodic decomposition.

These representations are the standard representations of two effective countably-based topologies τ, τ' such that τ is Polish, so thanks to Theorem 5.3.2 we just have to prove that τ' is generically weaker than τ, and check that this prove is effective. Therefore, we have translated a computability question into a topological one, which is much easier to answer.

Other dynamical systems. Our answer is very specific to a particular dynamical system, namely the shift map, which has a remarkable topological property: in the simplex of shift-invariant measures, the extremal points (the ergodic measures) are dense. It is then interesting to investigate the computability of the ergodic decomposition for dynamical systems whose simplex of invariant measures is less rich. It turns out that the situation can be as bad even in the simplest case where the simplex is one-dimensional, i.e. is a segment, and therefore has just two extremal points.

We have studied this problem with Coronel, Frank and Rojas, and proved that every finite-dimensional Π^0_1 simplex can be realized as the set of invariant measures of a computable dynamical system. For a suitable choice of a Π^0_1 segment, there exists a computable dynamical system which has exactly two ergodic measures μ_1, μ_2 such that both are not computable, but their average is. Therefore, it gives another answer to the problem, with the difference that the dynamical system is not natural like the shift map, but built on purpose. The results appear in [CFHR22].

5.3.2 Level of generality of the results

We worked on a set X endowed with two countably-based topologies τ, τ', such that τ is Polish and τ' is weaker than τ. Let us briefly discuss how general these assumptions are.

Incomparable topologies. We assumed that τ' is weaker than τ, but it is a mild assumption. Indeed, if τ' is not weaker than τ, then we can replace τ with the topology τ_1 generated by τ and τ', which is also countably-based and T_0. This change in topology has no effect in terms of computability, because a τ'-computable point is τ_1-computable iff it is τ-computable. However, the results apply only if τ_1 happens to be Polish.

Functions. The results can be applied to study *computable invariance* of functions, introduced by Brattka in [Bra05]. A function $f : X \to Y$ between represented spaces X, Y is **computably invariant** if it sends computable points to computable points. If (X, τ_X) and (Y, τ_Y) are countably-based T_0-spaces with their standard representations, then the study of a function f from X to Y can be turned into the comparisons between two topologies on a single space.

Indeed, let $Z = \mathrm{Graph}(f) \subseteq X \times Y$, let τ be induced by the product topology on $X \times Y$ and let τ' be the initial topology of the projection $\pi_X : Z \to X$, whose open sets are the sets $\pi_X^{-1}(U)$ with $U \in \tau_X$, so that (Z, τ') and (X, τ_X) are computably homeomorphic. Note that τ' is effectively weaker than τ. A point $(x, f(x))$ is τ-computable iff x is τ_X-computable and $f(x)$ is τ_Y-computable, and $(x, f(x))$ is τ'-computable iff x is τ_X-computable.

Non-Polish topology. We assumed that τ is Polish. It is a deliberately strong assumption which is at the core of our analysis. In Section 5.5, we give an extension of the result that can be applied when τ itself is not Polish, but for which a stronger Polish topology τ is available.

Non-countably-based topologies. Finally, we did not investigate what happens when the topologies are not countably-based but still have an admissible representation. This more general case seems to be a completely different story, because most of the arguments rely on specific properties of countably-based topologies. For instance, in a countably-based space, computing a point amounts to enumerating a subset of $\mathbb{N}$. Also, we use Polish topologies on which Baire category arguments can be applied, but to our knowledge there is no analog of Polish spaces among non-countably-based spaces.

5.4 Consequences of Solecki and Pawlikowski-Sabok's theorem

We now show two results. The first one is the proof of Theorem 5.2.1 that when $\mathrm{id} : (X, \tau') \to (X, \tau)$ is not σ-continuous, τ' is generically weaker than τ after restricting to a subspace. The second result is another relationship between the discontinuity of id and the difference between τ'-computability and τ-computability, expressed in terms of Weihrauch reducibility. We gather these results in one statement. These results are not published. Together with Djamel Eddine Amir, we have recently rewritten this chapter as an article, which is currently a preprint [AH23].

Theorem 5.4.1. *Let (X, τ) be Polish and $\tau' \subseteq \tau$ be countably-based T_0 and let $f = \mathrm{id} : (X, \tau') \to (X, \tau)$. The following conditions are equivalent:*

1. *f is not σ-continuous,*
2. *There exists $Y \subseteq X$ such that τ_Y is Polish and τ'_Y is generically weaker than τ_Y,*
3. $\lim$ *is Weihrauch reducible to f relative to an oracle,*
4. $\lim$ *is strongly Weihrauch reducible to f relative to an oracle.*

In that case (Y, τ_Y) is even homeomorphic to the Baire space $\mathcal{N}$. The proof uses a strong result from Descriptive Set Theory, Pawlikowski-Sabok's theorem [PS09, PS12]. This theorem says that every sufficiently definable map is either σ-continuous, or contains one particular function P, which is not σ-continuous. Let us state this result precisely.

A subset A of a Polish space X is **analytic** if it is the image of a continuous function $f : \mathcal{N} \to X$. A set $A \subseteq X$ is **bianalytic** if both A and $X \setminus A$ are analytic. A topological space is **analytic** if it embeds as an analytic subset of a Polish space. A function $f : X \to Y$ between analytic spaces is **bianalytic** if the preimage of every bianalytic set is bianalytic.

The Baire space $\mathcal{N} = \mathbb{N}^{\mathbb{N}}$ is endowed with two different topologies, both obtained as the product topology of some topology on $\mathbb{N}$. The first one $\tau_{\mathcal{N}}$ is the usual topology obtained from the discrete topology on $\mathbb{N}$. The second one, $\tau'_{\mathcal{N}}$, is obtained by identifying $\mathbb{N}$ with $\{0\} \cup \{2^{-n} : n \in \mathbb{N}\} \subseteq \mathbb{R}$, via $0 \mapsto 0$ and $n \mapsto 2^{-n}$ for $n > 0$. It makes $(\mathcal{N}, \tau'_{\mathcal{N}})$ homeomorphic to the Cantor space. Pawlikowski function P is defined as the identity from $(\mathcal{N}, \tau'_{\mathcal{N}})$ to $(\mathcal{N}, \tau_{\mathcal{N}})$.

Theorem 5.4.2 (Pawlikowski-Sabok [PS12])**.** *Let X, Y be analytic spaces and $f : X \to Y$ be bianalytic. Either f is σ-continuous or there exist two topological embeddings $\varphi : (\mathcal{N}, \tau'_{\mathcal{N}}) \to X$ and $\psi : (\mathcal{N}, \tau_{\mathcal{N}}) \to Y$ such that $f \circ \varphi = \psi \circ P$.*

This result was first proved by Solecki for Baire class 1 functions in [Sol98], and then improved in this form by Pawlikowski and Sabok. It was observed by Carroy a few years ago [Car21], and more recently by Lutz in [Lut21], that this result has a direct consequence in terms of Weihrauch reducibility: if P embeds in f as in the statement, then P is strongly Weihrauch reducible to f relative to an oracle.

We use the result in our setting, showing at the same time that some version of the result holds when X is not metrizable but still countably-based.

We start by reducing countably-based spaces to metrizable spaces. It is possible because the standard representation of countably-based spaces has very good properties, already exploited in [dBY09, dB13, CH20], that allow to reduce DST on countably-based spaces to DST on subspaces of the Baire space. We give another manifestation of this phenomenon, which is proved using similar techniques. Say that a function $f : X \to Y$ is **σ-computable** if there exist countably many sets $X_n \subseteq X$ such that $X = \bigcup_{n\in\mathbb{N}} X_n$ and each restriction $f_{|X_n} : X_n \to Y$ is computable.

Lemma 5.4.1 (σ-computable vs σ-computable realizer)**.** *Let X, Y be (effective) countably-based T_0-spaces with their standard representations. A function $f : X \to Y$ is σ-continuous (σ-computable) iff it has a σ-continuous (σ-computable) realizer.*

Proof. We prove the effective version, the non-effective version being obtained by relativization to any oracle.

One implication is straightforward. Assume that f is σ-computable, i.e. $X = \bigcup_{n\in\mathbb{N}} X_n$ and each $f\restriction_{X_n}$ is computable. We can assume that the sets X_n are pairwise disjoint, replacing X_n with $X_n \setminus (X_0 \cup \cdots \cup X_{n-1})$ if needed. Each $f\restriction_{X_n}$ has a computable realizer $F_n : \delta_X^{-1}(X_n) \to \mathcal{N}$. The combination of all F_n's is a σ-computable realizer of f.

We now prove the other implication. Assume that f has a σ-computable realizer $F : \mathrm{dom}(\delta_X) \to \mathcal{N}$, with $\mathrm{dom}(\delta_X) = \bigcup_{n\in\mathbb{N}} A_n$ and each $F\restriction_{A_n}$ is computable. For each $n \in \mathbb{N}$ and $\sigma \in \mathbb{N}^*$, let

$$X_{n,\sigma} = \{x \in \delta_X([\sigma]) : A_n \text{ is dense in } \delta_X^{-1}(x) \cap [\sigma]\},$$

where a set A is dense in a set B if B is contained in the closure of $A \cap B$.

Let us show that the restriction $f\restriction_{X_{n,\sigma}}$ is computable. Let $B_i \subseteq Y$ be a basic open set. We want to show that the preimage of B_i under this restriction is an effective open subset of $X_{n,\sigma}$, uniformly in i. As $\delta_Y \circ F$ is computable on A_n, there exists an effective open set $U_i \subseteq \mathcal{N}$, that can be computed uniformly in i, such that

$$U_i \cap A_n = (\delta_Y \circ F)^{-1}(B_i) \cap A_n = (f \circ \delta_X)^{-1}(B_i) \cap A_n. \tag{5.1}$$

Claim 5.4.1. One has

$$f^{-1}(B_i) \cap X_{n,\sigma} = \delta_X([\sigma] \cap U_i) \cap X_{n,\sigma}.$$

Proof of the claim. If $x \in X_{n,\sigma}$, then x has a δ_X-name $p \in [\sigma] \cap A_n$. If $x \in f^{-1}(B_i)$ then $p \in (f \circ \delta_X)^{-1}(B_i)$ so by (5.1), $p \in U_i$, which implies that $x \in \delta_X([\sigma] \cap U_i)$.

Conversely, if $x \in X_{n,\sigma}$ has a name $p \in [\sigma] \cap U_i$, then it has a name $q \in [\sigma] \cap U_i \cap A_n$ because A_n is dense in $[\sigma] \cap \delta_X^{-1}(x)$ and U_i is open. Again by (5.1), $q \in (f \circ \delta_X)^{-1}(B_i)$ so $x \in f^{-1}(B_i)$. □

The set $\delta_X([\sigma] \cap U_i)$ is an effective open set, uniformly in i, so $f\restriction_{X_{n,\sigma}}$ is computable.

It remains to show that $X = \bigcup_{n,\sigma} X_{n,\sigma}$. For $x \in X$, $\delta_X^{-1}(x)$ is Polish and is covered by $\bigcup_{n\in\mathbb{N}} A_n$, so some A_n must be somewhere dense in $\delta_X^{-1}(x)$. In other words, there must exist some $n \in \mathbb{N}$ and some $\sigma \in \mathbb{N}^*$ such that A_n is dense in $\delta_X^{-1}(x) \cap [\sigma]$, therefore $x \in X_{n,\sigma}$. □

Proof of Theorem 5.4.1. Of course, each one of conditions 2., 3. and 4. implies 1.

We show that 1. implies 2., 3. and 4. Let δ, δ' be representations of X that are admissible w.r.t. τ and τ' respectively. We can assume that they are open and that δ is total as τ is Polish. Moreover, by embedding (X, τ') in $\mathcal{P}(\omega)$ we assume that δ' is the restriction of the total open representation $\delta_{\mathcal{P}(\omega)}$ of $\mathcal{P}(\omega)$. Assume that f is not σ-continuous. We apply Theorem 5.4.2

to $g := f \circ \delta' : \mathrm{dom}(\delta') \to X$, where X is endowed with the Polish topology τ. We need to check that $\mathrm{dom}(\delta')$ is analytic and g is bianalytic. As $\mathrm{id} : (X, \tau) \to (X, \tau')$ is continuous, it has a continuous realizer $I : \mathcal{N} \to \mathrm{dom}(\delta')$, satisfying $\delta' \circ I = \delta$. The set

$$\begin{aligned} R &= \{(p, q) \in \mathcal{N} \times \mathcal{N} : \delta'(p) = \delta(q)\} \\ &= \{(p, q) \in \mathcal{N} \times \mathcal{N} : \delta'(p) = \delta' \circ I(q)\} \\ &= \{(p, q) \in \mathcal{N} \times \mathcal{N} : \delta_{\mathcal{P}(\omega)}(p) = \delta_{\mathcal{P}(\omega)} \circ I(q)\} \end{aligned}$$

is a $\undertilde{\mathbf{\Pi}}^0_2$-subset of $\mathcal{N} \times \mathcal{N}$, therefore its first projection, which is exactly $\mathrm{dom}(\delta')$, is analytic.

Let $A \subseteq (X, \tau)$ be bianalytic. Its preimage by g is

$$\begin{aligned} g^{-1}(A) &= \{p \in \mathrm{dom}(\delta') : f \circ \delta'(p) \in A\} \\ &= \{p \in \mathrm{dom}(\delta') : \exists q \in \mathcal{N}, \delta'(p) = \delta(q) \text{ and } \delta(q) \in A\} \end{aligned}$$

which is analytic. Applying the same argument to $g^{-1}(X \setminus A)$, we obtain that $g^{-1}(A)$ is bianalytic. Therefore g satisfies the assumptions of Theorem 5.4.2.

Now, g is not σ-continuous. Indeed, $g = f \circ \delta'$ has the same realizers as f. Applying Lemma 5.4.1 in one direction to f and in the other direction to g, we have: f is not σ-continuous, so it has no σ-continuous realizer as well as g, so g is not σ-continuous.

We can now apply Theorem 5.4.2 to g, which provides topological embeddings

$$\begin{aligned} \psi &: (\mathcal{N}, \tau_{\mathcal{N}}) \to (X, \tau), \\ \varphi &: (\mathcal{N}, \tau'_{\mathcal{N}}) \to (\mathcal{N}, \tau_{\mathcal{N}}), \end{aligned}$$

satisfying $g \circ \varphi = \psi \circ P$, i.e. $f \circ \delta' \circ \varphi = \psi \circ P$. As observed by Carroy [Car21], it implies that P is strongly Weihrauch reducible to f relative to an oracle computing φ and ψ. We have proved condition 4., which also implies condition 3.

Let us show condition 2. Both f and P are the identity functions, with different topologies on their input and ouput spaces. Therefore, the condition $f \circ \delta' \circ \varphi = \psi \circ P$ can be rewritten as $\delta' \circ \varphi = \psi$. As a result, we have:

(i) $\psi : (\mathcal{N}, \tau_{\mathcal{N}}) \to (X, \tau)$ is a topological embedding,

(ii) $\psi = \delta' \circ \varphi : (\mathcal{N}, \tau'_{\mathcal{N}}) \to (X, \tau')$ is continuous.

Let $Y = \mathrm{im}(\psi)$ and $\tau''_{\mathcal{N}}$ be the topology on $\mathcal{N}$ obtained as the preimage of τ'_Y by ψ and denoted by $\tau''_{\mathcal{N}} = \psi^{-1}(\tau'_Y)$ (it is the usually called the initial topology of ψ). Both functions

$$\begin{aligned} \psi &: (\mathcal{N}, \tau_{\mathcal{N}}) \to (Y, \tau_Y), \\ \psi &: (\mathcal{N}, \tau''_{\mathcal{N}}) \to (Y, \tau'_Y) \end{aligned}$$

are homeomorphisms, so we just need to prove that $\tau''_{\mathcal{N}}$ is generically weaker than $\tau_{\mathcal{N}}$.

By (ii), $\tau''_{\mathcal{N}}$ is weaker than $\tau'_{\mathcal{N}}$ which is generically weaker than $\tau_{\mathcal{N}}$, so $\tau''_{\mathcal{N}}$ is also generically weaker than $\tau_{\mathcal{N}}$. As a result, τ'_Y is generically weaker than τ_Y. Note that τ_Y is Polish because (Y, τ_Y) is homeomorphic to $(\mathcal{N}, \tau_{\mathcal{N}})$. □

Although Theorem 5.4.1 shows that the topological versions of Weihrauch reducibility and strong Weihrauch reducibility are equivalent in this context, we do not know whether their computable versions are still equivalent. In particular, we do not know whether Theorem 5.3.3 can be improved to obtain a strong Weihrauch reduction.

Open Question 5.4.3: Weihrauch vs strong Weihrauch reducibility

Let (X, τ) be an effective Polish space and τ' be a countably-based T_0-topology that is effectively weaker than τ. Are the following statements equivalent?

- lim is Weihrauch reducible to $\mathrm{id} : (X, \tau') \to (X, \tau)$,
- lim is strongly Weihrauch reducible to $\mathrm{id} : (X, \tau') \to (X, \tau)$.

5.5 Three topologies

The previous results have limitations. In this section, we propose a generalization that can be applied in other situations, and briefly discuss a concrete application. The results of this section are not published. We will see that the effective notion is more cumbersome (Definition 5.3.1), so probably more difficult to apply. However, as explained at the end of the section, we recently needed to carry out a construction which requires the analysis presented here, at least implicitly.

Let τ_1, τ_2 be two topologies on a set X, where τ_1 is weaker than τ_2. Our goal is again to build a point x that is τ_1-computable but not τ_2-computable. The previous results cannot be applied if for instance:

- The topology τ_2 is not Polish,
- The topology τ_2 is Polish, but the sets where τ_1 and τ_2 agree are not τ_2-meager.

In this section, we show how the results can be extended when there exists a third topology τ that is Polish and stronger than τ_2. One can think of τ_1 and τ_2 as the topologies one is interested in, and of τ as an auxiliary topology which drives the construction.

5.5.1 τ-generically weaker topology

Let (X, τ) be a computable Polish space and τ_1, τ_2 be effective countably-based topologies such that τ_1 is effectively weaker than τ_2 and τ_2 is effectively weaker than τ. In other words, the following functions are computable:

$$(X, \tau) \xrightarrow{\mathrm{id}} (X, \tau_2) \xrightarrow{\mathrm{id}} (X, \tau_1)$$

We want to build a τ_1-computable point that is not τ_2-computable. The previous results can be extended to this more general setting.

Definition 5.5.1. Say that τ_1 **is τ-generically weaker than** τ_2 if every set $C \subseteq X$ on which τ_1 and τ_2 agree is τ-meager.

Again, "meager" can be equivalently replaced by "nowhere dense", with the same argument as in Proposition 5.2.4. This notion is a generalization of Definition 5.2.1, which can be obtained by taking $\tau_1 = \tau'$ and $\tau_2 = \tau$.

Proposition 5.2.1 has an immediate analog.

Proposition 5.5.1. *If τ_1 is τ-generically weaker than τ_2, then there exists x in X such that no Turing machine translates τ_2-names of x into τ_1-names of x.*

Again, we give more concrete characterizations of this notion, which will eventually lead to an effective version.

Proposition 5.5.2 (Characterization of τ-generically weaker topologies)**.** *The following statements are equivalent:*

1. *τ_1 is τ-generically weaker than τ_2,*

2. *For every non-empty $B \in \tau$, there exists $U \in \tau_2$ such that there is no $V \in \tau_1$ satisfying*

$$B \cap \mathrm{cl}_\tau(U) = B \cap \mathrm{cl}_\tau(V),$$

3. *For every non-empty $B \in \tau$, there exist non-empty τ-open sets $B', B'' \subseteq B$ such that*

$$\begin{gathered} B' \textit{ is disjoint from } \mathrm{cl}_{\tau_2}(B''), \\ \textit{and } B' \textit{ is contained in } \mathrm{cl}_{\tau_1}(B''). \end{gathered}$$

In condition 2., one can restrict B and U to be basic open sets in their respective topologies. In condition 3., B and B' can also be assumed to be basic open sets, but not B''.

Proof. 1. $\Rightarrow$ 2. Let $B \in \tau$ be non-empty. We assume that for every $U \in \tau_2$ there exists $V \in \tau_1$ such that $B \cap \mathrm{cl}_\tau(U) = B \cap \mathrm{cl}_\tau(V)$, and show that τ_1 is not τ-generically weaker than τ_2. Let $(U_n)_{n\in\mathbb{N}}$ be an enumeration of the basic τ_2-open sets, and let $(V_n)_{n\in\mathbb{N}}$ be the corresponding τ_1-open sets. Let $C = \bigcap_n (U_n \triangle V_n)^c$. By definition of C, each U_n coincides with V_n on C, so τ_1 and τ_2 agree on C. We show that A is τ-dense in B. Each $(U_n \triangle V_n)^c$ is a G_δ-set for the topology τ, so it is sufficient to show that each one of them is τ-dense in B. The τ-closure of its complement is $\mathrm{cl}_\tau(U_n \triangle V_n)$, and its intersection with B is contained in $\partial U_n \cup \partial V_n$, so it is nowhere τ-dense in B. Therefore, A is τ-dense (even co-meager) in B. As a result, τ_1 is not τ-generically weaker than τ_2.

2. $\Rightarrow$ 1. Assume condition 2. holds. Let C be such that τ_1 and τ_2 agree on C. We show that C is nowhere τ-dense. Let $B \in \tau$ be non-empty, we want to show that B contain a non-empty τ-open W set disjoint from A. Let $U \in \tau_2$ come from condition 2. applied to B. As τ_1 and τ_2 agree on C, there exists $V \in \tau_1$ such that $U \cap C = V \cap C$. Let

$$W = B \cap \big(U \setminus \mathrm{cl}_\tau(V) \cup V \setminus \mathrm{cl}_\tau(U)\big).$$

W is disjoint from C, is contained in B and is non-empty: if it was empty, then $B \cap \mathrm{cl}_\tau(U) = B \cap \mathrm{cl}_\tau(V)$ would hold.

2. $\Rightarrow$ 3. Let $B \in \tau$ and let $U \in \tau_2$ be such that there is no $V \in \tau_1$ satisfying $B \cap \mathrm{cl}_\tau(U) = B \cap \mathrm{cl}_\tau(V)$. Let V be the maximal τ_1-open set such that $B \cap V \subseteq \mathrm{cl}_\tau(U)$. One has $B \cap \mathrm{cl}_\tau(V) \subseteq B \cap \mathrm{cl}_\tau(U)$ so the inclusion is strict. Therefore, there exists a non-empty τ-open set $B' \subseteq B \cap U$ that is disjoint from V. Take $B'' = B \setminus \mathrm{cl}_\tau(U)$. As $U \in \tau_2$ contains B' and is disjoint from B'', one has $B' \cap \mathrm{cl}_{\tau_2}(B'') = \emptyset$. If a τ_1-open set W intersects B', then $W \cap B \not\subseteq \mathrm{cl}_\tau(U)$ so W intersects $B \setminus \mathrm{cl}_\tau(U) = B''$. Therefore, $B' \subseteq \mathrm{cl}_{\tau_1}(B'')$.

3. $\Rightarrow$ 2. Let $B \in \tau$ and let $B', B'' \subseteq B$ satisfy $B' \cap \mathrm{cl}_{\tau_2}(B'') = \emptyset$ and $B' \subseteq \mathrm{cl}_{\tau_1}(B'')$. Define $U = X \setminus \mathrm{cl}_{\tau_2}(B'')$. Assume for a contradiction that there exists $V \in \tau_1$ such that $B \cap \mathrm{cl}_\tau(U) = B \cap \mathrm{cl}_\tau(V)$. B' is contained in $B \cap U \subseteq B \cap \mathrm{cl}_\tau(V)$, so V intersects B'. As $B' \subseteq \mathrm{cl}_{\tau_1}(B'')$, V intersects B''. It implies that U intersects B'', which contradicts the definition of U. $\square$

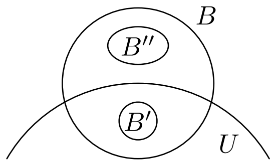


Figure 5.2: Illustration of Definition 5.5.1

5.5.2 Effective version and main result

We introduce an effective version of being generically τ-weaker, which will lead to Theorem 5.3.2. It is indeed an effective version, because the plain notion is obtained by relativization to an oracle.

Again (X, τ) is a computable Polish space and τ_1, τ_2 are countably-based topologies such that τ_1 is effectively weaker than τ_2 and τ_2 is effectively weaker than τ.

Main Definition 5.5.1: Effectively generically weaker

Say that τ_1 is **effectively τ-generically weaker** than τ_2 if given $B \in \tau$, one can compute non-empty $B', B'' \in \tau$ and $U \in \tau_2$ such that:

- $B' \subseteq B \cap U$,
- $B'' \subseteq B \setminus U$,
- $B' \subseteq \mathrm{cl}_{\tau_1}(B'')$, i.e., every $V \in \tau_1$ intersecting B' intersects B''.

Here, B, B' and U are basic open sets represented by indices, but B'' may be a general effective open set. Note that when $\tau_2 = \tau$, this definition is equivalent to Definition 5.3.1 (in one direction, take $U = B' = U_B$ and $B'' = B \setminus \overline{U}_B$; in the other direction, take $U_B = B'$).

It is an effective version of condition 3. in Proposition 5.5.2, as the set U witnesses that B' is disjoint from $\mathrm{cl}_{\tau_2}(B'')$. We now state the main result of this section.

Main Theorem 5.5.2: τ_1-computable but not τ_2-computable

If τ_1 is effectively τ-generically weaker than τ_2, then there exists $x \in X$ that is τ_1-computable but not τ_2-computable.

Proof idea. The proof is similar to the proof of Theorem 5.3.2. It is an application of the priority method with finite injury. The whole construction is an algorithm building some x and enumerating its basic τ_1-neighborhoods. For each machine M there is a strategy to make sure that M does not enumerate the basic τ_2-neighborhoods of x. These strategies are organized using the priority method. Each strategy is given a restrain by strategies with higher priorities and imposes restrains to lower priority strategies. A restrain imposed to a strategy is a ball $B \in \tau$, which means that the strategy is allowed to move the current point x inside B only (a ball is the analog of a cylinder in a construction on the Cantor space).

We simply show the strategy against one machine attempting to enumerate the basic τ_2-neighborhoods of x. We are given a restrain B, i.e. we are not allowed to move the point x outside B. We compute the corresponding B', B'' and U from Definition 5.5.1. We restrain the requirements with lower priority to stay in B'. If M eventually outputs (the index of) U, then let $V \in \tau_1$ be the current τ_1-neighborhood of x corresponding to the current enumeration of the basic τ_1-neighborhoods of x. As $x \in B' \cap V$, V intersects B' so it intersects B''. Therefore, we can move from x to some $x' \in V \cap B''$. We restart the requirements having lower priority, given them a restrain to stay in $V \cap B''$.

Once the higher priority requirements have settled, either M never outputs U, so $x \in B'$ which is contained in U, or M eventually outputs U, so $x \in B''$ which is disjoint from U. In each case, M does not enumerate the basic τ_2-neighborhoods of x. □

5.5.3 Application

Definition 5.5.1 is rather cumbersome and one might wonder whether it is really useful in practice. Our recent experience suggests that it is indeed useful. Building directly a τ_1-computabe point that is not τ_2-computable can be difficult in practice, because there are many objects to take care of at the same time. Having an abstract result like Theorem 5.5.2 is very helpful as it isolates the key effective topological properties making the construction possible, and then takes charge of the technical construction. Therefore, it makes the proof easier to find and easier to read. We present an interesting application of this result, for which the analysis developed in this section is illuminating.

Together with Djamel Eddine Amir and as a part of his PhD project, we are studying sets having *computable type*. If X is a topological space, then a **copy** of X in Y is a subspace of Y that is homeomorphic to X. If (X, A) is a pair consisting of a topological space X and a subset $A \subseteq X$, then a copy of (X, A) in Y is a pair (X', A') with $A' \subseteq X' \subseteq Y$ such that X' is homemorphic to X and the homeomorphism sends A' to A.

Definition 5.5.2 ([IS18])**.** If X is a compact Polish space and $A \subseteq X$ is a compact subset, then the pair (X, A) has **computable type** if for every copy of (X, A) in any computable metric space, the following implication holds:

$$X \text{ and } A \text{ are effectively compact} \implies X \text{ is computable.}$$

A compact Polish space X has computable type if the pair $(X, \emptyset)$ has computable type.

J. Miller [Mil02] proved that spheres $\mathbb{S}_n$ have computable type, and balls with their boundaries $(\mathbb{B}_{n+1}, \mathbb{S}_n)$ have computable type. Iljazović and Sušić [IS18] proved more generally that compact manifolds without boundary M, as well as compact manifolds with boundary $(M, \partial M)$, have computabe type. In [AH22], we investigated which pairs of simplicial complexes have computable type and obtained the following result, among others:

Theorem 5.5.3 ([AH22])**.** *Let X be a finite simplicial complex and A a subcomplex having empty interior. The following statements are equivalent:*

1. *The pair (X, A) has computable type,*
2. *There exists $\epsilon > 0$ such that every continuous function $f : X \to X$ satisfying $f_{|A} = \mathrm{id}_A$ and $d(f, \mathrm{id}_X) < \epsilon$ is surjective.*

We implicitly use the construction in the proof of Theorem 5.5.2 to show 1. ⇒ 2., or more accurately, ¬2. ⇒ ¬1. The idea is to work in the computable Polish space $(\mathcal{H}(Q), \tau)$ of homeomorphisms from the Hilbert cube Q to itself, to fix a computable copy of (X, A) in Q and consider two topologies τ_1 and τ_2, related to the upper Vietoris and Vietoris topologies respectively, generated by the following sets:

- For each pair of open sets $U, V \subseteq Q$, the set $\{f \in \mathcal{H}(Q) : f(X) \subseteq U \text{ and } f(A) \subseteq V\}$ is τ_1-open,
- For each open set $U \in Q$, the set $\{f \in \mathcal{H}(Q) : f(X) \cap U \neq \emptyset\}$ is τ_2-open, and each τ_1-open set is τ_2-open.

Assuming $\neg$2., we then build a copy of (X, A) that is τ_1-computable but not τ_2-computable. The construction implicitly relies on the fact that τ_1 is effectively τ-generically weaker than τ_2, which is implied by the existence of non-surjective functions that are arbitrarily close to the identity.

Example 5.5.1 (Dunce hat). As an example we show that the dunce hat D does not have computable type, i.e. it has a copy $X \subseteq Q$ which is effectively compact but not computable. D is obtained from a solid triangle by gluing its three edges, with one orientation reversed (see Figure 5.3). We show that for every $\epsilon > 0$, there exists a non-surjective continuous function $f : D \to D$ that is ϵ-close to the identity and apply Theorem 5.5.3 (note that there is no boundary, i.e. we are dealing with the pair $(D, \emptyset)$).

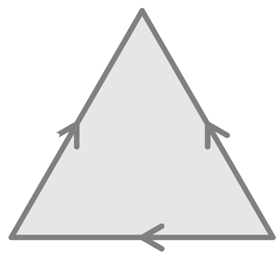


Figure 5.3: The dunce hat is obtained by gluing the edges according to their orientations

We are not aware of any direct construction of an effectively compact copy that is not computable, or any simple visualization of such a copy. It contrasts with the case of the line segment for which it is easy to build a non-computable, effectively compact copy: take $[0, r] \in \mathbb{R}$ where r is a non-computable right-c.e. number. $\triangleleft$

We saw in Example 5.2.3 that a vector space structure has interesting consequences on the comparison between topologies induced by norms. In particular, it makes it easier to prove that a topology is effectively generically weaker than the other, because a discontinuity at one point can be transferred at any other point and at any scale, using the vector space operations.

The analysis of $\mathcal{H}(Q)$ has a similar flavor. Although it is not a vector space, it is a topological (even Polish) group, so everything that happens at a point g can be transferred to any other point h, by applying the continuous action of hg^{-1}. This observation makes it easier to prove that τ_1 is effectively τ-generically weaker than τ_2.

Chapter 6

Genericity of semicomputable objects

Contents

6.1 Introduction

The effectivization of mathematics leads to introduce, for each mathematical notion and each possible formulation of it, an effective version. Usually different notions and different formulations naturally lead to different effective versions. However, proving separation results between these notions can be difficult, involving sophisticated techniques based on diagonalizations.

We are in a situation where the result to be proved is usually expected, because there is no reason why two differently defined notions would coincide, but where the proof can be very technical and very specific to each problem, hence not reusable. We would like to improve the situation by unifying seemingly different arguments and obtain general theorems capturing certain types of constructions, that could be used to prove new results without having to repeat the construction each time. We have two very specific goals:

- Prove general theorems capturing construction techniques,
- Make the assumptions of the theorems easy to check.

Our second goal is very important, because we want the application of the theorem to be much easier than a direct construction. A way to achieve this goal is to find conditions that can be expressed using well-understood concepts, such as topological concepts, rather than *ad hoc* conditions that are designed for the construction to work.

This problem has been addressed in may ways in the literature. A brilliant contribution was made by Jockusch [Joc80] and Kurtz [Kur81] who introduced the notions of 1-generic and weakly-1-generic points. Their results capture many constructions of Δ^0_2 sets in a very elegant and simple way. In particular, the results identify what the properties should look like topologically to make the construction possible: these properties should be dense open sets, in some effective way. Moreover, these notions make sense not only for subsets of $\mathbb{N}$, but for points of any computable Polish space.

Capturing constructions of c.e. sets is more challenging, but was successfully achieved by many people. Lachlan [Lac73] developed a general framework which is powerful, modular and very good at spotting the key topological ingredients of the general construction. Variations on 1-generic sets were proposed by Ingrassia [Ing81], Maass [Maa82], Jockusch [Joc85], Nerode and Remmel [NR85]. Although these approaches all reach the first goal, we feel that they do not reach the second goal as successfully. Applying these results, i.e. showing that the conditions of the theorems are satisfied in a particular situation, does not seem to be much easier than doing the construction directly. It may explain why these approaches are, to our knowledge, not used in practice, despite their great merits.

We introduce another variation on the notion of 1-genericity and show that it captures a certain type of construction, which is the simplest form of the priority method with finite injury. The definitions are topological, which often makes them easy to check in particular situations.

We have been looking for extensions of our result to capture more sophisticated forms of priority arguments, but failed to find conditions that are general and simple. It may not be possible to reach this goal, however we are still optimistic that the impression of familiarity that one feels when performing similar constructions in different situations is a sign that they share common features that need to be isolated and conceptualized.

The results of this chapter are published in [Hoy17, HG17].

6.2 Upper-genericity

Like in the previous chapter, we work on a set endowed a computable Polish topology τ and an effectively weaker countably-based topology τ'. In the previous chapter, we introduced a technique to build a τ'-computable point that is no τ-computable. In this chapter we go further, by building a τ'-computable point which is, to some extent, generic w.r.t. τ, i.e. which satisfies many properties that are typical in the sense of Baire category. One can think of a τ'-computable point as being "semicomputable", in the sense that only partial information can be computed about it, in comparison with τ-computable points for which more information is available. Examples of semicomputable objects are c.e. subsets of $\mathbb{N}$ or left-c.e. reals. We present the definition and main result, and then give a few applications and discuss their limitations.

Let (X, τ) be a topological space.

Definition 6.2.1 (Specialization pre-order)**.** The **specialization pre-order** $\preceq_\tau$ induced by a topology τ is defined by

$$x \preceq_\tau y \text{ if every } \tau\text{-neighborhood of } x \text{ is a } \tau\text{-neighborhood of } y.$$

If the space is T_0 then its specialization pre-order is actually an order, i.e. it is anti-symmetric. If the space is T_1 then its specialization pre-order is trivial: it is the equality relation. So we are interested in topologies that are not T_1.

Let (X, τ) be a Polish space and τ' a topology that is weaker than τ, i.e. $\tau' \subseteq \tau$, as in the previous chapter.

Definition 6.2.2 (Lower boundary)**.** A point $x \in X$ belongs to the **lower boundary** of a τ-open set $U \subseteq X$ if

- $x \notin U$,
- There exists a sequence $x_n \in U$ converging to x in the topology τ, such that $x \preceq_{\tau'} x_n$.

The only difference with the τ-boundary of U is that the sequence x_n is required to lie above x in the specialization pre-order $\preceq_{\tau'}$. Observe that in the trivial case when τ' is T_1, $\preceq_{\tau'}$ is the equality relation, so the lower boundary is always empty.

Main Definition 6.2.1: Upper-generic point

A point $x \in X$ is **upper-generic** if it does not belong to the lower boundary of any τ-effective open set.

This is a weakening of the notion of 1-generic point. Again, in the trivial case when τ' is T_1, every point is upper-generic. The notion of upper-genericity depends on both τ and τ', but we do not mention them explicitly to avoid heavy notations.

We now state the main result of this chapter. It uses one some extra effectiveness assumption about the topologies τ and τ'.

Assumption 1: One can decide whether a finite intersection of basic open sets from both topologies τ and τ' is non-empty.

Main Theorem 6.2.2: Existence of upper-generic computable points

Let (X, τ) be a computable Polish space and τ' an effective countably-based topology that is effectively weaker than τ and satisfying Assumption 1. For any sequence of uniformly effective dense τ-open sets $U_n \subseteq X$, there exists $x \in \bigcap_n U_n$ such that:

- x is τ'-computable,
- x is upper-generic.

The value of this result depends on the topologies. Again, if τ' is T_1 then the result is vacuously true as every point is upper-generic. We will look at interesting examples of non-T_1 topologies.

For each specific topology τ', we might adapt the terminology to the pre-order $\preceq_{\tau'}$. For instance, when $\preceq_{\tau'}$ is the natural ordering on real numbers, upper-generic reals will be called *right-generic*; when $\preceq_{\tau'}$ is the reverse inclusion ordering on closed sets, upper-generic closed sets will be called *inner-generic*; when $\preceq_{\tau'}$ is the inclusion ordering on subsets of $\mathbb{N}$, we keep the name and refer to *upper-generic* sets.

Example 6.2.1 (Friedberg-Muchnik). Friedberg and Muchnik's pair of Turing incomparable c.e. sets can be obtained using Theorem 6.2.2. The proof of this theorem is precisely an application of the technique introduced by Friedberg and Muchnik, so Theorem 6.2.2 should be thought as an abstraction of their technique.

Let $2^{\mathbb{N}}$ be the Cantor space endowed with the usual Polish topology τ. Let τ' be the Scott topology. Theorem 6.2.2 implies the existence of an upper-generic c.e. set $A \subseteq \mathbb{N}$ whose even and odd parts A_0 and A_1 are co-infinite (the latter property is an intersection of uniformly effective dense τ-open sets).

If A is such a set, then A_0 and A_1 are Turing incomparable. Indeed, let M be a Turing machine and

$$V_M = \{B \subseteq \mathbb{N} : \exists n, M^{B_0}(n) = 0 \text{ and } B_1(n) = 1\}.$$

V_M is an effective τ-open set and if A_1 is co-infinite and $M^{A_0} = A_1$, then A belongs to the lower-boundary of V_M, contradicting the upper-genericity of A. Symmetrically, $M^{A_1} = A_0$ is not possible. $\triangleleft$

6.3 Genericity among the left-c.e. reals

Left-c.e. real numbers play an important role in computability theory, computable analysis and algorithmic randomness. Their rich structure has been thoroughly investigated in many ways. In this section we study some of the properties shared by the "generic" left-c.e. reals, which can be made precise using Definition 6.2.1.

Let X be the unit interval $[0,1]$ with the Euclidean topology τ which is Polish. Let τ' be the so-called lower topology generated by the semi-lines $(x, 1]$. The τ'-computable reals are the left-c.e. reals. The specialization order $\preceq_{\tau'}$ is the natural ordering $\leq$ of real numbers. Let us reformulate the definition of upper-generic elements in this particular case.

Main Definition 6.3.1: Right-generic real

A real number $x \in [0,1)$ is **right-generic** if for every effective Euclidean open set $U \subseteq [0,1]$, either $x \in U$ or there exists $\epsilon > 0$ such that $[x, x+\epsilon] \cap U = \emptyset$.

Let us give a first illustration of the properties of right-generic reals. For every real number x, the binary expansion of x can be computed from any Cauchy name of x, i.e. any sequence of rational numbers $(q_n)_{n\in\mathbb{N}}$ satisfying $|q_n - x| \leq 2^{-n}$. However, computing the nth bit may require to know x within arbitrarily high precision compared to n. We show that for right-generic reals, this precision cannot be bounded by a computable function.

Let us give more precise definitions in order to state the result. Let $\mu : \mathbb{N} \to \mathbb{N}$. Say that a Turing machine M computes the binary expansion of x from Cauchy names of x, with **modulus** μ, if for every $n \in \mathbb{N}$ and every rational q such that $|q - x| < 2^{-\mu(n)}$, $M(n, q)$ outputs the nth bit of the binary expansion of x.

Proposition 6.3.1. *Let $x \in [0,1)$ be right-generic. There is no Turing machine computing binary expansion of x from Cauchy names of x with a computable modulus.*

Proof. Let $\mu : \mathbb{N} \to \mathbb{N}$ be a computable function and M a Turing machine computing the binary expansion of x with modulus μ. Let U be the set of numbers y on which the computation is incorrect:

$$U = \{y \in [0,1] : \exists q, n, |q - y| < 2^{-\mu(n)} \text{ and } M(q,n) = 0 \\ \text{but the } n\text{th bit of } y \text{ is } 1\}.$$

The set U is an effective Euclidean open set that does not contain x. As x is right-generic, there exists $\epsilon > 0$ such that $[x, x+\epsilon)$ is disjoint from U.

We are going to find a contradiction by observing that if x is right-generic, then its binary expansion contains arbitrarily large chunks of 1s at early positions. Indeed, for $p \in \mathbb{N}$, let

$$V_p = \{y \in [0,1] : \exists n \geq p, y[n : \mu(n)] = 011\dots1\},$$

where $y[n : \mu(n)]$ denotes the part of the binary expansion of y between position n and $\mu(n)$. Each V_p is a dense effective Euclidean open set, so it contains all the right-generic reals.

Let n be arbitrarily large and such that $x[n : \mu(n)] = 011\dots1$. Let y be obtained by replacing $x[n : \mu(n)]$ by $100\dots0$. Let q be such that $x < q < x + 2^{-\mu(n)}$. One easily checks that $y \in U$, witnessed by q and n. However, y lies on the right of x and is arbitrarily close to x, so x belongs to the down-boundary of U, contradicting the right-genericity of x. □

It happens that this notion of genericity for left-c.e. reals is related to other classes of left-c.e. real numbers: the **weakly-1-generic** reals and the **simple** reals (not to be confused with Post's simple sets of natural numbers). These three classes are contained in each other. We present separation results.

The important feature of these results is that their proofs use notions of genericity for suitable topologies in order to build left-c.e. reals separating the classes, instead of building them by hand. Therefore, in order to build an object, one can only focus on analyzing individually the properties that this object should satisfy and not on the construction itself, which is usually cumbersome and captured once for all by Theorem 6.2.2. Again, a computability-theoretic construction relies on topology.

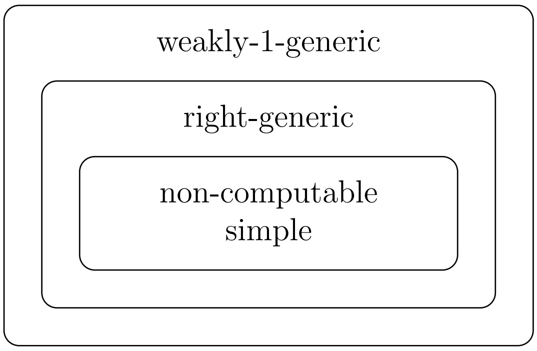


Figure 6.1: Comparison between 3 classes of reals

6.3.1 Separation between weakly-1-generic and right-generic reals

This result appears in [Hoy17].

A real number is **weakly-1-generic** if it belongs to every effective dense open set. One easily has the following chain of implications:

$$\text{1-generic} \implies \text{right-generic} \implies \text{weakly-1-generic}$$

In order to build a left-c.e. weakly-1-generic real x that is not right-generic, we build a Π^0_1-set C such that:

- $x = \min C$, so that x is left-c.e.,
- The complement of C is dense on the right of x, so that x is not right-generic,

- The boundary of C contains only weakly-1-generic reals, so that x is weakly-1-generic.

We achieve the second condition by making the boundary of C perfect, i.e. without isolated point.

In the Vietoris topology on the space of closed subsets of $[0,1]$, typical elements have empty interior, which is incompatible with condition 3. We need to choose another topology, in which every typical element is regular, i.e. is the closure of its interior. The topology τ_{int} is generated by ($a < b$ are rational numbers):

- The *hit sets* $\{C \subseteq [0,1] : (a,b) \cap C \neq \emptyset\}$,
- The *miss sets* $\{C \subseteq [0,1] : [a,b] \cap C = \emptyset\}$,
- The sets $\{C \subseteq [0,1] : [a,b] \subseteq \text{int}(C)\}$.

Describing C in the topology τ_{int} amounts to describing C in the Vietoris topology together with enumerating its interior. Note that using this description, one can compute the boundary of C in the Vietoris topology.

In this computable Polish topology, the class of regular closed sets with a perfect boundary is a dense effective G_δ-subset of X. The topology τ' is generated by the miss sets, its computable elements are the Π^0_1-sets. Theorem 6.2.2 implies the existence of a regular Π^0_1-set C whose boundary is perfect and that is inner-generic in the topology τ_{int}.

Theorem 6.3.2. *Let C be an effectively closed subset of $[0,1]$. Assume that*

- *C is regular,*
- *The boundary of C is perfect,*
- *C is inner-generic in the topology τ_{int}.*

Then the number $\min C$ is left-c.e., weakly-1-generic but not right-generic.

Proof sketch. We first show that the boundary of C contains only weakly-1-generic reals. Let $U \subseteq [0,1]$ be a dense effective open set. The class $\mathcal{U}$ of closed sets whose boundary is contained in U is an effective open subset of (X,τ). For every neighborhood $\mathcal{N}$ of C in (X,τ), there exists a subset C' of C in $\mathcal{N} \cap \mathcal{U}$. As C is inner-generic, C belongs to $\mathcal{U}$.

The number $x = \min C$ is not right-generic because the complement of C is an effective open set that is dense on the right of x (otherwise x would be isolated in the boundary of C). □

6.3.2 Separation between simple and right-generic reals

The results of this section appear in [HG17].

Downey and LaForte [DL02] introduced the notion of simple reals, and proved that non-computable left-c.e. simple reals exist.

Definition 6.3.1 (Simple real)**.** A left-c.e. real number is **simple** if for every prefix-free c.e. set of strings $E \subseteq \{0,1\}^*$ such that $x = \sum_{\sigma \in E} 2^{-|\sigma|}$, E is computable.

Right-generic reals are related to simple reals as follows.

Proposition 6.3.2. *Every non-computable simple left-c.e. real is right-generic.*

Proof. Let x be non-computable left-c.e. and simple. Let $U \subseteq [0,1]$ be an effective open set that does not contain x. The set $V = [0,x) \cup U$ is effectively open, so there exists a prefix-free c.e. set $E \subseteq \{0,1\}^*$ such that $V = \bigcup_{\sigma\in E}[\sigma]$. Let $E_< = \{\sigma \in E : \sigma \leq_{lex} x\}$, where $\leq_{lex}$ is the lexicographic ordering. $E_<$ is a prefix-free c.e. set and $x = \sum_{\sigma\in E_x} 2^{-|\sigma|}$. As x is simple, $E_<$ is computable, so the set $E_> := E \setminus E_<$ is c.e. As a result, $U \setminus [0,x) = \bigcup_{\sigma\in E_>}[\sigma]$ is an effective open set, so its infimum is right-c.e. As x is not computable, the infimum differs from x, so there exists $\epsilon > 0$ such that $[x, x+\epsilon)$ is disjoint from U. □

We now show that the implication is strict. Instead of building an example by hand, we let the genericity framework do it for us. We start with the observation that if $A \subseteq \mathbb{N}$ is a non-computable c.e. set, then the real number

$$x_A := \sum_{n\in A} \frac{1}{n^2}$$

is left-c.e. and is not simple. We show that if A is c.e. and upper-generic for a suitable topology, then x_A is right-generic. To achieve this, we need to use a topology that makes typical sets sparse in the following sense.

Definition 6.3.2 (Sparse set)**.** A set $A \subseteq \mathbb{N}$ is **sparse** if for every n, almost every element of A is a multiple of 2^n.

In the Cantor topology, typical sets are not sparse, where "typical" is understood in the sense of Baire category. We consider another topology τ_{sparse} on the power set of $\mathbb{N}$, in which typical sets are sparse. The topology τ_{sparse} is generated by the cylinders from the Cantor topology together with the sets

$$\{A \subseteq \mathbb{N} : \text{every element of } A \text{ larger than } k \text{ is a multiple of } 2^n\}.$$

In the topology τ_{sparse}, the class of sparse sets is a dense effective G_δ-sets. We let τ' be the Scott topology, whose computable elements are the c.e. sets. We can therefore apply Theorem 6.2.2, which implies the existence of sparse c.e. sets that are upper-generic in the topology τ_{sparse}.

Theorem 6.3.3 (Right-generic but not simple)**.** *Let $A \subseteq \mathbb{N}$ be a sparse c.e. set that is upper-generic in the topology τ_{sparse}. The real $x_A = \sum_{n\in A} 1/n^2$ is right-generic.*

Proof sketch. The important property of any sparse set A is that the set of real numbers x_B, where B is a superset of A, contains an interval $[x_A, x_A+\epsilon]$ for some $\epsilon > 0$, which is a consequence of the results of Graham [Gra64]. If some effective open set $U \subseteq [0,1]$ is dense on the right of x_A then the pre-image $\mathcal{U} = \{B : x_B \in U\}$ is dense above A. As A is upper-generic, A belongs to $\mathcal{U}$ so x_A belongs to U. □

The sequence $1/n^2$ appearing in Theorem 6.3.3 can be replaced by any sequence that is sufficiently effective and does not converge too quickly to 0.

Theorem 6.3.4. *Let $(u_n)_{n\in\mathbb{N}}$ be a computable sequence of positive reals with a computable sum, such that u_{n+1}/u_n converges to 1. If $A \subseteq \mathbb{N}$ is a sparse c.e. set that is upper-generic in the topology τ_{sparse}, then $\sum_{n\in A} u_n$ is right-generic (and not simple).*

6.4 Other applications

We have been using this framework to easily build examples in other situations. We have studied the properties of Π^0_1 triangles or disks, trying to understand how the property of being Π^0_1 is reflected in their parameters such as their coordinates. In this investigation, a fruitful approach is to study the property of "generic" Π^0_1 triangles or disks, using the notion of genericity presented in this chapter. These results have been obtained with a M2 intern Diego Nava Saucedo and completed with a postdoctoral fellow Donald M. Stull, and published in [HSS18].

We have also investigated an extension of the notion of left-c.e. real to points of Euclidean spaces, by considering the set of directions along which the projections of a point are left-c.e. We used it to understand for which coefficients a, b, c the quadratic polynomial $a + bX + cX^2$ is a left-c.e. function of X. This work has applications to an important notion of reducibility between real numbers, called Solovay reducibility. This work was done in collaboration with Donald M. Stull during his postdoctoral fellowship and was published in [HS19].

In these studies, genericity helps understanding the limit cases without having to build counter-examples by hand.

6.5 Baire category on the left-c.e. reals

We briefly present an attempt to capture more powerful versions of the priority method with finite injury, using an effective version of Baire category. These results are not published.

Definition 6.5.1. Say that an open set $U \subseteq [0, 1]$ is **left-c.e. dense** if there is an algorithm that given any rational interval (a, b), produces two left-c.e. numbers $c < d$ such that $(c, d) \subseteq (a, b) \cap U$.

Definition 6.5.2. A set $A \subseteq [0, 1]$ is **large** if it contains the intersection of an effective sequence of left-c.e. dense open sets. A set is **small** if its complement is large.

Theorem 6.5.1 (Baire category on the left-c.e. reals)**.** *Every large set contains a left-c.e. real.*

In other words, the set of left-c.e. real numbers is not small. The proof uses the priority method with finite injury. Contrary to the proof of Theorem 6.2.2, each requirement may act several times (in addition to the injuries).

- For each left-c.e. real x, the set $\{x\}$ is small. However, it cannot be uniform in x, otherwise the whole set of left-c.e. reals would be small, contradicting Theorem 6.5.1. The argument is different whether x is computable or not.
- The set of right-c.e. real numbers if small.
- The set of sums $\sum_{n\in E} \frac{1}{n^2}$ for E ranging over the c.e. sets is small. It shows that Theorem 6.5.1 captures some constructions that Theorem 6.2.2 does not (see Theorem 6.3.3).

However, we do not know whether this approach is more powerful than right-genericity. This problem can be formulated as follows:

Open Question 6.5.2: Right-genericity vs Baire category

Is the set of right-generic left-c.e. reals large, in the sense of Definition 6.5.2?

Moreover, this approach seems to be very specific to the left-c.e. real numbers and cannot be extended to c.e. sets.

Chapter 7

Descriptive complexity on admissibly represented spaces

Contents

7.1 Introduction

The topological spaces having an admissible representation are precisely the represented spaces on which topology and computability are closely related: between such spaces, the continuous functions are precisely the functions that are computable relative to some oracle. Those spaces have been characterized by Schröder as the T_0 quotients of countably-based spaces, and most natural topological spaces fall into this category.

One is tempted to think that on these spaces, topology and relative computability are the two faces of the same coin. However we show in this chapter that it is not quite true as they provide non-equivalent notions of descriptive complexity. Descriptive complexity classes, traditionally studied on Polish spaces, can be defined on an arbitrary topological space, and measure the complexity of describing a subset as a countable boolean combination of open sets. On a represented space, it is possible to follow the alternative approach of transferring descriptive complexity classes from the Baire space to the represented space, by defining the complexity of a set as the complexity of its preimage under the representation.

These two measures of descriptive complexity, that we respectively call *topological* and *symbolic*, turn out to be equivalent on countably-based spaces, as shown by de Brecht and Yamamoto [dBY09, dB13], based on previous results by Saint Raymond [Ray07]. First results in this direction were previously proved by Brattka [Bra05] for metric spaces and Selivanov [Sel08] for ω-continuous domains. However the relationship between symbolic and topological complexity on other admissibly represented spaces has not been previously studied, although symbolic complexity was considered by de Brecht and Pauly in [PdB15]. We carried out this investigation in various classes of spaces, identifying the extent to which they differ and trying to understand why they differ.

Our results suggest that relative computability is more closely related to the sequential aspects of the space than to its topological aspects. It is already known from the theory of represented spaces that representations are closely related to sequential spaces, through the following results due to Schröder [Sch02a]:

- Admissibly represented spaces are sequential[3],
- The topology induced by the product representation of two spaces is not the product topology but its sequentialization,
- The topology induced by the subspace representation of a space is not the subspace topology but its sequentialization.

Our results show that the mismatch between topological and sequential constructs is at the origin of the mismatch between symbolic and topological complexity:

- Among coPolish spaces, the spaces on which symbolic and topological complexity agree at the simplest level are precisely the Fréchet-Urysohn spaces, i.e. the spaces on which sequential and topological closures agree,
- Our examples of sets whose symbolic complexity is strictly below their topological complexity are built from non-sequential subspaces and non-sequential products.

This work has started during the internship of Antonin Callard in 2019 and was published in [CH20, Hoy20b]. Antonin Callard was awarded the VCLA Outstanding Undergraduate Thesis Award 2020 for his work.

Let us cite a few articles which are somehow related to the problem. Pauly and de Brecht [PdB15] give an example of a represented space which is not admissible because its topology is trivial, on which symbolic complexity is non-trivial (Theorem 37 and the next paragraph in the article). Grassin [Gra74] and Selivanov [Sel84] investigated the relationship between *index* complexity and topological complexity: they worked on numbered sets rather than represented spaces, so the analog of the symbolic complexity of a set is the complexity of its index set; in that setting, only effective complexity classes are considered.

7.2 Descriptive complexity

A **descriptive complexity class** Γ is a family $\{\Gamma(X) : X \text{ topological space}\}$, where each $\Gamma(X)$ is a class of subsets of the topological space X. We will mainly consider the classes of the Borel hierarchy and the Hausdorff difference hierarchy, as well as their effective versions. For simplicity, we only consider the finite levels of these hierarchies, although some of the results can be extended to the whole hierarchies.

[3] Admissibility is sometimes formulated in a weaker form allowing non-sequential topologies. However the two notions of admissibility are closedly related: every weakly admissibly represented space can be turned into a strongly admissible one by keeping the same representation but taking a stronger topology, namely the sequentialization of the original topology, which also coincides with the final topology of the representation. The two topologies do not induce the same continuous functions, but the same sequentially continuous ones, which are exactly the functions that are computable relative to some oracle.

The location of a set in those hierarchies measures the complexity of describing the set as a boolean combination of open sets. Although the definitions apply to any topological space, they are mostly studied on countably-based spaces and in particular on Polish or quasiPolish spaces, on which structural theorems can be proved. It is not clear whether these hierarchies are really interesting on topological spaces which are not countably-based. The results we present in this chapter suggest that they do not behave very well in general and that another approach to descriptive complexity is needed. Indeed, representations induce notions of descriptive complexity, which we call *symbolic complexity* as opposed to *topological complexity*, which often behave in a better way.

Let X be a topological space. In what follows, m, n are natural numbers.

The Borel hierarchy.

- Let $\mathbf{\Sigma}^0_1(X)$ be the class of open subsets of X,
- Let $\mathbf{\Sigma}^0_{n+1}(X)$ be the class of countable unions of differences of sets in $\mathbf{\Sigma}^0_n(X)$,
- Let $\mathbf{\Pi}^0_n(X)$ be the class of complements of sets in $\mathbf{\Sigma}^0_n(X)$,
- Let $\mathbf{\Delta}^0_n(X) = \mathbf{\Sigma}^0_n(X) \cap \mathbf{\Pi}^0_n(X)$.

The hierarchy is actually defined at all levels indexed by countable ordinals, and contains all the Borel sets.

The Hausdorff difference hierarchy.

- Let $\mathbf{D}_1(X) = \mathbf{\Sigma}^0_1(X)$ is the class of open sets,
- Let $\mathbf{D}_{n+1}(X)$ be the class of sets $U \setminus A$ where U is open and $A \in \mathbf{D}_n(X)$,
- Let $\mathbf{D}_\omega(X)$ be the class of sets $\bigcup_{n<\omega} A_{2n+1} \setminus A_{2n}$, where $(A_n)_{n<\omega}$ is any growing family of open sets.

The hierarchy is actually defined at all levels indexed by countable ordinals, and analogously defined starting from $\mathbf{\Sigma}^0_\beta(X)$ in place of $\mathbf{\Sigma}^0_1(X)$. For simplicity, we restrict our attention to the first levels of this hierarchy, although our articles deal with the general case.

Each complexity class comes with its effective version, written in lightface (for instance $\mathbf{\Sigma}^0_2$ becomes Σ^0_2) and defined similarly but involving computable sequences of effective open sets.

Important results of Descriptive Set Theory show that on Polish spaces,

- These hierarchies are proper,
- Each level $\mathbf{\Sigma}^0_\alpha(X)$ or $\mathbf{D}_\alpha(X)$ has a complete set,
- The extension of the Hausdorff difference hierarchy $(\mathbf{D}_\alpha(X))_{\alpha<\omega_1}$ contains exactly the $\mathbf{\Delta}^0_2(X)$-sets (Hausdorff-Kuratowski theorem).

More details can be found in [Kec95]. These results have been extended to the class of quasiPolish spaces by de Brecht [dB13].

Symbolic complexity. Representations are used to naturally transfer notions from the Baire space $\mathcal{N}$ to other spaces, like computability of points and functions. In the same way, they induce notions of descriptive complexity, measured in the Baire space rather than in the target space.

Main Definition 7.2.1: Symbolic complexity

Let (X, δ_X) be a represented space, where $\delta_X : \mathcal{N} \to X$ is total. If Γ is a descriptive complexity class, then the corresponding **symbolic complexity class**, denoted by $[\Gamma](X)$, consists of the sets $A \subseteq X$ such that $\delta_X^{-1}(A) \in \Gamma(\mathcal{N})$.

We assume that δ_X is total for simplicity, but the results apply to the more general case, requiring $\delta_X^{-1}(A) \in \Gamma(\mathrm{dom}(\delta_X))$. Note that from the point of view of a Turing machine, the symbolic complexity of a set is more relevant than its topological complexity because the machine works with names of points rather than the points themselves.

We will only work with admissible representations of topological spaces. Usual complexity classes Γ are closed under taking continuous preimages, and in that case, one always has

$$\Gamma(X) \subseteq [\Gamma](X) \tag{7.1}$$

because δ_X is continuous. The general goal of this chapter is to investigate for which X and which Γ the inclusion (7.1) is an equality.

7.3 Countably-based spaces

Yamamoto and de Brecht [dBY09, dB13] proved that in a countably-based T_0-space with its standard representation, one has $\Gamma(X) = [\Gamma](X)$ for each Γ in the Borel or Hausdorff hierarchies.

$$\begin{array}{ccc}
\vdots & & \vdots \\
[\boldsymbol{\Sigma}^0_3] & = & \boldsymbol{\Sigma}^0_3 \\
[\boldsymbol{\Sigma}^0_2] & = & \boldsymbol{\Sigma}^0_2 \\
[\boldsymbol{\Delta}^0_2] & = & \boldsymbol{\Delta}^0_2 \\
[\mathbf{D}_n] & = & \mathbf{D}_n \\
[\boldsymbol{\Sigma}^0_1] & = & \boldsymbol{\Sigma}^0_1
\end{array}$$

Figure 7.1: Symbolic and topological complexity agree on countably-based spaces

During Callard's internship, we observed that the proof is effective, giving an effective version of this result [CH20].

Theorem 7.3.1. *Let X be an effective countably-based T_0-space with its standard representation. For $\Gamma \in \{\boldsymbol{\Sigma}^0_n, \mathbf{D}_n(\boldsymbol{\Sigma}^0_m)\}$, the equality $[\Gamma](X) = \Gamma(X)$ is uniformly computable.*

It means that there is a uniformly computable translation between Γ-descriptions and $[\Gamma]$-descriptions. For instance, if $\Gamma = \boldsymbol{\Sigma}^0_2$, then

- A $\mathbf{\Sigma}^0_2$-description of a set $A \subseteq X$ is given by two sequences of open sets $A_i, B_i \subseteq X$ such that $A = \bigcup_i A_i \setminus B_i$, where A_i, B_i are described by giving the open sets $\delta_X^{-1}(A_i)$ and $\delta_X^{-1}(B_i)$,
- A $[\mathbf{\Sigma}^0_2]$-description of a set $A \subseteq X$ is given by two sequences of open sets $A_i, B_i \subseteq \mathcal{N}$ such that $\delta_X^{-1}(A) = \bigcup_i A_i \setminus B_i$.

In particular, the result implies that for effective classes $\Gamma \in \{\Sigma^0_n, \mathrm{D}_n(\Sigma^0_m)\}$, one has $\Gamma(X) = [\Gamma](X)$. We will see spaces X where symbolic and topological complexity agree but not their effective versions, more precisely $[\mathbf{D}_n](X) = \mathbf{D}_n(X)$ but $[\mathrm{D}_n](X) \subsetneq \mathrm{D}_n(X)$ (Theorems 7.5.1 and 7.5.2).

We show that the preceding theorem is optimal in the sense that it holds for countably-based spaces *only* [CH20].

Theorem 7.3.2. *Let (X, δ_X) be an admissibly represented space. The following statements are equivalent:*

- *The equality $[\mathbf{D}_2(X)] = \mathbf{D}_2(X)$ holds and is uniformly computable, relative to some oracle,*
- *X is countably-based.*

Example 7.3.1. There is a space X which is not countably-based but for which the equality holds $[\mathbf{D}_2(X)] = \mathbf{D}_2(X)$ (although non-uniformly). This space is the quotient $\mathbb{R}/\mathbb{Z}$ where all the integers are glued together (see Figure 7.2). It is coPolish (see next section) and therefore has an admissible representation as follows: a name for a point x is any $n \in \mathbb{N}$ such that x belongs to the central $2n+1$ petals and a name of x as a point of the metric space consiting of those petals. The previous result show that there is no uniformly computable way of translating $[\mathbf{D}_2(X)]$-descriptions into $\mathbf{D}_2(X)$-descriptions (even relative to any oracle). However, there is a uniformly computable translation taking an extra bit of information indicating whether the special point (i.e., the equivalence class of $\mathbb{Z}$) belongs to the set or not. In particular, the effective classes coincide: $[\mathrm{D}_2](X) = \mathrm{D}_2(X)$.

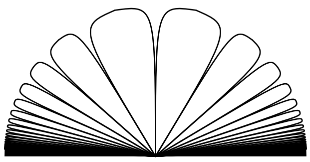


Figure 7.2: The coPolish space $\mathbb{R}/\mathbb{Z}$

◁

7.4 CoPolish spaces

CoPolish spaces are a particularly interesting class of spaces, studied by Schröder in [Sch04] mainly because they admit a simple complexity theory. They are one of the simplest classes of spaces which are not always countably-based.

Definition 7.4.1 (CoPolish space)**.** A **coPolish space** is the inductive limit of a growing sequence of locally compact metrizable spaces.

More precisely, let $(X_n)_{n\in\mathbb{N}}$ be a sequence of locally compact metrizable spaces with $X_n \subseteq X_{n+1}$. They induce a coPolish space $X = \bigcup_n X_n$ where a set $U \subseteq X$ is open if and only if each $U \cap X_n$ is open in X_n.

CoPolish spaces have a natural admissible representation: a name of $x \in X$ is any pair (n, p) where $n \in \mathbb{N}$ is such that $x \in X_n$ and $p \in \mathcal{N}$ is any name of x as an element of X_n (being countably-based, X_n has an admissible representation). The original definition assumes that the spaces X_n are compact, but it is not difficult to see that the two definitions are equivalent.

Example 7.4.1 (Polynomials). The space $\mathbb{R}[X]$ of polynomials with real coefficients can be endowed with a coPolish topology. It is the inductive limit of the subspaces $X_n \cong \mathbb{R}^{n+1}$ of polynomials of degree at most n with the Euclidean topology. A name of a polynomial consists in an upper bound n on its degree and names of its coefficients $p_0, \dots, p_n$. $\triangleleft$

Because of the particular properties of coPolish spaces, symbolic and topological complexity coincide after some level.

Proposition 7.4.1. *If X is coPolish and $n \geq 2$, then $[\utilde{\mathbf{\Sigma}}^0_n](X) = \utilde{\mathbf{\Sigma}}^0_n(X)$.*

However symbolic and topological complexity can disagree below $\utilde{\mathbf{\Delta}}^0_2$ and we pinpoint the precise disagreement. We already saw in Part I, Section 2.3 an example of a set A that belongs to $[\utilde{\mathbf{D}}_2](X)$ but not to $\utilde{\mathbf{D}}_2(X)$. By the previous proposition, one has $[\utilde{\mathbf{D}}_2](X) \subseteq \utilde{\mathbf{\Delta}}^0_2(X)$. With Callard, we showed that this gap is optimal in general [CH20].

Main Theorem 7.4.1: Symbolic vs topological complexity

In the space X of polynomials, there exists a set $A \in [\utilde{\mathbf{D}}_2](X)$ that is not in any level below $\utilde{\mathbf{\Delta}}^0_2(X)$:

$$A = \left\{ \frac{1}{k_1} + \frac{X^{k_1}}{k_2} + \frac{X^{k_2}}{k_3} + \dots + \frac{X^{k_{n-2}}}{k_{n-1}} + \frac{X^{k_{n-1}}}{k_n} : k_1 < k_2 < \dots < k_n \text{ and } n \text{ even} \right\}.$$

We summarize the comparison between symbolic and topological complexity on coPolish spaces in Figure 7.3. It is then natural to ask why symbolic and topological disagree and

$$\begin{tikzcd}
{[\utilde{\mathbf{\Sigma}}^0_\alpha]} \arrow[r, equal] & \utilde{\mathbf{\Sigma}}^0_\alpha \\
\vdots & \vdots \\
{[\utilde{\mathbf{\Sigma}}^0_2]} \arrow[r, equal] & \utilde{\mathbf{\Sigma}}^0_2 \\
{[\utilde{\mathbf{\Delta}}^0_2]} \arrow[r, equal] & \utilde{\mathbf{\Delta}}^0_2 \\
\vdots & \vdots \\
{[\utilde{\mathbf{D}}_2]} \arrow[uur] & \utilde{\mathbf{D}}_2 \arrow[l] \\
{[\utilde{\mathbf{\Sigma}}^0_1]} \arrow[r, equal] & \utilde{\mathbf{\Sigma}}^0_1
\end{tikzcd}$$

Figure 7.3: Symbolic vs topological complexity on coPolish spaces
($A \to B$ means $A \subseteq B$, with no equality in general)

what symbolic complexity actually measures. The next result from [CH20] shows that the disagreement is a manifestation of the fact that sequential and topological notions are not always equivalent outside countably-based spaces.

A topological space is **Fréchet-Urysohn** if the topological closure of any subset coincides with its sequential closure, which is the set of limits of sequences of points in the set. Equivalently, a space is Fréchet-Urysohn if and only if every topological subspace is sequential.

Theorem 7.4.2. *Let X be a coPolish space. The following statements are equivalent:*

- *X is Fréchet-Urysohn,*
- $\underset{\sim}{\mathbf{D}}_2(X) = [\underset{\sim}{\mathbf{D}}_2](X)$,
- *For every $n < \omega$, $\underset{\sim}{\mathbf{D}}_n(X) = [\underset{\sim}{\mathbf{D}}_n](X)$.*

Example 7.4.2. This result cannot be extended to the next level ω. We again consider the space $X = \mathbb{R}/\mathbb{Z}$ from Example 7.3.1 which is coPolish and Fréchet-Urysohn. One can show that $[\underset{\sim}{\mathbf{D}}_\omega](X) \nsubseteq \underset{\sim}{\mathbf{D}}_\omega(X)$ by carefully choosing for each n a set of complexity $\underset{\sim}{\mathbf{D}}_n$ on the nth petal and taking their union (see Proposition 5.6 in [CH20]). ◁

We have partially extended Theorem 7.4.2 to Hausdorff spaces in [Hoy20b].

Theorem 7.4.3. *Let X be a Hausdorff admissibly represented space. If X is not Fréchet-Urysohn, then $[\underset{\sim}{\mathbf{D}}_2](X) \nsubseteq \underset{\sim}{\mathbf{D}}_2(X)$.*

Informal proof. We use the Arens' space S_2, which is the following subspace of $\mathbb{R}[X]$:

$$\mathrm{S}_2 = \{0\} \cup \left\{\frac{1}{n} : n \in \mathbb{N}\right\} \cup \left\{\frac{1}{n} + \frac{1}{k}X^n : n, k \in \mathbb{N}\right\}.$$

One has $[\underset{\sim}{\mathbf{D}}_2](\mathrm{S}_2) \nsubseteq \underset{\sim}{\mathbf{D}}_2(\mathrm{S}_2)$ and a witness is the set $A = \{0\} \cup \{\frac{1}{n} + \frac{1}{k}X^n : n, k \in \mathbb{N}\}$, which is the example given in 2.3. One can prove that X is not Fréchet-Urysohn if and only if X contains a closed copy of S_2. In that case, A embeds as a subset of X and $A \in [\underset{\sim}{\mathbf{D}}_2](X)$ but $A \notin \underset{\sim}{\mathbf{D}}_2(X)$. □

We will see below (Theorem 7.5.1) that the Hausdorff condition cannot be dropped. Whether Theorem 7.4.3 can be turned into an equivalence is an open problem.

Open Question 7.4.4: Symbolic vs topological complexity in Hausdorff spaces

Let X be a Hausdorff admissibly represented space. If X is Fréchet-Urysohn, does equality $[\underset{\sim}{\mathbf{D}}_2](X) = \underset{\sim}{\mathbf{D}}_2(X)$ hold?

To summarize this section, symbolic and topological complexity already disagree in ones of the simplest spaces which are not countably-based. Moreover, the key idea in the separation results is to exploit the fact that sequential and topological closures may not coincide, or said differently, that the topological subspaces of a sequential space are not always sequential.

7.5 Spaces of open sets

Another natural class of spaces is obtained by starting from a represented space (X, δ_X) and building the space $\mathcal{O}(X)$ of open subsets of X for the final topology of δ_X, which has a canonical admissible representation. A beautiful theorem of Schröder states that the representation of $\mathcal{O}(X)$ induces the Scott topology induced by the inclusion ordering and is admissible (even when the representation of X is not!). This result is central in showing that every T_0 quotient of countably-based space has an admissible representation:

- As a quotient of a countably-based space, X has a canonical representation, which is not admissible in general,
- X can then be embedded in $\mathcal{O}(\mathcal{O}(X))$ by identifying a point $x \in X$ with the set of its open neighborhoods,
- The restriction of the admissible representation of $\mathcal{O}(\mathcal{O}(X))$ is an admissible representation of X.

The order structure on $(\mathcal{O}(X), \subseteq)$ has the interesting consequence that symbolic and topological complexity agree at the first levels. In particular, the next result from [Hoy20b] shows that Theorem 7.4.3 cannot be extended to general non-Hausdorff spaces.

Theorem 7.5.1. *Let X be admissibly represented. For every $n < \omega$, one has*

$$[\mathbf{D}_n](\mathcal{O}(X)) = \mathbf{D}_n(\mathcal{O}(X)).$$

Informal proof. The argument is similar to the one given by Grassin [Gra74] in the setting of numbered sets. There is a simple characterization of $\mathbf{D}_n(\mathcal{O}(X))$ which is analogous to the definition of Scott open set: $A \in \mathbf{D}_n(\mathcal{O}(X))$ if and only if

- There is no $n+1$-chain, i.e. no sequence $U_0 \subseteq U_1 \subseteq \ldots \subseteq U_n$ of open sets such that $U_i \in A$ iff i is even,
- A is approximable, i.e. for every directed set $D \subseteq \mathcal{O}(X)$ such that $\sup D \in A$, D intersects A.

It is not difficult to show that if A has an $n+1$-chain, then A is hard for the class $\check{\mathbf{D}}_n$ (which is the class of complements of sets in $\mathbf{D}_n$), and if A is not approximable then A is $\mathbf{\Pi}^0_2$-hard.

The key is then to observe that hardness is related to *symbolic* complexity: if $A \in [\mathbf{D}_n](\mathcal{O}(X))$ then A is not $\check{\mathbf{D}}_n$-hard (and even less $\mathbf{\Pi}^0_2$-hard). Therefore, A has no $n+1$-chain and is approximable, so $A \in \mathbf{D}_n(\mathcal{O}(X))$. □

$$\begin{array}{ccc}
[\mathbf{D}_\omega] & \longleftarrow & \mathbf{D}_\omega \\
\vdots & & \vdots \\
[\mathbf{D}_n] & = & \mathbf{D}_n \\
\vdots & & \vdots \\
[\mathbf{D}_2] & = & \mathbf{D}_2 \\
[\mathbf{\Sigma}^0_1] & = & \mathbf{\Sigma}^0_1
\end{array}$$

Figure 7.4: Symbolic vs topological complexity on $\mathcal{O}(X)$
($A \to B$ means $A \subseteq B$, with no equality in general)

This result cannot be extended to level $\mathbf{D}_\omega$ in general. We will see that symbolic and topological complexity heavily disagree at that level when X is a non-locally compact Polish space (Theorem 7.5.4). The comparison between symbolic and topological complexity at the first levels is summarized in Figure 7.4.

Effective classes. The proof of Theorem 7.5.1 is not constructive and we show that the corresponding effective classes sometimes disagree. The class D_2 consists of differences of two *effective* open sets. Let $\mathcal{N}_1$ be the space of functions $\mathbb{N} \to \mathbb{N}$ having at most 1 non-zero value. It is the minimal Polish space that is not locally compact (when X is locally compact, $\mathcal{O}(X)$ is countably-based so symbolic and topological complexity agree on $\mathcal{O}(X)$ in a computable uniform way).

Theorem 7.5.2. *One has* $[\mathrm{D}_2](\mathcal{O}(\mathcal{N}_1)) \nsubseteq \mathrm{D}_2(\mathcal{O}(\mathcal{N}_1))$.

So there exists a set $A \in [\mathrm{D}_2](\mathcal{O}(\mathcal{N}_1))$ which by Theorem 7.5.1 is a difference of two open sets, but not a difference of two *effective* open sets.

7.5.1 Spaces of open subsets of Polish spaces

We now focus on spaces $\mathcal{O}(X)$ where X is a Polish space with an admissible total representation. We can establish a rather precise picture of the relationship between symbolic and topological complexity, depending on the compactness properties of the space X.

The 4 classes. The first observation is that when X is locally compact, for instance $X = \mathbb{R}$, $\mathcal{O}(X)$ is countably-based so it behaves very well in terms of descriptive complexity: symbolic and topological complexity coincide. We split the whole class of Polish spaces into four disjoint classes, ranging from the locally compact spaces to the non σ-compact spaces.

Let $X_{\mathrm{nk}} = \{x \in X : x \text{ has no compact neighborhood}\}$, which is a closed subset of X.

Definition 7.5.1. Let X be a Polish space.

1. $X \in$ Class I if $X_{\mathrm{nk}} = \emptyset$, i.e. X is locally compact,
2. $X \in$ Class II if $X_{\mathrm{nk}} \neq \emptyset$ is finite,
3. $X \in$ Class III if $X_{\mathrm{nk}} \neq \emptyset$ is infinite and X is σ-compact,
4. $X \in$ Class IV if X is not σ-compact.

Observe that these four classes form a partition of the whole class of Polish spaces, and that the union of Classes I, II, III is the class of σ-compact spaces.

Example 7.5.1. Let us give one example for each class:

1. $\mathbb{R}$ belongs to Class I,
2. $\mathcal{N}_1 = \{f \in \mathcal{N} : f \text{ takes at most one positive value}\}$ belongs to Class II, with one element having no compact neighborhood, namely the zero function f_0,
3. $\mathbb{N} \times \mathcal{N}_1$ belongs to Class III, where the elements with no compact neighborhood are the pairs (n, f_0),
4. $\mathcal{N}$ belongs to Class IV.

$\triangleleft$

Moreover, the three latter spaces are minimal in their respective classes, i.e. they embed into every space of their classes.

Proposition 7.5.1. *Let X be Polish.*

- *$X \in$ Classes II, III or IV $\iff$ X contains a closed copy of $\mathcal{N}_1$,*
- *$X \in$ Classes III or IV $\iff$ X contains a $\underset{\sim}{\mathbf{D}}_2$ copy of $\mathbb{N} \times \mathcal{N}_1$,*
- *$X \in$ Class IV $\iff$ X contains a closed copy of $\mathcal{N}$.*

Classification. We now relate the behavior of symbolic complexity on $\mathcal{O}(X)$ to the class of X. We first locate the symbolic complexity classes.

Theorem 7.5.3 (Classification – Positive results)**.** *Let X be Polish.*

1. *If $X \in$ Class I, then $[\underset{\sim}{\mathbf{\Sigma}}^0_k](\mathcal{O}(X)) = \underset{\sim}{\mathbf{\Sigma}}^0_k(\mathcal{O}(X))$ for all k,*
2. *If $X \in$ Class II, then $[\underset{\sim}{\mathbf{\Sigma}}^0_k](\mathcal{O}(X)) = \underset{\sim}{\mathbf{\Sigma}}^0_k(\mathcal{O}(X))$ for $k \geq 3$,*
3. *If $X \in$ Class III, then $[\underset{\sim}{\mathbf{\Sigma}}^0_k](\mathcal{O}(X)) \subseteq \underset{\sim}{\mathbf{\Sigma}}^0_{k+2}(\mathcal{O}(X))$ for $k \geq 2$.*

Informal proof. If $X \in$ Class I, i.e. if X is locally compact, then $\mathcal{O}(X)$ is countably-based [Sch02a], so symbolic and topological complexity coincide there (Theorem 7.3.1).

If $X \in$ Class II, then up to a finite set, X is countably-based, and it can be shown that this finite set does not affect the complexity of sets for levels $k \geq 3$.

If $X \in$ Class III, then X is σ-compact, so its open sets are σ-compact as well. Therefore, for each open set B, the corresponding set $P_B = \{U \in \mathcal{O}(X) : B \subseteq U\}$ belongs to $\underset{\sim}{\mathbf{\Pi}}^0_2(\mathcal{O}(X))$. Therefore, the images of cylinders under the representation of $\mathcal{O}(X)$ are $\underset{\sim}{\mathbf{\Pi}}^0_2$-sets. It can then be proved that the topological complexity of a set does not exceed its symbolic complexity by more than 2 levels. □

We then identify the gaps between symbolic and topological complexity.

Main Theorem 7.5.4: Classification – Negative results

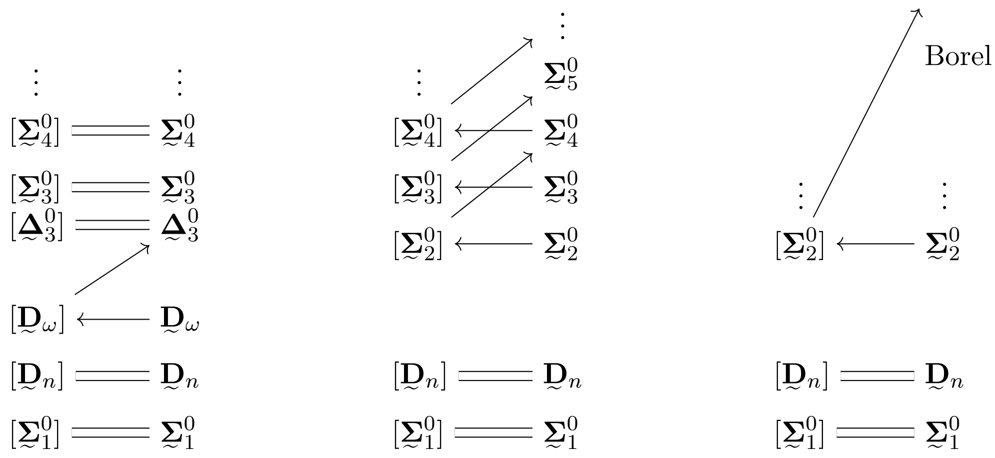
Let X be Polish.

1. If $X \notin$ Class I, then $[\underset{\sim}{\mathbf{D}}_\omega](\mathcal{O}(X))$ contains a set that is $\underset{\sim}{\mathbf{\Delta}}^0_3$ and not below,
2. If $X \notin$ Class II, then $[\underset{\sim}{\mathbf{\Sigma}}^0_k](\mathcal{O}(X))$ contains a set that is $\underset{\sim}{\mathbf{\Sigma}}^0_{k+1}$ and not below (for $k \geq 2$),
3. If $X \notin$ Class III, then $[\underset{\sim}{\mathbf{\Sigma}}^0_2](\mathcal{O}(X))$ contains a non-Borel set.

The results are summarized in Figure 7.5. We observe that two phenomena are possible:

- For some spaces, the classes $[\underset{\sim}{\mathbf{\Sigma}}^0_k]$ and $\underset{\sim}{\mathbf{\Sigma}}^0_k$ differ for low values of k and then coincide after some rank (if X is in Class II, then they coincide for $k \geq 3$).
- For other spaces, the classes $[\underset{\sim}{\mathbf{\Sigma}}^0_k]$ and $\underset{\sim}{\mathbf{\Sigma}}^0_k$ never coincide (if X is in Class III or IV).

For spaces in class III, the picture is not complete and despite our efforts we leave the following question open:



(a) Class II (b) Class III (c) Class IV

Figure 7.5: Comparison between symbolic and topological complexity on $\mathcal{O}(X)$

Open Question 7.5.5: Symbolic vs topological complexity on $\mathcal{O}(X)$

Let X belong to class III, for instance $X = \mathbb{N} \times \mathcal{N}_1$:

- Does inclusion $[\boldsymbol{\Sigma}^0_k(\mathcal{O}(X))] \subseteq \boldsymbol{\Sigma}^0_{k+1}(\mathcal{O}(X))$ hold?
- Or does $[\boldsymbol{\Sigma}^0_k(\mathcal{O}(X))]$ contain a set which is in $\boldsymbol{\Sigma}^0_{k+2}$ and not below?

A similar study could also be done when X is not Polish. For spaces in class IV like $\mathcal{N}$, we also raise the following question:

Open Question 7.5.6: Symbolic vs topological complexity on $\mathcal{O}(\mathcal{N})$

Is every set in $[\boldsymbol{\Delta}^0_2](\mathcal{O}(\mathcal{N}))$ Borel?

7.5.2 Discussion

These results give a rather precise picture of the relationship between symbolic and topological complexity on spaces of open subsets of Polish spaces. Let us now explain why they disagree.

Again, the key idea to prove separation results is to use the mismatch between sequential and topological concepts. All the separation results for spaces of open sets rely on the fact that the product topology of two sequential spaces is not in general sequential. Here is a simple example. When X is Hausdorff, it is well-know that the topology on $X \times \mathcal{O}(X)$ is sequential if and only if X is locally compact. Moreover, when X is not locally compact, the set $\mathscr{E} = \{(x, U) \in X \times \mathcal{O}(X) : x \in U\}$ is *sequentially* open but not open for the product topology. Because we are dealing with admissibly represented spaces, saying that $\mathscr{E}$ is sequentially open is equivalent to saying that it is open for the final topology of the product representation; indeed, there is an algorithm that can semidecide membership in $\mathscr{E}$.

The set $\mathscr{E}$ can then be used to build another set whose symbolic complexity is smaller than its topological complexity. How far X is from being locally compact has an impact on the topological complexity of $\mathscr{E}$ for the product topology, which implies specific results depending on the class X belongs to.

Observe that there is a reason why we use the set $\mathscr{E}$. Among the sets in $\underset{\sim}{\mathbf{\Sigma}}^0_1(X \times \mathcal{O}(X))$, it has maximal topological complexity for the product topology. Indeed, it is Wadge-complete for that class, when considering Wadge reductions that are continuous for the product topology.

Proposition 7.5.2. *In the product topology on $X \times \mathcal{O}(X)$, the set $\mathscr{E}$ is Wadge-complete for the class $\underset{\sim}{\mathbf{\Sigma}}^0_1(X \times \mathcal{O}(X))$.*

In other words, every sequentially open subset of $X \times \mathcal{O}(X)$ is Wadge-reducible to $\mathscr{E}$ via a function that is continuous for the product topology.

Proof. Let $\mathcal{U} \in \underset{\sim}{\mathbf{\Sigma}}^0_1(X \times \mathcal{O}(X))$. The mapping

$$\begin{aligned} \phi_{\mathcal{U}} : \mathcal{O}(X) &\to \mathcal{O}(X) \\ U &\mapsto \{x \in X : (x, U) \in \mathcal{U}\} \end{aligned}$$

is continuous, simply because it is continuously realizable ($\mathcal{U}$ can be seen as a function from $\mathcal{O}(X) \times X$ to the Sierpiński space, and currying that function gives $\phi_{\mathcal{U}}$). Therefore, one has

$$\begin{aligned} (x, U) \in \mathcal{U} &\iff x \in \phi_{\mathcal{U}}(U) \\ &\iff (x, \phi_{\mathcal{U}}(U)) \in \mathscr{E} \end{aligned}$$

so $\mathcal{U}$ is Wadge-reducible to $\mathscr{E}$ via the function $(\mathrm{id}, \phi_{\mathcal{U}})$. As this function is a pair of continuous functions, it is continuous for the product topology on $X \times \mathcal{O}(X)$. $\square$

Therefore, the complexity of $\mathscr{E}$ in the product topology, being maximal, is an indication of the discrepancy between the product topology and its sequentialization, and is the key ingredient in the proofs of the previous results:

- For $X \in$ Class I, $\mathscr{E}$ is open in the product topology (which is sequential),
- For $X \in$ Class II or III, $\mathscr{E}$ is $\underset{\sim}{\mathbf{\Pi}}^0_2$ in the product topology,
- For $X \in$ Class IV, $\mathscr{E}$ is not Borel in the product topology.

Chapter 8

Describing the open subsets of a represented space

Contents

8.1 Introduction

A way of understanding the information given by a representation is to identify the algorithmic tasks that can be performed using this representation. One of the simplest tasks one can imagine is to decide a property of the represented objects. It is often convenient to study the semidecidable properties rather than the decidable ones because they are often simpler to describe. Moreover, the decidable properties can be recovered as the ones which, together with their complements, are semidecidable.

As always, studying the relativized notion helps. A property is semidecidable if and only if its preimage under the representation is effectively open. Therefore, a property is semidecidabe relative to an oracle if and only if its preimage under the representation is open. Thus we are led to the following questions:

Problem

In a represented space,

- Identify the Σ^0_1 or semidecidable sets,
- Identify the $\underset{\sim}{\boldsymbol{\Sigma}}^0_1$ or open sets in the final topology of the representation.

The word "identify" is vague, but intuitively we would like to have an explicit description of those sets, or a receipe to build them from simpler objects. This problem usually boils down to finding a base or a subbase of the topology.

Example 8.1.1. In an effective countably-based T_0-space with its standard representation, the open sets are the unions of basic open sets, the semidecidable sets are the computable unions of basic open sets. ◁

Example 8.1.2. The space of real polynomials $\mathbb{R}[X]$ can be endowed with the representation giving an upper bound on the degree together with the coefficients. For $P \in \mathbb{R}[X]$, we denote by p_i its ith coefficient, so that $P = \sum_{i\in\mathbb{N}} p_i X^i$ (with $p_i = 0$ for almost all i). A subbase of open sets is given by the sets (see Theorem 8.2.1 below):

- $\{P \in \mathbb{R}[X] : p_i \in (a, b)\}$ where $i \in \mathbb{N}$ and $a, b \in \mathbb{Q}$,
- $\{P \in \mathbb{R}[X] : \forall i \in \mathbb{N}, |p_i| < \epsilon_i\}$, where $(\epsilon_i)_{i\in\mathbb{N}}$ is a sequence of positive rational numbers.

◁

For other spaces, finding such a basis is often difficult, because the definition of the final topology does not give a way to build a basis. There are spaces (such as the Kleene-Kreisel spaces presented in this chapter) in which no explicit basis is known. In that case, a way to tackle the problem is to identify the complexity of describing such a basis. From this point of view, the countably-based spaces are the simplest spaces because there is a *countable* indexing of a basis. For other topological spaces, the base-complexity introduced in [dBSS16] measures the simplest way of indexing a base of the space.

We start in Section 8.2 with the class of coPolish spaces, for which a rather concrete basis can be found. We continue in Section 8.3 with the notion of base-complexity of a space and our results about the base-complexity of Kleene-Kreisel functionals.

8.2 Open subsets of coPolish spaces

The class of coPolish spaces is one of the simplest classes of spaces beyond the countably-based spaces. We already saw them in the previous chapter (Definition 7.4.1). The definition of the open subsets of a coPolish space do not tell us what they look like, or how they can be obtained or described. In this section we provide a very explicit basis of the topology of a coPolish space. We will see an application of this result in Chapter 9, which is a characterization of the semidecidable properties of functions in any subrecursive class, like the primitive recursive functions (Theorem 9.4.1).

The content of this section is not published.

We are going to use an alternative way of describing coPolish spaces. Let (X, d) be a separable metric space, such that X can be expressed as the union $X = \bigcup_{n\in\mathbb{N}} X_n$ of a growing sequence of compact subsets $X_n \subseteq X_{n+1}$. Let τ be the inductive limit topology on X: $U \in \tau$ if each $U \cap X_n$ is d-open in X_n. It makes (X, τ) a coPolish space, and we will see that every coPolish space can be obtained this way (Proposition 8.2.2). Note that the topology τ is at least as strong as the topology induced by the metric d, and may be strictly stronger.

Proposition 8.2.1. *Let $\overline{\epsilon} = (\epsilon_n)_{n\in\mathbb{N}}$ be a sequence of positive real numbers. The set*

$$A_{\overline{\epsilon}} := \{x \in X : \forall n, d(x, X_n) < \epsilon_n\}$$

is open in the inductive limit topology.

Proof. The key is that the universal quantification over $n \in \mathbb{N}$ reduces to a *finite* quantification when x is taken in some X_p. More precisely, the intersection of $A_{\overline{\epsilon}}$ with X_p is $\{x \in X_p : \forall n < p, d(x, X_n) < \epsilon_n\}$ because for $n \geq p$ one has $d(x, X_n) = 0 < \epsilon_n$. Therefore, on X_p, $A_{\overline{\epsilon}}$ is a finite intersection of open sets so it is open. □

Theorem 8.2.1. *A subbase for the inductive limit topology on X is given by:*

- *The basic metric balls induced by the metric d,*
- *The sets $A_{\overline{\epsilon}}$, where $\overline{\epsilon}$ is any sequence of positive rational numbers.*

Proof. Let $U \subseteq X$ be τ-open, i.e. open in the inductive limit topology, and for each n, let U_n be a d-open subset of X such that $U \cap X_n = U_n \cap X_n$. Let $x \in U$ and $n_0 \in \mathbb{N}$ be such that $x \in X_{n_0}$.

As $x \in U \cap X_{n_0} \subseteq U_{n_0}$, there exists a basic metric ball B containing x, such that $\overline{B} \subseteq U_{n_0}$.

We define a sequence $\epsilon_n > 0$ so that $x \in B \cap A_{\overline{\epsilon}} \subseteq U$, which proves the result. More precisely, letting

$$A_n = \overline{N}(X_0, \epsilon_0) \cap \ldots \cap \overline{N}(X_{n-1}, \epsilon_{n-1}),$$

where $\overline{N}(S, r) = \{y \in X : d(y, S) \leq r\}$, we inductively define ϵ_n so that

$$\overline{B} \cap X_{n_0} \subseteq \overline{B} \cap A_n \subseteq U_n. \tag{8.1}$$

These inclusions imply what we want: $B \cap A_{\overline{\epsilon}} \subseteq \bigcap_n B \cap A_n \subseteq \bigcap_n U_n \subseteq U$.

We start with $\epsilon_0, \ldots, \epsilon_{n_0-1}$ large enough so that $\overline{B} \subseteq A_{n_0}$.

Assume by induction that (8.1) is satisfied for some $n \geq n_0$. One has $\overline{B} \cap A_n \cap X_n \subseteq U_n \cap X_n \subseteq U_{n+1}$, in other words

$$X_n \subseteq U_{n+1} \cup (\overline{B} \cap A_n)^c.$$

The left-hand side is d-compact and the right-hand side is d-open, so there exists $\epsilon_n > 0$ such that $\overline{N}(X_n, \epsilon_n) \subseteq U_{n+1} \cup (\overline{B} \cap A_n)^c$, i.e.

$$\overline{B} \cap A_{n+1} = \overline{B} \cap A_n \cap \overline{N}(X_n, \epsilon_n) \subseteq U_{n+1},$$

which concludes the induction step and finishes the proof. □

The proof is effective: if we start from a computable metric space (X, d) such that the sets X_n are uniformly effectively compact, then every effective open set in the inductive limit topology is a computable union of intersections of metric balls with sets $A_{\overline{\epsilon}}$, where the sequences $\overline{\epsilon}$ are uniformly computable.

Let us now show that every coPolish space can be obtained by starting from a metric.

Proposition 8.2.2. *Let X be a coPolish space. There exists a metric d on X inducing the original topology on each X_n.*

Proof. For each n, let $(s_{i,n})_{i \in \mathbb{N}}$ be a dense sequence in X_n. Let $f_{i,n} : X_n \to \mathbb{R}$ be defined as $f_{i,n}(x) = d_n(x, s_{i,n})$. By repeated application of the Tietze extension theorem from X_p to X_{p+1}, each function $f_{i,n}$ can be extended to a function $\tilde{f}_{i,n} : X \to \mathbb{R}$ that is continuous w.r.t. the coPolish topology. The function $f : X \to \mathbb{R}^\infty$ sending x to $(\tilde{f}_{i,n}(x))_{i,n \in \mathbb{N}}$ is an injective continuous function from X to a metric space, therefore it induces a metric d on X, defined by $d(x, y) = d(f(x), f(y))$. As each X_n is compact, the restriction of f to X_n is a homeomorphism between X_n and $f(X_n)$, so d induces the original topology on X_n. □

Again the proof is effective: if the sets X_n are computable metric spaces, uniformy in n, then the metric d is computable using the computable Tietze extension theorem proved by Weihrauch [Wei01].

We will use Theorem 8.2.1 in Chapter 9 to study the semidecidable properties of subrecursive functions.

8.3 Base-complexity hierarchy

We now study the complexity of indexing a base of a general topological space. Let us recall the definition from [dBSS16]. We work with a topological space X with an admissible representation δ_X. The space $\mathcal{O}(X)$ of open subsets of X can be endowed with the Scott topology and has a natural admissible representation: an open set $U \in \mathcal{O}(X)$ can be described by giving its pre-image $\delta_X^{-1}(U)$, which is an open subset of $\mathcal{N}$, as any list of finite strings whose infinite extensions are exactly the elements of $\delta_X^{-1}(U)$.

We recall the **projective hierarchy** $(\utilde{\boldsymbol{\Sigma}}^1_n)_{n\in\mathbb{N}}$:

- A set $A \subseteq \mathcal{N}$ is $\utilde{\boldsymbol{\Sigma}}^1_1$ or **analytic** if $A = f(\mathcal{N})$ for some continuous function $f : \mathcal{N} \to \mathcal{N}$,
- A set $A \subseteq \mathcal{N}$ is $\utilde{\boldsymbol{\Sigma}}^1_n$ if $A = f(\mathcal{N} \setminus B)$ for some $B \in \utilde{\boldsymbol{\Sigma}}^1_n$ and some continuous function $f : \mathcal{N} \to \mathcal{N}$.

Definition 8.3.1 (Base-complexity)**.** A space X has **base-complexity** $\utilde{\boldsymbol{\Sigma}}^1_n$ if there exists a continuous indexing $\phi : A \to \mathcal{O}(X)$ of a basis of X, for some $A \in \utilde{\boldsymbol{\Sigma}}^1_n(\mathcal{N})$.

Note that in the base-complexity hierarchy, there is no need to consider the $\utilde{\boldsymbol{\Pi}}^1_n$ levels, because being $\utilde{\boldsymbol{\Pi}}^1_n$-based is equivalent to being $\utilde{\boldsymbol{\Sigma}}^1_{n+1}$-based: indeed, the $\utilde{\boldsymbol{\Sigma}}^1_{n+1}$ sets are the continuous images of $\utilde{\boldsymbol{\Pi}}^1_n$ sets.

This hierarchy and the results of this section also extend to countable ordinal levels, but we only consider the finite levels for simplicity. The article [Hoy22] contains the general results.

In [dBSS16], it is partially shown that the hierarchy of topological spaces does not collapse: for each n, there is a $\utilde{\boldsymbol{\Sigma}}^1_{n+3}$-based space which is not $\utilde{\boldsymbol{\Sigma}}^1_n$-based. However whether $n+3$ can be replaced by $n+1$ is left open. We positively solve this problem (see Main Theorem 8.3.1 below).

A related problem is to identify the base-complexity of spaces of Kleene-Kreisel functionals $\mathbb{N}\langle n\rangle$.

Definition 8.3.2 (Kleene-Kreisel functionals)**.** The Kleee-Kreisel spaces $\mathbb{N}\langle n\rangle$ are inductively defined as follows:

$$\mathbb{N}\langle 0\rangle = \mathbb{N},$$
$$\mathbb{N}\langle n+1\rangle = \mathbb{N}^{\mathbb{N}\langle n\rangle} = \mathbb{N}^{\mathbb{N}^{\cdot^{\cdot^{\cdot^{\mathbb{N}}}}}}.$$

The space $\mathbb{N}\langle n\rangle$ comes with a natural function space representation, which induces a topology on $\mathbb{N}\langle n\rangle$ and which is admissible for that topology.

It is proved in [dBSS16] that $\mathbb{N}\langle n\rangle$ is $\utilde{\boldsymbol{\Sigma}}^1_{n+1}$-based, but whether it is $\utilde{\boldsymbol{\Sigma}}^1_n$-based is left open. We negatively solve this problem.

Main Theorem 8.3.1: Base-complexity of Kleene-Kreisel spaces

For each $n \geq 2$, $\mathbb{N}\langle n\rangle$ is not $\utilde{\boldsymbol{\Sigma}}^1_n$-based. In particular, the base-complexity hierarchy is proper.

Before discussing the proof of that result, let us explain what it really means. The topology on $\mathbb{N}\langle n\rangle$ is defined as the final topology of the representation: a set is open if its preimage under the representation is open. Therefore, it is defined *implicitly* but not *explicitly*. It does not tell us what the open sets look like or how they can be built from simpler objects. Still, the definition of the final topology gives a recipe: take an open subset U of $\mathcal{N}$, and test whether it is consistent with the representation in the sense that two names of the same point are on the same side of U; if that is the case, then U induces an open subset of $\mathbb{N}\langle n\rangle$, obtained as the image of U under the representation. However, this recipe is useless because it does not give us more information about how to obtain the open sets. Theorem 8.3.1 tells us that in a sense, there is no better recipe.

Indeed, the complexity of testing whether an open set $U \subseteq \mathcal{N}$ is consistent is precisely $\boldsymbol{\Pi}^1_n$, which explains why $\mathbb{N}\langle n\rangle$ is $\boldsymbol{\Pi}^1_n$-based, or equivalently $\boldsymbol{\Sigma}^1_{n+1}$-based. Therefore, the fact that $\mathbb{N}\langle n\rangle$ is not $\boldsymbol{\Sigma}^1_n$-based tells us that there is no simpler way of describing its open sets. However, it does not exclude the possibility of finding a more "concrete" indexing of the open subsets of $\mathbb{N}\langle n\rangle$ by elements of $\mathbb{N}\langle n+1\rangle$ for instance (they can be continuously indexed by a $\boldsymbol{\Pi}^1_n$-set), where "concrete" is understood in an informal and subjective sense.

We now briefly explain the proof of Theorem 8.3.1.

Informal proof. Our goal is to prove that there is no continuous surjection from a $\boldsymbol{\Sigma}^1_n$-subset of $\mathcal{N}$ to $\mathcal{O}(\mathbb{N}\langle n\rangle)$. We use the diagonal argument. To do so we first show that it can be reformulated as follows: there is no continuous surjection

$$\phi : \mathbb{N}\langle n\rangle \to \mathcal{C}(\mathbb{N}\langle n\rangle, \mathcal{O}(\mathbb{N}\langle n\rangle)). \tag{8.2}$$

This reformulation follows from the following facts:

- The $\boldsymbol{\Sigma}^1_n$-sets are the continuous images of $\mathbb{N}\langle n\rangle$ (Theorem 6.1 in [SS15]),
- Two elements of $\mathbb{N}\langle n\rangle$ can be encoded into one, i.e. there is a homeomorphism between $\mathbb{N}\langle n\rangle$ and $\mathbb{N}\langle n\rangle \times \mathbb{N}\langle n\rangle$, which we write $\mathbb{N}\langle n\rangle \cong \mathbb{N}\langle n\rangle \times \mathbb{N}\langle n\rangle$,
- So $\mathcal{O}(\mathbb{N}\langle n\rangle) \cong \mathcal{O}(\mathbb{N}\langle n\rangle \times \mathbb{N}\langle n\rangle) \cong \mathcal{C}(\mathbb{N}\langle n\rangle, \mathcal{O}(\mathbb{N}\langle n\rangle))$.

We then show that there is a continuous multi-valued function $h : \mathcal{O}(\mathbb{N}\langle n\rangle) \rightrightarrows \mathcal{O}(\mathbb{N}\langle n\rangle)$ without fixed-point, which means that for each input x, the set of outputs $h(x)$ does not contain x (see Theorem 8.3.2 below).

We finally apply the diagonal argument. If h was single-valued, then the argument would go as follows: if there is a continuous surjection $\phi : \mathbb{N}\langle n\rangle \to \mathcal{C}(\mathbb{N}\langle n\rangle, \mathcal{O}(\mathbb{N}\langle n\rangle))$ as in (8.2), then we take its diagonal composed with h, obtaining a continuous function $F \in \mathcal{C}(\mathbb{N}\langle n\rangle, \mathcal{O}(\mathbb{N}\langle n\rangle))$ defined by $F(x) = h(\phi(x)(x))$. This function F cannot be in the range of ϕ, because h has no fixed-point so for each x, $F \neq \phi(x)$ because $F(x) \neq \phi(x)(x)$. It contradicts the surjectiveness of ϕ.

However, this argument does not work because h is *multi-valued*, so the function F that we obtain is multi-valued as well: it is not an element of $\mathcal{C}(\mathbb{N}\langle n\rangle, \mathcal{O}(\mathbb{N}\langle n\rangle))$. However, it is possible to fix this problem by replacing $\mathbb{N}\langle n\rangle$ with an intermediate space on which every continuous multi-valued function admits a continuous single-valued selector. We do not present the details here, the complete solution can be found in [Hoy22]. $\square$

One of the key ingredients in the proof is the existence of a continuous multi-valued function $h : \mathcal{O}(\mathbb{N}\langle n\rangle) \rightrightarrows \mathcal{O}(\mathbb{N}\langle n\rangle)$ without fixed-point. This property is not specific to $\mathbb{N}\langle n\rangle$ and actually holds for any space which is not countably-based.

Theorem 8.3.2. *Let X be an admissibly represented topological space. The following statements are equivalent:*

- *X is countably-based,*
- *Every continuous multi-valued function $h : \mathcal{O}(X) \rightrightarrows \mathcal{O}(X)$ has a fixed-point.*

Observe that both Knaster-Tarski's and Kleene's fixed-point theorems imply that for *every* X, every continuous *single-valued* function $h : \mathcal{O}(X) \to \mathcal{O}(X)$ has a fixed-point. Thus, Theorem 8.3.2 is an illustration of the flexibility of multi-valued functions, in comparison with single-valued functions, and of their ability to detect topological properties of the underlying space. Another example of this flexibility is related to Brouwer's fixed-point theorem. That theorem asserts that every continuous function $h : [0,1]^n \to [0,1]^n$ has a fixed-point, while for every T_1-space X which is not a singleton, there exists a continuous multi-valued function $h : X \rightrightarrows X$ without fixed-point.

We explain the proof of Theorem 8.3.2.

Informal proof. We show that if X is not countably-based, then we can build a continuous multi-valued function $h : \mathcal{O}(X) \rightrightarrows \mathcal{O}(X)$ without fixed-point. One can think of h as an algorithm taking an open set $U \subseteq X$ as input and producing another open set $V \subseteq X$, such that $V \neq U$. The fact that h is a multi-valued function means that V does not only depend on U, but also on the way U is presented.

A name of an open set U essentially consists in a growing sequence of sets U_n such that $U = \bigcup_n U_n$. When X is countably-based, we can assume that the sets U_n are open (they are finite unions of basic open sets). However, when X is not countably-based, the U_n's are not open in general, and one can prove that there always exists an open set V, a point $x_0 \in V$ and a name of V consisting of a growing sequence V_n such that no V_n is a neighborhood of x_0. So we can produce a name of the neighborhood V of x_0 in such a way that no finite approximation of V is a neighborhood of x_0.

The pseudo-algorithm for h is the following. We start producing a name of V and test in parallel whether $x_0 \in U$. If $x_0 \notin U$, then in the limit we output V, which is different from U as it contains x_0. If we eventually see that $x_0 \in U$, then we stop producing an output, which is currently some V_n, and search for some $y \in U \setminus V_n$. We know that such a y exists, because V_n is not a neighborhood of x so it cannot contain U. Once we have found y, we extend the current output as a name of an open set that does not contain y: simply take $X \setminus \mathrm{cl}(\{y\})$. In that case, the open set that we produce is different from U because it does not contain y, which is in U.

To make the argument work, we actually need to show that the set $X \setminus \mathrm{cl}(\{y\})$ is consistent with our current output V_n. The trick is that we can always assume that the sets V_n, although not open, are upper sets for the specialization ordering, so that if $y \notin V_n$, then $\mathrm{cl}(\{y\})$ is disjoint from V_n. □

8.3.1 Discussion

Let us come back to our original question in Section 2.4 about higher-order functionals. Let us take $n = 2$, so that $\mathbb{N}\langle n\rangle = \mathbb{N}\langle 2\rangle = \mathbb{N}^{\mathbb{N}^{\mathbb{N}}}$. An element of $\mathbb{N}\langle 2\rangle$ is a continuous function $F : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ (also written $F(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$) and is represented as a list of pairs $(\sigma, n) \in \mathbb{N}^* \times \mathbb{N}$ such that $F(f) = n$ for all $f : \mathbb{N} \to \mathbb{N}$ extending σ, and so that every $f : \mathbb{N} \to \mathbb{N}$ has a prefix σ in this list. A way to understand what information such names give about F is to identify the properties of F that are semidecidable given such names. One can easily come up with more and more complicated properties:

1. Whether F is not constant is semidecidable,

2. Let $h : \mathbb{N} \to \mathbb{N}$ be some computable function. Whether $F(h) = 0$ is decidable,
3. Let $h \in \mathcal{N}$ be some computable function. Whether F is constantly 0 on $\{p \in \mathcal{N} : p \leq h\}$ is decidable,
4. For each $n \in \mathbb{N}$, let $h_n \in \mathcal{N}$ be some computable function. Whether $F(\lambda p.F(h_n)) = 0$ is decidable,
5. Let $F_0 \in \mathbb{N}\langle 2\rangle$ be computable. Whether $F \neq F_0$ is semidecidable.
6. etc.

Properties 2.,3.,4. give classes of decidable properties obtained by varying h, h_n. These classes of properties have computable Π^0_2-indexings because the total computable functions h have a computable Π^0_2-indexing.

Property 5. gives a class of semidecidable properties obtained by varying F_0. This class has a computable Π^1_1-indexing, because the computable elements F_0 of $\mathbb{N}\langle 2\rangle$ have a computable Π^1_1-indexing.

The effective version of Theorem 8.3.1 shows that the general semidecidable properties have no Π^1_1-indexing so there are many more properties, expressible in more complicated ways.

Chapter 9

Finite representations

Contents

9.1 Introduction

In my work as well as in most of the modern literature on computable analysis, the model of computation consists of oracle or type-2 Turing machines, i.e. machines that take an infinite object as input. This approach, initiated by Kreitz and Weihrauch in [KW85], has its origins in the Polish school of computable analysis led by Grzegorczyk.

Another approach to computable mathematics, known as the Russian school, was led by Markov. It is based on the computation model of ordinary, or type-1 Turing machines, that take a finite object as input. It is done by taking a constructivist viewpoint, allowing computable objects only, and representing them by finite programs. This approach culminated in the works of Ershov on numberings.

The comparison between these two models has been thoroughly investigated in the 50s, notably by Rice [Ric53], Shapiro [Sha56], Kreisel, Lacombe, Shoenfield [KLS57], Friedberg [Fri58] and Ceitin [Cei62]. Most of the results identify situations when these two models are equivalent (for instance, functions from computable reals numbers to computable real numbers), but Friedberg's result provides an example witnessing the difference. Further comparisons have been studied, for instance by Spreen [Spr98, Spr01, Spr10] and Hertling [Her05].

We are interested in comparing these two approaches.

The first difference is that Markov computability is limited to working with computable objects only, which already separates the two models: there is a computable function from the computable real numbers to the computable real numbers that cannot be extended to a computable function from the whole set of real numbers to real numbers. However we ignore this difference by comparing Markov and type-2 computability on computable objects only. A famous theorem due to Kreisel-Lacombe-Shoenfield and independently Ceitin states that the two models give the same class of computable functions from the space of computable real numbers to itself.

The second difference is that an infinite name can be derived from a finite program but not the other way round, so finite programs may contain more information than infinite names, making Markov computability more powerful than type-2 computability. Friedberg eventually showed that the two models are indeed non-equivalent in this sense.

We are interested in understanding the second difference through, by investigating what additional information is given by a finite representation, in comparison with an infinite one.

We are going to see that the only additional information given by a finite program is not contained in the program itself but in its size: every algorithmic task that can be performed given a finite program can be as well carried out given an infinite name and any upper bound on the size of a program computing the object.

Moreover, we study the case of objects lying in an arbitrary subrecursive class, i.e. objects that can be described by programs in a restricted programming language with total programs only. We give a characterization of the decidable properties of such objects, when represented by finite programs.

9.2 Historical results

Let us first review important results from the 50s comparing Markov and type-2 computability. The input objects are either partial or total computable functions on the natural numbers. The problem is then to investigate which properties of such objects are decidable or semidecidable in the two models.

For partial computable functions, the decidable and semidecidable properties are the same in the Markov and type-2 models. This is the content of Rice's and Rice-Shapiro's theorems [Ric53, Sha56].

For total computable functions, the Kreisel-Lacombe-Shoenfield/Ceitin's theorem states that the decidable properties are the same in the two models [KLS57, Cei62]. However, Friedberg surprisingly showed that the semidecidable properties are not the same in the two models [Fri58].

Computable functions	Decidability	Semidecidability
Partial	Markov $\equiv$ type-2 *Rice*	Markov $\equiv$ type-2 *Rice−Shapiro*
Total	Markov $\equiv$ type-2 *Kreisel et al/Tseitin*	Markov $>$ type-2 *Friedberg*

Figure 9.1: Comparison between Markov and type-2 computability

Selivanov [Sel84] showed that for partial functions, Markov computability and type-2 computability strongly disagree at the next level, decidability with 2 mind changes. More precisely, there is a property of partial functions which is:

- Decidable with 2 mind-changes in the Markov model,
- Not Σ_2^0 in the type-2 model.

9.3 What additional information?

We study objects living in an effective countably-based topological T_0-space X with its standard representation δ_S. In order to be fair in the comparison between the two models, we actually work with the subspace X_c of computable points of X.

- A δ_S**-name** of a point $x \in X$ is any enumeration of its basic neighborhoods,
- An **index** of x is any $i \in \mathbb{N}$ such that the ith partial computable function φ_i is a name of x.

In the type-2 model, points are given via names while in the Markov model, points are given via indices. Let δ_M be the representation of computable points by indices, where M stands for Markov.

In this section, we show that in many cases, having an index of x is equivalent to having a name of x and any upper bound on an index of x.

9.3.1 An intermediate representation

In order to formulate the results, we introduce another representation δ_K of X_c.

Definition 9.3.1. We define a representation $\delta_K :\subseteq \mathcal{N} \to X_c$ as follows. A δ_K**-name** of $x \in X_c$ is a pair (n, p) where

- n is any upper bound on the minimal index of x,
- p is a δ_S-name of x.

Note that having an upper bound on the minimal index of x is equivalent to having a finite set of natural numbers containing an index of x. The letter K in δ_K stands for Kolmogorov complexity, because the minimal index of x is a quantity that is similar to the Kolmogorov complexity of x.

Observe that from an index one can compute a δ_K-name, from which one can compute a δ_S. In general one cannot go in the other direction.

Proposition 9.3.1. *In general, the representations δ_{X_c}, δ_K and δ_M are not pairwise computably equivalent.*

It can be proved for $X_c = \mathcal{N}_c$ and many other spaces. Although δ_K and δ_M are usually non-equivalent, we show that they can be interchangeably used to perform most algorithmic tasks. More precisely, we show that they induce the same effective descriptive complexity classes, at least at some levels.

Friedberg and Selivanov examples show that the representations δ_S and δ_M do not induce the same effective descriptive complexity classes.

9.3.2 δ_K and δ_M are interchangeable

In Chapter 7 we considered the symbolic complexity of a subset of a represented space as the descriptive complexity of the corresponding set of names. We recall Definition 7.2.1: in a represented space (X, δ_X), and for a descriptive complexity class Γ,

$$A \in [\Gamma](X) \iff \delta^{-1}(A) \in \Gamma(\mathrm{dom}(\delta_X)).$$

We can apply this definition to δ_K and δ_M, which are representations of X_c.

Main Theorem 9.3.1: Finite vs infinite representation

Let X be an effective countably-based T_0-space and let $\Gamma \in \{\Sigma^0_1, \mathrm{D}_n, \Sigma^0_2\}$. The representations δ_K and δ_M induce the same complexity classes $[\Gamma](X_c)$.

For instance, the case $\Gamma = \Sigma^0_1$ can be reformulated as follows. A subset $A \subseteq X_c$ is semidecidable given an index of $x \in X_c$ if and only if A is semidecidable given a name of x and an upper bound on the minimal index of x.

In more concrete terms, it means that the only intrinsic information that a finite program p contains about the function $f : \mathbb{N} \to \mathbb{N}$ it computes is:

- The values of f, obtained by running the program p on any input,
- A bound on the Kolmogorov complexity of f, i.e. on the length of the shortest program computing f, obtained by measuring the length of p.

In particular, no intrinsic information about f is contained in the details of the code of the program.

9.4 Subrecursive classes

The previous results give some information about what properties can be decided or semidecided when the objects are given via finite programs. We would like to go further and obtain a characterization of these properties, or a way to generate all of them. It seems to be a difficult problem in general, but we can provide such a characterization in a restricted setting, where the objects are functions from a subrecursive class.

Perhaps surprisingly, the key for solving this purely computability-theoretic result is to use topology, by showing how a subrecursive class behaves like a coPolish space and using the understanding of coPolish spaces that we developed in CHapter 8.

Definition 9.4.1. A **subrecursive class** is a class $\mathcal{C}$ of total computable functions from $\mathbb{N}$ to $\mathbb{N}$ together with an effective enumeration $\mathcal{C} = \{f_i : i \in \mathbb{N}\}$, such that f_i is computable uniformly in i. If $f_i = f$ then i is called a **$\mathcal{C}$-index** of f.

Primitive recursive functions and usual complexity classes are subrecursive classes, but the whole class of total computable functions is not a subrecursive class. In a subrecursive class, an index can be seen as a program in some restricted programming language whose programs are all total. Having a $\mathcal{C}$-index of f is stronger than having any index in the standard numbering of partial computable functions. Is it strictly stronger? Does it provide more information about f? Does it allow to decide more properties of functions?

The answer is positive. Let us introduce a class of properties that are decidable from $\mathcal{C}$-indices, but not necessarily from arbitrary indices.

Definition 9.4.2. Let $\mathcal{C}$ be a subrecursive class. The **$\mathcal{C}$-complexity** of $f : \mathbb{N} \to \mathbb{N}$ is

$$K_{\mathcal{C}}(f) = \min\{i \in \mathbb{N} : f_i = f\}.$$

The **$\mathcal{C}$-complexity** of f up to n is

$$K_{\mathcal{C}}(f, n) = \min\{i \in \mathbb{N} : f_i(0) = f(0), \dots, f_i(n) = f(n)\}.$$

There are a few observations to make:

- f belongs to $\mathcal{C}$ if and only if $K_{\mathcal{C}}(f)$ is finite,
- For $f \in \mathcal{C}$, $K_{\mathcal{C}}(f,n)$ is nondecreasing in n and eventually constant with value $K_{\mathcal{C}}(f)$,
- For any $\mathcal{C}$-index i of f, one has $K_{\mathcal{C}}(f,n) \leq K_{\mathcal{C}}(f) \leq i$ for all $n \in \mathbb{N}$,
- The function $n \mapsto K_{\mathcal{C}}(f,n)$ is computable, given f as oracle.

A **computable order** is a nondecreasing unbounded computable function $h : \mathbb{N} \to \mathbb{N}$. Typical examples are $h(n) = n$, $h(n) = \lfloor\sqrt{n}\rfloor$ or $h(n) = \lfloor\log(n)\rfloor$.

Definition 9.4.3. Let h be a computable order. A function $f : \mathbb{N} \to \mathbb{N}$ is h**-compressible** if for every n, $K_{\mathcal{C}}(f,n) \leq h(n)$.

Proposition 9.4.1. *Let h be a computable order. For $f \in \mathcal{C}$, being h-compressible is decidable given a $\mathcal{C}$-index of f.*

Proof. Given a $\mathcal{C}$-index i of f, one has $K_{\mathcal{C}}(f,n) \leq K_{\mathcal{C}}(f) \leq i$ for all n. Compute n_0 such that $h(n_0) \geq i$. f is h-compressible if and only if $K_{\mathcal{C}}(f,n) \leq h(n)$ for all $n < n_0$, which is decidable. □

However, for most recursive classes, being h-compressible is not decidable given f as oracle. Observe that for trivial classes is may be decidable: if $\mathcal{C}$ is the class of constant functions with the obvious numbering $f_i(n) = i$, having a $\mathcal{C}$-index for f is equivalent to having f as oracle: given f as oracle, one can evaluate $i := f(0)$, which a $\mathcal{C}$-index of f (actually, the only one).

The class $\mathcal{C}$ is a subset of the Baire space endowed with the product topology. A function $f \in \mathcal{C}$ is not isolated in $\mathcal{C}$ if every neighborhood of f contains an another element of $\mathcal{C}$.

Proposition 9.4.2. *If $\mathcal{C}$ contains a non-isolated function then for some computable order h, being h-compressible is not decidable given f as oracle.*

Proof. Let f be a non-isolated function and i an index of f. We define h. First, let $h(0) = i$, so that f is h-compressible. We build h so that for each n there is some function $g \in \mathcal{C}$ that coincides with f on inputs $0, \dots, n$ and that is not h-compressible. As a result, f does not belong to the interior of the set of h-compressible, which is therefore not open.

Assume that $h(n)$ is defined. As f is not isolated there exist infinitely many functions that coincide with f on inputs $0, \dots, n$. As a result, $\mathcal{C}$ contains a function g such that $K_{\mathcal{C}}(g) > h(n)+1$, hence $K_{\mathcal{C}}(g,p) > h(n) + 1$ for some $p > n$. One can compute an index of such a g and p by exhaustive search. Let then $h(p) = h(n) + 1$ (and h is constant between n and $p - 1$). By construction, g is not h-compressible. □

Main Theorem 9.4.1: Semidecidable properties of subrecursive functions

Every semidecidable property P can be written as an effective disjunction of basic properties of the form "$f \in [u]$" and of the form "f is h-compressible".

More precisely, $P \subseteq \mathcal{C}$ is semidecidable if and only if there exist a computable sequence of finite strings $(u_i)_{i\in\mathbb{N}}$ and a computable sequence of order functions $(h_i)_{i\in\mathbb{N}}$ such that

$$f \in P \iff \exists i, f \in [u_i] \text{ and } f \text{ is } h_i\text{-compressible.}$$

In order to prove this result, we first compare two ways of representing a function $f \in \mathcal{C}$. The **$\mathcal{C}$-representation** is the representation using $\mathcal{C}$-indices: a $\delta_{\mathcal{C}}$-name for f is any $\mathcal{C}$-index of f, i.e. any i such that $f = f_i$ (names usually live in $\mathcal{N}$, but $\mathbb{N}$ can be easily embedded into $\mathcal{N}$, identifying i with the constant function with value i). The **K-representation** is defined as follows: a δ_K-name for f is any pair (k, f) such that k is an upper bound on $K_{\mathcal{C}}(f)$ (again, (k, f) can be seen as an element g of $\mathcal{N}$, defined by $g(0) = k$ and $g(n+1) = f(n)$). Observe that δ_K-names can be uniformly computed from $\delta_{\mathcal{C}}$-names: send i to the pair (i, f_i).

We obtain an analog to Theorem 9.3.1, but the proof is much easier because $\mathcal{C}$-indices only represent total functions, contrary to arbitrary indices.

Lemma 9.4.1. *The representations $\delta_{\mathcal{C}}$ and δ_K induces the same semidecidable properties.*

Proof. Let $A \subseteq \mathcal{C}$ be a $\delta_{\mathcal{C}}$-semidecidable set. Given a δ_K-name (k, f) for f, accept f if and only if for each $i \leq k$, either $f_i \in A$ or $f_i \neq f$. One can easily check that f is accepted if and only if $f \in A$. □

Proof of Theorem 9.4.1. The key is to observe that δ_K is a representation of $\mathcal{C}$ as a coPolish space, therefore we can apply the effective version of Theorem 8.2.1. Indeed, the class $\mathcal{C}$ can be decomposed into a growing union of uniformly effective compact sets

$$\mathcal{C}_n = \{f \in \mathcal{C} : \exists i \leq n, f_i = f\}.$$

In other words, $f \in \mathcal{C}_n \iff K_{\mathcal{C}}(f) \leq n$. The sets $\mathcal{C}_n$ are effectively compact in the metric topology inherited from the Baire space. The effective version of Theorem 8.2.1 shows that the δ_K-semidecidable properties in this coPolish space are generated by:

- The basic metric balls, which are the cylinders,
- The sets $A_\epsilon = \{f \in \mathcal{C} : \forall n, d(f, \mathcal{C}_n) < \epsilon_n\}$.

Taking $\epsilon_n = 2^{-h(n)}$, the set A_ϵ is precisely the set of h-compressible functions. □

The proof of this result shows how topology, notably the use of coPolish spaces, can give indsight into a purely computability-theoretic question.

9.5 Indexing complexity

Theorem 9.3.1 gives some information about the Markov semidecidable properties of total computable functions, but does not tell us what they look like in general, i.e. it does not provide an explicit description of them. We would like to obtain an analog to Theorem 9.4.1 for the whole class of total computable functions.

The idea of a "concrete description" is vague and can hardly be formalized. Let us have a closer look to the example of Theorem 9.4.1. It gives an explicit description of the semidecidable properties of functions in a subrecursive class, which is a recipe to build any such property from simpler objects: given a computable sequence of order functions $h_i : \mathbb{N} \to \mathbb{N}$ and a computable sequence of finite strings u_i, one can build a semidecidable property, and all of them can be obtained this way. One can measure the "simplicity" of this recipe by observing that it induces a computable indexing $(P_i)_{i \in A}$ of the semidecidable properties for some index set $A \in \Pi^0_2(\mathbb{N})$. One can show that this is optimal, i.e. that there is no Σ^0_2-indexing of the semidecidable properties (a Σ^0_2-indexing could actually be converted into a total indexing, which can be proved impossible).

In the quest to obtain an explicit description of the Markov semidecidable properties of total computabe functions, we identify their minimal indexing complexity, and show that they have no Π^0_2-indexing. It implies in particular that there is no effective receipe to build any such property from a computable sequence of order functions like in Theorem 9.4.1. On the other hand, the very definition of Markov semidecidable properties induces a natural Π^0_3-indexing (whether a Turing machine semidecides a property, i.e. whether it behaves the same of all indices of each total computable function, is a Π^0_3 predicate).

Theorem 9.5.1 ([Hoy22])**.** *There is no computable indexing $(P_i)_{i\in A}$ of the Markov semidecidable properties, with $A \in \Pi^0_2(\mathbb{N})$.*

Observe that the same result holds if $A \in \Sigma^0_3(\mathbb{N})$ because a Σ^0_3-indexing can always be turned into a Π^0_2-indexing: a Σ^0_3-set A is $A = \{i : \exists j, \langle i,j\rangle \in B\}$ where $B \in \Pi^0_2(\mathbb{N})$, so an indexing $(P_i)_{i\in A}$ induces the indexing $(Q_k)_{k\in B}$ where $Q_{\langle i,j\rangle} = P_i$.

Solving this question has been particularly difficult and was possible only after studying a different topic, the base-complexity of topological spaces (see Chapter 8) and developing a technique using fixed-points free algorithms. Again, a topological understanding can give effective tools to solve a computability-theoretic question.

Informal proof. As one might expect, the proof is based on the diagonal argument: given such an indexing, build a new property which is not listed.

A key ingredient in the diagonal argument is the existence of an algorithm with no fixed-point point: given an index of a Markov semidecidable property P, one can compute an index of a Markov semidecidable property $Q \neq P$ (it is not true of type-2 semidecidable, i.e. effectively open properties).

As in the proof of Theorem 8.3.1, more development is needed to make the diagonal argument actually work. The main issue is that the fixed-point free algorithm is not extensional: given different indices of the same P, it will output different properties $Q \neq P$. Half of the proof is about overcoming this problem. □

Chapter 10

Future directions

10.1 Computability of compact Polish spaces

We present several interrelated research projects about the computable aspects of compact Polish spaces, that we have already started.

10.1.1 Encoding information in a space

I have recently worked on the computability of compact Polish spaces, in collaboration with Kihara and Selivanov [HKS20]. The general problem is to understand, for a compact Polish space, the difficulty of producing a presentation of the space, which is the sequence of values of some complete metric one some dense sequence. This type of questions originates from computable structure theory, which is traditionally about countable structures such as graphs, countable groups or countable partial orders, and is about the intrinsic information that can be encoded in a structure. The study of topological spaces is particularly challenging because it is much more difficult to encode and extract information from a topology than from a discrete algebraic structure.

We have proved in [HKS20] that it is not possible to encode information in Polish spaces: if a set $A \subseteq \mathbb{N}$ is computable relative to every presentation of a Polish space X, then A must be computable. One of the main questions that remains open is whether information can be encoded into *compact presentations* of compact Polish spaces. A compact presentation is a presentation together with a list of the finite covers of the spaces. The precise question is then: is there a non-computable set $A \subseteq \mathbb{N}$ and a compact Polish space X whose compact presentations all compute A?

10.1.2 Computable type

I am currently working with a PhD student Djamel Eddine Amir on a related project about computability of compact sets. Several notions of computability exist for compact sets, the two most important ones are often called *computable sets* and *semicomputable sets*. It turns out that for some compact sets, these notions are equivalent, and that this equivalence comes from the topological properties of the set. It has led to the following definition (already encountered in Section 5.5.3): a set has **computable type** if every homeomorphic copy that is semicomputable is actually computable. It was proved by Miller [Mil02] that spheres have computable type, by Iljazović and Sušić [IS18] that closed manifolds have computable type, and other sets were studied by Iljazovič and his co-authors.

Our goal is to study broader classes of compact sets, and to relate more explicitly this computability property to the topological properties of the sets. We are led to investigating the computable aspects of algebraic topology, in particular homology.

We have obtained purely topological characterizations of the finite simplicial complexes having computable type [AH22]. In particular, these results enable one to reduce the global property to a local property of the vertices of the simplicial complex. Preliminary results in the 2-dimensional case suggest a relationship between this local property and homology. Our next project is to establish the precise relationship between the computable type property and homology.

10.1.3 Descriptive complexity of topological invariants

Informally, this project is about the difficulty of recognizing a certain set, given arbitrarily precise pictures of it. For instance, if one has access to a compact set drawn on a screen at any precision, how difficult is it to recognize that it is homeomorphic to a line segment? to a circle? to a disk? What properties can be recognized using a limited level of complexity?

All these questions are about the descriptive complexity of topological invariants. This question has been partially studied in the literature: it is known for instance that path-connectedness is Π^1_2-complete, simple connectedness is Π^1_1-hard [Bec92], local connectedness is Π^0_3-complete [DS20]. Most of the topological invariants arising from topology have high complexity and we are especially interested in the expressiveness of low complexity invariants. In other words, we want to understand what properties can be detected with limited complexity, for instance Σ^0_2.

A standard result from Descriptive Set Theory about actions of Polish groups implies that if we fix a compact space X, then the complexity of recognizing that a compact set is homeomorphic to X is Borel. It opens a whole line of research:

- For each compact space X, what is the exact complexity of recognizing its copies? We have already obtained that the complexity of recognizing copies of the line segment is precisely Π^0_3,
- Given two compact spaces X, Y, what is the complexity of distinguishing X from Y?
- Our work on the computable type property also reveals that Σ^0_2 topological invariants play an important role, which raises the question: which pairs of spaces X, Y can be distinguished using Σ^0_2 invariants?

10.2 Multi-valued functions

We briefly explain two projects about multi-valued functions.

10.2.1 Structural aspects of multi-valued functions

Functions are central in most mathematical theories. Each mathematical structure comes with a corresponding notion of morphisms, i.e. structure-preserving functions. By contrast, multi-valued, or set-valued functions are rather unusual.

Multi-valued functions are important in computable analysis, because many problems are naturally multi-valued: each instance may have several acceptable solutions. Moreover, recent works have shown that multi-valued functions can play an important role even in the study of problems that do not involve computability. In particular, multi-valued functions are able to capture certain properties of topological spaces, that ordinary functions cannot.

Let us give examples of this phenomenon. In a represented space (X, δ_X), the following problems have been recently studied:

1. Given a non-empty closed set $A \subseteq X$ with positive information, produce an element $x \in A$ (this problem is called *overt choice* in [dBPS20]),

2. Given a point $x \in X$ as input, output a point $y \neq x$ [BG21, Hoy22],

It turns out that when the represented space is a topological space, the solvability of these problems captures essential properties of the space:

- When X is countably-based and T_1, problem 1. is solvable if and only if X is quasiPolish [dBPS20]; when X is countably-based, a variant of problem 1. (taking any $\boldsymbol{\Pi}^0_2$-set rather than a closed set as input) is solvable if and only if X is quasiPolish (we are preparing an article presenting this result),

- When X is countably-based, problem 2. is unsolvable if and only if X is an ω-continuous domain with the Scott topology; when $X = \mathcal{O}(Y)$ is a space of open sets, then problem 2. is unsolvable if and only if Y is countably-based [Hoy22].

We would like to identify other classes of spaces where these problems are solvable. More generally, we want to investigate what properties of a space can be captured by multi-valued functions, and to better understand the interaction between multi-valued functions and structures.

10.2.2 Descriptive complexity of multi-valued functions

Descriptive Set Theory (DST) provides a full understanding of the complexity of subsets of Polish spaces. In computable analysis, one is naturally led to work in represented spaces rather than Polish spaces and with multi-valued functions rather than sets. Therefore, it would be useful to develop measures of descriptive complexity of multi-valued functions between represented spaces.

Let us briefly explain a feature of DST from which we would like to take inspiration. Descriptive set theory provides a notion of reducibility between sets (Wadge reducibility) as well as complexity hierarchies (Borel hierarchy, Hausdorff difference hierarchy, fine Hierarchy). Each individual set (among the Borel sets) can be assigned a definite complexity, and the comparison between sets reduces to the comparison between their complexities.

The situation is very different for functions and multi-valued functions. They can be compared using Weihrauch reducibility and its variants such as computable reducibility [BGP21, Dzh15], but there is no intrinsic complexity for an individual multi-valued function. Multi-valued functions have a wide range of aspects for which different complexity notions may be defined: the complexity of decomposing the input space to make it computable on each piece, the complexity of convergence of a mind-change algorithm, the minimal advice that is needed to make it computable, etc. These aspects are central in the literature on the computability of multi-valued functions and Weihrauch reducibility, for instance in [Bra05, BP10, Zie12, PdB12].

It would be interesting to define and study intrinsic notions of complexity for multi-valued functions. It would help classify problems by comparing their respective complexities. Moreover, knowing the intrinsic complexity (or complexities) of a problem is a more condensed and precise piece of information than knowing how it compares to all other problems, in the same way as saying that the set $\mathbb{Q}$ of rational numbers is Σ^0_2-complete in $\mathbb{R}$ summarizes all the isolated facts that each problem is or is not reducible to it.

We will take inspiration from notions recently introduced by Goh, Pauly and Valenti [GPV21] and Dzhafarov, Solomon and Yokoyama respectively: the deterministic and first-order parts of a multi-valued functions offer a way to capture some of its aspects.

Main definitions, theorems and open questions

Main definitions

Main theorems

Main open questions

Bibliography

[AH22] Djamel Eddine Amir and Mathieu Hoyrup. Computability of finite simplicial complexes. In *ICALP 2022*, LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. To appear. 8, 23, 46, 86

[AH23] Djamel Eddine Amir and Mathieu Hoyrup. Comparing computability in two topologies. Preprint, 2023. 40

[BCSS12] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation*. SpringerLink : Bücher. Springer New York, 2012. 4

[BDH$^+$12] Laurent Bienvenu, Adam R. Day, Mathieu Hoyrup, Ilya Mezhirov, and Alexander Shen. A constructive version of Birkhoff's ergodic theorem for Martin-Löf random points. *Information and Computation*, 210:21–30, 2012. 7, 22

[Bec92] Howard Becker. Descriptive set theoretic phenomena in analysis and topology. In Haim Judah, Winfried Just, and Hugh Woodin, editors, *Set Theory of the Continuum*, pages 1–25, New York, NY, 1992. Springer US. 86

[BEGH15] Guillaume Bonfante, Mohamed El-Aqqad, Benjamin Greenbaum, and Mathieu Hoyrup. Immune systems in computer virology. In Arnold Beckmann, Victor Mitrana, and Mariya Ivanova Soskova, editors, *Evolving Computability - 11th Conference on Computability in Europe, CiE 2015, Bucharest, Romania, June 29 - July 3, 2015. Proceedings*, volume 9136 of *Lecture Notes in Computer Science*, pages 127–136. Springer, 2015. 7, 23

[BG21] Vasco Brattka and Guido Gherardi. Completion of choice. *Ann. Pure Appl. Log.*, 172(3):102914, 2021. 87

[BGH$^+$11] Laurent Bienvenu, Peter Gács, Mathieu Hoyrup, Cristóbal Rojas, and Alexander Shen. Algorithmic tests and randomness with respect to a class of measures. *Proceedings of the Steklov Institute of Mathematics*, 274(1):34–89, November 2011. 7, 22

[BGP21] Vasco Brattka, Guido Gherardi, and Arno Pauly. Weihrauch complexity in computable analysis. In Vasco Brattka and Peter Hertling, editors, *Handbook of Computability and Complexity in Analysis*, Theory and Applications of Computability, pages 367–417. Springer, Cham, 2021. 87

[BH02] Vasco Brattka and Peter Hertling. Topological properties of real number representations. *Theor. Comput. Sci.*, 284(2):241–257, 2002. 4, 32

[BH03] Olivier Bournez and Mathieu Hoyrup. Rewriting logic and probabilities. In Robert Nieuwenhuis, editor, *RTA*, volume 2706 of *Lecture Notes in Computer Science*, pages 61–75. Springer, 2003. 23

[BH10] Laurent Bienvenu and Mathieu Hoyrup. Une brève introduction à la théorie de l'aléatoire. *Gazette des mathématiciens*, 123:35–47, 2010. 23

[BH21] V. Brattka and P. Hertling. *Handbook of Computability and Complexity in Analysis*. Theory and Applications of Computability. Springer International Publishing, 2021. 4

[BHS17] Laurent Bienvenu, Mathieu Hoyrup, and Alexander Shen. Layerwise computability and image randomness. *Theory Comput. Syst.*, 61(4):1353–1375, 2017. 7, 22

[BHW08] Vasco Brattka, Peter Hertling, and Klaus Weihrauch. A tutorial on computable analysis. In S. Barry Cooper, Benedikt Löwe, and Andrea Sorbi, editors, *New Computational Paradigms*, pages 425–491. Springer New York, 2008. 27

[Bis67] E. Bishop. *Foundations of Constructive Analysis*. McGraw-Hill series in higher mathematics. McGraw-Hill, 1967. 4

[BP03] Vasco Brattka and Gero Presser. Computability on subsets of metric spaces. *Theoretical Computer Science*, 305(1-3):43–76, 2003. 32

[BP10] Vasco Brattka and Arno Pauly. Computation with advice. In Xizhong Zheng and Ning Zhong, editors, *Proceedings Seventh International Conference on Computability and Complexity in Analysis, CCA 2010, Zhenjiang, China, 21-25th June 2010.*, volume 24 of *EPTCS*, pages 41–55, 2010. 32, 87

[Bra99] Vasco Brattka. Computable invariance. *Theor. Comput. Sci.*, 210(1):3–20, 1999. 32

[Bra05] Vasco Brattka. Effective Borel measurability and reducibility of functions. *Mathematical Logic Quarterly*, 51(1):19–44, 2005. 4, 39, 57, 87

[BW99] Vasco Brattka and Klaus Weihrauch. Computability on subsets of euclidean space i: Closed and compact subsets. *Theoretical Computer Science*, 219(1-2):65–93, 1999. 32

[Car21] Raphaël Carroy. Personal communication. 2021. 40, 42

[Cei62] G. S. Ceitin. Algorithmic operators in constructive metric spaces. *Trudy Matematiki Instituta Steklov*, 67:295–361, 1962. English translation: *American Mathematical Society Translations*, series 2, 64:1-80, 1967. 77, 78

[CFHR22] Daniel Coronel, Alexander Frank, Mathieu Hoyrup, and Cristóbal Rojas. Realizing semicomputable simplices by computable dynamical systems. *Theoretical Computer Science*, 2022. To appear. 7, 21, 39

[CH20] Antonin Callard and Mathieu Hoyrup. Descriptive complexity on non-Polish spaces. In Christophe Paul and Markus Bläser, editors, *37th International Symposium on Theoretical Aspects of Computer Science, STACS 2020, March 10-13, 2020, Montpellier, France*, volume 154 of *LIPIcs*, pages 8:1–8:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. 7, 14, 23, 41, 58, 60, 61, 62, 63

[dB13] Matthew de Brecht. Quasi-Polish spaces. *Ann. Pure Appl. Logic*, 164(3):356–381, 2013. 4, 14, 41, 57, 59, 60

[dBPS20] Matthew de Brecht, Arno Pauly, and Matthias Schröder. Overt choice. *Computability*, 9(3-4):169–191, 2020. 87

[dBSS16] Matthew de Brecht, Matthias Schröder, and Victor Selivanov. Base-complexity classifications of qcb_0-spaces. *Computability*, 5(1):75–102, 2016. 16, 70, 72

[dBY09] Matthew de Brecht and Akihiro Yamamoto. Σ^0_α - Admissible Representations (Extended Abstract). In Andrej Bauer, Peter Hertling, and Ker-I Ko, editors, *6th International Conference on Computability and Complexity in Analysis (CCA'09)*, volume 11 of *OpenAccess Series in Informatics (OASIcs)*, Dagstuhl, Germany, 2009. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. 14, 41, 57, 60

[DL02] Rodney G. Downey and Geoffrey LaForte. Presentations of computably enumerable reals. *Theor. Comput. Sci.*, 284(2):539–555, 2002. 54

[DS20] Gabriel Debs and Jean Saint Raymond. The descriptive complexity of connectedness in Polish spaces. *Fundamenta Mathematicae*, 249(3):261–286, 2020. 86

[Dzh15] Damir D. Dzhafarov. Cohesive avoidance and strong reductions. *Proceedings of the American Mathematical Society*, 143(2):869–876, 2015. 87

[Eda97] Abbas Edalat. Domains for computation in mathematics, physics and exact real arithmetic. *The Bulletin of Symbolic Logic*, 3(4):401–452, 1997. 4

[Esc04] Martín Escardó. Synthetic topology: of data types and classical spaces. *Electronic Notes in Theoretical Computer Science*, 87:21–156, 2004. Proceedings of the Workshop on Domain Theoretic Methods for Probabilistic Processes. 4

[FGH13] Hugo Férée, Walid Gomaa, and Mathieu Hoyrup. On the query complexity of real functionals. In *Proceedings of the 28th Annual IEEE Symposium on Logic in Computer Science, LICS 2013, June 25-28, New Orleans, LA, USA*, pages 103–112. IEEE Computer Society, 2013. 7, 23

[FGH14] Hugo Férée, Walid Gomaa, and Mathieu Hoyrup. Analytical properties of resource-bounded real functionals. *J. Complexity*, 30(5):647–671, 2014. 7, 23

[FH13] Hugo Férée and Mathieu Hoyrup. Higher-order complexity in analysis. CCA 2013, July 8-10, Nancy, 2013. 23

[FHHP10] Hugo Férée, Emmanuel Hainry, Mathieu Hoyrup, and Romain Péchoux. Interpretation of stream programs: Characterizing type 2 polynomial time complexity. In Otfried Cheong, Kyung-Yong Chwa, and Kunsoo Park, editors, *Algorithms and Computation - 21st International Symposium, ISAAC 2010, Jeju Island, Korea, December 15-17, 2010, Proceedings, Part I*, volume 6506 of *Lecture Notes in Computer Science*, pages 291–303. Springer, 2010. 7, 23

[FHHP15] Hugo Férée, Emmanuel Hainry, Mathieu Hoyrup, and Romain Péchoux. Characterizing polynomial time complexity of stream programs using interpretations. *Theor. Comput. Sci.*, 585:41–54, 2015. 7, 23

[Fri58] Richard M. Friedberg. Un contre-exemple relatif aux fonctionnelles récursives. *Comptes Rendus de l'Académie des Sciences*, 247:852–854, 1958. 77, 78

[GHR09a] Peter Gács, Mathieu Hoyrup, and Cristóbal Rojas. Randomness on computable probability spaces - A dynamical point of view. In Susanne Albers and Jean-Yves Marion, editors, *STACS*, LIPIcs, pages 469–480. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2009. 21

[GHR09b] Stefano Galatolo, Mathieu Hoyrup, and Cristóbal Rojas. A constructive Borel-Cantelli lemma. Constructing orbits with required statistical properties. *Theoretical Computer Science*, 410(21-23):2207–2222, 2009. 21

[GHR10a] Stefano Galatolo, Mathieu Hoyrup, and Cristóbal Rojas. Computing the speed of convergence of ergodic averages and pseudorandom points in computable dynamical systems. In Xizhong Zheng and Ning Zhong, editors, Proceedings Seventh International Conference on *Computability and Complexity in Analysis,* Zhenjiang, China, 21-25th June 2010, volume 24 of *Electronic Proceedings in Theoretical Computer Science*, pages 7–18. Open Publishing Association, 2010. 21

[GHR10b] Stefano Galatolo, Mathieu Hoyrup, and Cristóbal Rojas. Effective symbolic dynamics, random points, statistical behavior, complexity and entropy. *Information and Computation*, 208(1):23–41, 2010. 22

[GHR11a] Péter Gács, Mathieu Hoyrup, and Cristóbal Rojas. Randomness on computable probability spaces - a dynamical point of view. *Theory of Computing Systems*, 48(3):465–485, 2011. 22

[GHR11b] Stefano Galatolo, Mathieu Hoyrup, and Cristóbal Rojas. Dynamics and abstract computability: computing invariant measures. *Discrete and Continuous Dynamical Systems - A*, 29(1):193–212, jan 2011. 7, 21

[GHR12] Stefano Galatolo, Mathieu Hoyrup, and Cristóbal Rojas. Statistical properties of dynamical systems - simulation and abstract computation. *Chaos, Solitons and Fractals*, 45(1):1–14, 2012. 7, 21

[GPV21] Jun Le Goh, Arno Pauly, and Manlio Valenti. Finding descending sequences through ill-founded linear orders. *The Journal of Symbolic Logic*, 86(2):817–854, 2021. 88

[Gra64] R. L. Graham. On finite sums of reciprocals of distinct nth powers. *Pacific J. Math.*, 14(1):85–92, 1964. 55

[Gra74] Jacques Grassin. Index sets in Ershov's hierarchy. *The Journal of Symbolic Logic*, 39:97–104, 3 1974. 58, 64

[Grz57] Andrzej Grzegorczyk. On the definitions of computable real continuous functions. *Fundamenta Mathematicae*, 44(1):61–71, 1957. 4

[Her05] Peter Hertling. A Banach-Mazur computable but not Markov computable function on the computable real numbers. *Ann. Pure Appl. Log.*, 132(2-3):227–246, 2005. 77

[HG17] Mathieu Hoyrup and Walid Gomaa. On the extension of computable real functions. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017,* *Reykjavik, Iceland, June 20-23, 2017*, pages 1–12. IEEE Computer Society, 2017. 7, 12, 22, 50, 54

[HKL08] Mathieu Hoyrup, Arda Kolçak, and Giuseppe Longo. Computability and the morphological complexity of some dynamics on continuous domains. *Theoretical Computer Science*, 398(1-3):170–182, 2008. 21

[HKS20] Mathieu Hoyrup, Takayuki Kihara, and Victor Selivanov. Degree Spectra of Homeomorphism Types of Polish Spaces. working paper or preprint, April 2020. 8, 85

[Hoy07] Mathieu Hoyrup. Dynamical systems: stability and simulability. *Mathematical Structures in Computer Science*, 17(2):247–259, 2007. 21

[Hoy11] Mathieu Hoyrup. Randomness and the ergodic decomposition. In Benedikt Löwe, Dag Normann, Ivan N. Soskov, and Alexandra A. Soskova, editors, *CiE*, volume 6735 of *Lecture Notes in Computer Science*, pages 122–131. Springer, 2011. 7, 22

[Hoy12] Mathieu Hoyrup. The dimension of ergodic random sequences. In Christoph Dürr and Thomas Wilke, editors, *STACS*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012. 7, 22

[Hoy13] Mathieu Hoyrup. Computability of the ergodic decomposition. *Annals of Pure and Applied Logic*, 164(5):542–549, 2013. 7, 21, 32, 38

[Hoy14] Mathieu Hoyrup. Irreversible computable functions. In Ernst W. Mayr and Natacha Portier, editors, *31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014), STACS 2014, March 5-8, 2014, Lyon, France*, volume 25 of *LIPIcs*, pages 362–373. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014. 6, 11, 12, 22, 37, 38

[Hoy15] Mathieu Hoyrup. Que calcule cet algorithme ?, 2015. Images des Mathématiques, CNRS. 19, 23

[Hoy16a] Mathieu Hoyrup. The decidable properties of subrecursive functions. In *ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 108:1–108:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. 7, 19, 23

[Hoy16b] Mathieu Hoyrup. The typical constructible object. In Arnold Beckmann, Laurent Bienvenu, and Natasa Jonoska, editors, *Pursuit of the Universal - 12th Conference on Computability in Europe, CiE 2016, Paris, France, June 27 - July 1, 2016, Proceedings*, volume 9709 of *Lecture Notes in Computer Science*, pages 115–123. Springer, 2016. 24

[Hoy17] Mathieu Hoyrup. Genericity of weakly computable objects. *Theory Comput. Syst.*, 60(3):396–420, 2017. 6, 11, 12, 22, 50, 53

[Hoy18] Mathieu Hoyrup. Topological analysis of representations. In Florin Manea, Russell G. Miller, and Dirk Nowotka, editors, *Sailing Routes in the World of Computation - 14th Conference on Computability in Europe, CiE 2018, Kiel, Germany, July 30 - August 3, 2018, Proceedings*, volume 10936 of *Lecture Notes in Computer Science*, pages 214–223. Springer, 2018. 24

[Hoy20a] Mathieu Hoyrup. Algorithmic randomness and layerwise computability. In Johanna N. Y. Franklin and Christopher P. Porter, editors, *Algorithmic Randomness: Progress and Prospects*, chapter 4, pages 115–133. Cambridge University Press, Cambridge, 2020. 7, 24

[Hoy20b] Mathieu Hoyrup. Descriptive complexity on non-Polish spaces II. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 132:1–132:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. 7, 14, 23, 58, 63, 64

[Hoy22] Mathieu Hoyrup. The fixed-point property for represented spaces. *Annals of Pure and Applied Logic*, 173(5):103090, 2022. 7, 17, 19, 23, 72, 73, 83, 87

[HR09a] Mathieu Hoyrup and Cristóbal Rojas. An application of Martin-Löf randomness to effective probability theory. In Klaus Ambos-Spies, Benedikt Löwe, and Wolfgang Merkle, editors, *CiE*, volume 5635 of *Lecture Notes in Computer Science*, pages 260–269. Springer, 2009. 7, 22

[HR09b] Mathieu Hoyrup and Cristóbal Rojas. Applications of effective probability theory to Martin-Löf randomness. In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris E. Nikoletseas, and Wolfgang Thomas, editors, *ICALP 2009*, volume 5555 of *Lecture Notes in Computer Science*, pages 549–561. Springer, 2009. 7, 22

[HR09c] Mathieu Hoyrup and Cristóbal Rojas. Computability of probability measures and Martin-Löf randomness over metric spaces. *Information and Computation*, 207(7):830–847, 2009. 22

[HR15] Mathieu Hoyrup and Cristóbal Rojas. On the information carried by programs about the objects they compute. In *STACS 2015, March 4-7, 2015, Garching, Germany*, volume 30 of *LIPIcs*, pages 447–459. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. 7, 19, 23

[HR17] Mathieu Hoyrup and Cristóbal Rojas. On the information carried by programs about the objects they compute. *Theory Comput. Syst.*, 61(4):1214–1236, 2017. 7, 19, 23, 32

[HR21] Mathieu Hoyrup and Jason Rute. Computable Measure Theory and Algorithmic Randomness. In Vasco Brattka and Peter Hertling, editors, *Handbook of Computability and Complexity in Analysis*. Springer, 2021. 7, 24

[HRSS19] Mathieu Hoyrup, Cristóbal Rojas, Victor L. Selivanov, and Donald M. Stull. Computability on quasi-Polish spaces. In Michal Hospodár, Galina Jirásková, and Stavros Konstantinidis, editors, *Descriptional Complexity of Formal Systems - 21st IFIP WG 1.02 International Conference, DCFS 2019, Košice, Slovakia, July 17-19, 2019, Proceedings*, volume 11612 of *Lecture Notes in Computer Science*, pages 171–183. Springer, 2019. 7, 23

[HRW11] Mathieu Hoyrup, Cristóbal Rojas, and Klaus Weihrauch. Computability of the Radon-Nikodym derivative. In Benedikt Löwe, Dag Normann, Ivan N. Soskov, and Alexandra A. Soskova, editors, *CiE*, volume 6735 of *Lecture Notes in Computer Science*, pages 132–141. Springer, 2011. 7, 22

[HRW12] Mathieu Hoyrup, Cristóbal Rojas, and Klaus Weihrauch. Computability of the Radon-Nikodym derivative. *Computability*, 1(1):3–13, 2012. 7, 22

[HS19] Mathieu Hoyrup and Donald M. Stull. Semicomputable points in Euclidean spaces. In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen, editors, *44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, August 26-30, 2019, Aachen, Germany*, volume 138 of *LIPIcs*, pages 48:1–48:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. 7, 12, 22, 56

[HSS18] Mathieu Hoyrup, Diego Nava Saucedo, and Don M. Stull. Semicomputable geometry. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPIcs*, pages 129:1–129:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. 7, 12, 22, 56

[HY88] J.G. Hocking and G.S. Young. *Topology*. Addison-Wesley series in mathematics. Dover Publications, 1988. 27

[Ilj11] Zvonko Iljazović. Co-c.e. spheres and cells in computable metric spaces. *Logical Methods in Computer Science*, 7(3), 2011. 32

[Ing81] M.A. Ingrassia. *P-genericity for Recursively Enumerable Sets*. PhD thesis, University of Illinois at Urbana-Champaign, 1981. 50

[IS18] Zvonko Iljazović and Igor Sušić. Semicomputable manifolds in computable topological spaces. *J. Complex.*, 45:83–114, 2018. 32, 46, 85

[Joc80] Carl G. Jockush. Degrees of generic sets. In F. R. Drake and S. S. Wainer, editors, *Recursion Theory: its Generalisations and Applications*, pages 110–139. Cambridge University Press, 1980. 50

[Joc85] Carl G. Jockusch. Genericity for recursively enumerable sets. In Heinz-Dieter Ebbinghaus, Gert H. Müller, and Gerald E. Sacks, editors, *Recursion Theory Week*, pages 203–232, Berlin, Heidelberg, 1985. Springer Berlin Heidelberg. 50

[Jon99] Sara Hawtrey Jones. Applications of the Baire category theorem. *Real Anal. Exchange*, 23(2):363–394, 1999. 11

[Kec95] Alexander S. Kechris. *Classical Descriptive Set Theory*. Springer, January 1995. 59

[KLS57] Georg Kreisel, Daniel Lacombe, and Joseph R. Shoenfield. Fonctionnelles récursivement définissables et fonctionnelles récursives. *Comptes Rendus de l'Académie des Sciences*, 245:399–402, 1957. 77, 78

[KNP19] Takayuki Kihara, Keng Meng Ng, and Arno Pauly. Enumeration degrees and non-metrizable topology. arXiv:1904.04107, 2019. 32, 33

[KP14] Takayuki Kihara and Arno Pauly. Point degree spectra of represented spaces. *CoRR*, abs/1405.6866, 2014. 4, 32, 33

[Kur81] S. Kurtz. *Randomness and genericity in the degrees of unsolvability*. PhD thesis, University of Illinois at Urbana, 1981. 50

[KW85] Christoph Kreitz and Klaus Weihrauch. Theory of representations. *Theoretical Computer Science*, 38:35 – 53, 1985. 4, 18, 77

[Lac59] Daniel Lacombe. Quelques procédés de définition en topologie récursive. In A. Heyting, editor, *Journal of Symbolic Logic*, pages 129–158. Amsterdam: North-Holland Pub. Co., 1959. 4

[Lac73] A.H. Lachlan. The priority method for the construction of recursively enumerable sets. In A.R.D. Mathias and H. Rogers, editors, *Cambridge Summer School in Mathematical Logic*, volume 337 of *Lecture Notes in Mathematics*, pages 299–310. Springer Berlin Heidelberg, 1973. 50

[Lut21] Patrick Lutz. *Results on Martin's Conjecture*. PhD thesis, University of California, Berkeley, 2021. 40

[Maa82] Wolfgang Maass. Recursively enumerable generic sets. *The Journal of Symbolic Logic*, 47(4):809–823, 1982. 50

[Mil02] Joseph S. Miller. Effectiveness for embedded spheres and balls. *Electronic Notes in Theoretical Computer Science*, 66(1):127 – 138, 2002. CCA 2002, Computability and Complexity in Analysis (ICALP 2002 Satellite Workshop). 32, 46, 85

[NR85] A. Nerode and J. B. Remmel. Generic objects in recursion theory. In Heinz-Dieter Ebbinghaus, Gert H. Müller, and Gerald E. Sacks, editors, *Recursion Theory Week*, pages 271–314, Berlin, Heidelberg, 1985. Springer Berlin Heidelberg. 50

[Pau16] Arno Pauly. On the topological aspects of the theory of represented spaces. *Computability*, 5(2):159–180, 2016. 4, 27

[PdB12] Arno Pauly and Matthew de Brecht. Non-deterministic computation and the Jayne-Rogers theorem. In Benedikt Löwe and Glynn Winskel, editors, *Proceedings 8th International Workshop on Developments in Computational Models, DCM 2012, Cambridge, United Kingdom, 17 June 2012*, volume 143 of *EPTCS*, pages 87–96, 2012. 32, 87

[PdB15] Arno Pauly and Matthew de Brecht. Descriptive set theory in the category of represented spaces. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 438–449. IEEE Computer Society, 2015. 58

[PER89] Marian B. Pour-El and J. Ian Richards. *Computability in Analysis and Physics*. Perspectives in Mathematical Logic. Springer, Berlin, 1989. 32, 36, 37

[PS09] Janusz Pawlikowski and Marcin Sabok. The Solecki dichotomy for bianalytic maps. Preprint, 2009. 34, 40

[PS12] Janusz Pawlikowski and Marcin Sabok. Decomposing Borel functions and structure at finite levels of the Baire hierarchy. *Annals of Pure and Applied Logic*, 163(12):1748–1764, 2012. 34, 40

[Ray07] Jean Saint Raymond. Preservation of the Borel class under countable-compact-covering mappings. *Topology and its Applications*, 154(8):1714 – 1725, 2007. 57

[Ric53] Henry G. Rice. Classes of recursively enumerable sets and their decision problems. *Transactions of the American Mathematical Society*, 74(2):pp. 358–366, 1953. 77, 78

[Ric54] H. G. Rice. Recursive real numbers. *Proceedings of the American Mathematical Society*, 5:784–791, 1954. 4

[Šan68] N.A. Šanin. *Constructive Real Numbers and Constructive Function Spaces*, volume 21 of *Translations of Mathematical Monographs*. American Mathematical Society, Providence, 1968. 4

[Sch02a] Matthias Schröder. *Admissible Representations for Continuous Computations*. PhD thesis, FernUniversität Hagen, 2002. 4, 14, 29, 58, 66

[Sch02b] Matthias Schröder. Extended admissibility. *Theor. Comput. Sci.*, 284(2):519–538, 2002. 14

[Sch04] Matthias Schröder. Spaces allowing type-2 complexity theory revisited. *Math. Log. Q.*, 50(4-5):443–459, 2004. 4, 61

[Sch21] Matthias Schröder. Admissibly represented spaces and Qcb-spaces. In Vasco Brattka and Peter Hertling, editors, *Handbook of Computability and Complexity in Analysis*. Springer, 2021. 27

[Sel84] Victor L. Selivanov. Index sets in the hyperarithmetical hierarchy. *Siberian Mathematical Journal*, 25:474–488, 1984. 58, 78

[Sel08] Victor Selivanov. On the difference hierarchy in countably based t0-spaces. *Electronic Notes in Theoretical Computer Science*, 221:257–269, 12 2008. 57

[Sha56] Norman Shapiro. Degrees of computability. *Transactions of the American Mathematical Society*, 82:281–299, 1956. 77, 78

[Sol98] Slawomir Solecki. Decomposing Borel sets and functions and the structure of Baire class 1 functions. *Journal of the American Mathematical Society*, 11(3):521–550, 1998. 34, 40

[Spr98] Dieter Spreen. On effective topological spaces. *Journal of Symbolic Logic*, 63(1):185–221, 1998. 77

[Spr01] Dieter Spreen. Representations versus numberings: on the relationship of two computability notions. *Theor. Comput. Sci.*, 262(1):473–499, 2001. 77

[Spr10] Dieter Spreen. Effectivity and effective continuity of multifunctions. *J. Symb. Log.*, 75(2):602–640, 2010. 77

[SS15] Matthias Schröder and Victor L. Selivanov. Hyperprojective hierarchy of qcb_0-spaces. *Computability*, 4(1):1–17, 2015. 73

[Tay11] Paul Taylor. *Foundations for Computable Topology*, pages 265–310. Springer Netherlands, Dordrecht, 2011. 4

[Tur36] Alan Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2, 42:230–265, 1936. 4

[Wei00] Klaus Weihrauch. *Computable Analysis*. Springer, Berlin, 2000. 4, 27, 29, 31

[Wei01] Klaus Weihrauch. On computable metric spaces Tietze-Urysohn extension is computable. In Jens Blanck, Vasco Brattka, and Peter Hertling, editors, *Computability and Complexity in Analysis*, pages 357–368, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. 71

[Zie02] Martin Ziegler. Computability on regular subsets of euclidean space. *Mathematical Logic Quarterly*, 48(S1):157–181, 2002. 32

[Zie12] Martin Ziegler. Real computation with least discrete advice: A complexity theory of nonuniform computability with applications to effective linear algebra. *Annals of Pure and Applied Logic*, 163(8):1108 – 1139, 2012. Continuity, Computability, Constructivity: From Logic to Algorithms. 4, 32, 87

Index